TESIS DE MAESTRÍA EN INGENIERÍA DE SOFTWARE

UNIVERSIDAD
NACIONAL
DE LA PLATA

MODELOS DE PREDICCIÓN AVANZADOS PARA EL CÁLCULO DE RESERVAS EN LA INDUSTRIA ASEGURADORA

AUTOR: **ING. ALEJANDRO AGUIRRE**

DIRECTOR: **DR. MARIO MATÍAS URBIETA**

CONTENIDO

1 INTRODUCCIÓN

La industria del seguro es una actividad que se caracteriza por ofrecer un producto que, si se lo considera en forma aislada, se rige a partir del principio de equidad, en donde se realiza un contrato de seguro cuyo precio se ajusta lo más posible a las características del riesgo que cubre, garantizando todos los derechos y obligaciones que dimanen de dicha operación (Nieto de Alba, 1964). Pero desde otra perspectiva predomina el principio de estabilidad, en donde surge el problema de las medidas que se deben tomar para garantizar la estabilidad de la Compañía Aseguradora. Esta es una de las partes más delicadas, en donde la falta de técnica puede llevar consigo que no se mantenga dicha estabilidad o a que ésta se consiga con un elevado precio del seguro, resultando un perjuicio para la masa de asegurados.

Desde el punto de vista de los clientes, las compañías de seguros venden promesas. A cambio de un precio o prima, ante la ocurrencia de un siniestro garantizan que el asegurado obtendrá una cierta cantidad de beneficio de acuerdo con una suma asegurada (Kopczyk, 2019). Después de la venta de dicha póliza, pueden pasar muchos años hasta que ese siniestro ocurra, por lo que para cubrir los pagos futuros, las aseguradoras establecen una reserva llamada Reserva de Siniestro o Reserva de Riesgos, calculada por el departamento financiero y los actuarios, que son los estadísticos en una compañía de seguros.

Dichas reservas representan la provisión económica que las empresas aseguradoras deben disponer para hacer frente al pago de los posibles siniestros que podrían ocurrir en el futuro, como consecuencia directa de las coberturas vigentes que les brindan a sus asegurados (Moreno, 2015). Su cálculo se basa en la siniestralidad esperada dentro del ámbito de cobertura de los seguros, y en los gastos administrativos que esa intermediación genera, y es independiente al monto de la prima cobrada a los clientes.

Los países definen de forma normativa los tipos de reservas y su proceso de cálculo, siendo en nuestro país la Superintendencia de Seguros de la Nación (SSN) la entidad reguladora, quien establece en el Artículo 33 de su Reglamento General de la Actividad Aseguradora, las reservas que deben figurar en los balances y situaciones patrimoniales de las empresas aseguradoras.

Si bien la ley define los procesos de cálculo, los cuales son complejos procedimientos iterativos sobre grandes volúmenes de datos, es importante que las empresas dispongan de herramientas más simples de estimación (López Prior, 2016). Cuando mejor sea la predicción de esas reservas que representan los riesgos de siniestralidad, mayores serán las mejoras tanto en ahorro de costos directos, como en la aplicación de primas personalizadas que permitan capturar clientes.

Adicionalmente, esa capacidad de predecir el monto correcto de un siniestro tiene un impacto significativo en las decisiones de administración y en los estados financieros de las empresas aseguradora (Kopczyk, 2019). Prediciendo la cantidad final de siniestros y estimando la Reserva de Riesgos con precisión, las aseguradoras puedan por un lado utilizar efectivamente el capital que no esté inmovilizado o destinado a construir reservas, y por otro lado tomar mejores decisiones de gestión sobre inversiones, nuevos productos y estrategia de ventas, generando además confianza y estabilidad a través de estados financieros precisos. No genera una preocupación si en un período un rubro o producto ofrecido por una aseguradora es deficitario si se tiene en cuenta sus

resultados técnicos provenientes de la diferencia entre lo cobrado y lo pagado, porque la ganancia se puede generar a partir de los resultados financieros producto de invertir el dinero en el tiempo transcurrido entre que se cobra la prima y se paga el siniestro.

Tradicionalmente, las aseguradoras predicen el monto de la reserva tomando grupos similares de pólizas y analizando su desarrollo histórico de siniestros. Basado en patrones de datos históricos, se producen algunas estimaciones de los montos de los siniestros. Además, se deben incluir ajustes expertos para acomodar el hecho de que los datos son históricos. Luego, los grupos de pólizas se agregan para obtener el monto de la Reserva de siniestros que se colocará en el estado financiero.

Dicho método produce una buena estimación de la Reserva de Riesgos, pero el pago final de esos siniestros siempre es algo diferente al esperado. Esa diferencia responde a dos razones:

1. Primero, los actuarios toman el grupo de pólizas similares y no tienen en cuenta las características individuales de los asegurados. Por lo tanto, puede resultar que algunas personas no estén siguiendo los patrones de un grupo, y las afirmaciones son mucho más pequeñas o más altas para ellos, lo que afecta el resultado agregado.

2. En segundo lugar, en el tiempo entre la presentación inicial de un siniestro y el pago completo, el monto final puede cambiar drásticamente. Por lo tanto, las aseguradoras tienen que establecer una reserva pendiente adicional por dichas pérdidas que se han incurrido.

Por otro lado, y en contraste con éstos procedimientos tradicionales y normativos, la industria del seguro tiene una necesidad urgente de innovación luego del arribo del concepto de “Insurtech”, que propone un mercado mucho más competitivo que se basa justamente en la innovación y en la aplicación de tecnología al negocio, que permita tomar decisiones enfocadas en un cliente cada vez más exigente (Alarcón Madrid L, 2018). Dicho contexto se caracteriza por una movilidad permanente en los clientes y una pérdida de fidelidad con las aseguradoras, manifestándose siempre en movimiento en busca de mejores condiciones y servicios. Esa movilidad se ha elevado a un nivel en el que nunca antes había estado, en donde la personalización de la oferta se convierte en vital para ser competitivo, y en un futuro, para la supervivencia en el mercado. De ésta manera se hace patente la necesidad de un cambio en la interacción con los clientes, que ha dado origen a la aparición de nuevos competidores directos en el sector capaces de apalancar su ventaja competitiva en la relación ofrecida al cliente mediante el uso de nuevas tecnologías, especialmente aprovechando la actual y creciente capacidad de captación (BigData) y análisis de datos (Analítica Avanzada).

En ese sentido, se pueden utilizar ciertos beneficios de usar un enfoque de Machine Learning en el problema de predicción de reservas, ya que los algoritmos pueden descubrir patrones a partir de datos que no se usan en enfoques tradicionales (ej.: todas las características de un asegurado) (Kopczyk, 2019). Por otro lado, debido al enfoque automatizado pueden predecir los pagos en un nivel de póliza individual capturando efectos no lineales en los datos originados justamente en esas características ocultas para los procedimientos tradicionales de estimación. Todos estos puntos se traducen en una mejor previsibilidad y una mayor estabilidad del modelo de reservas. Además, el uso de la inteligencia artificial puede automatizar en gran medida el proceso,

de modo que los especialistas pueden dedicar su tiempo al análisis de resultados, en lugar de ocuparse de los cálculos descuidados en una planilla. Como hay una aceleración en el proceso, los representantes de ventas pueden obtener inmediatamente el costo de siniestro esperado, basado en los datos anteriores que conducen a mejores decisiones de ventas.

El objetivo general de éste trabajo es analizar el funcionamiento de diferentes modelos de predicción avanzados, basados en técnicas de Machine Learning, que puedan estimar las reservas futuras con antelación a la ejecución de los tradicionales procedimientos de cálculo usados en la industria de acuerdo a lo definido por la ley. De ésta manera disponer de una herramienta rápida de estimación que permita planificar y tomar decisiones con anterioridad a la finalización de los procesos batch que realizan los cálculos definitivos al final de cada ciclo contable.

Dichos modelos pueden funcionar tanto a nivel de las reservas de pólizas individuales, que permitan disponer de una estimación rápida incluso en el mismo momento de la cotización, que origine decisiones que pueden llegar hasta el rechazo del negocio por parte de la aseguradora. También los modelos pueden funcionar a nivel de las reservas generales mensuales y disponer de una curva en una serie temporal, que contenga la estimación de la reserva general tanto para el mes en curso como para los próximos meses, que permita tomar decisiones relacionadas con los instrumentos de inversión de corto y mediano plazo, sin estar acoplados a la espera de los largos procesos de cálculo de reserva mencionados anteriormente.

Los objetivos específicos del presente trabajo son:

- Relevar y seleccionar los tipos de reservas que dispone una empresa aseguradora sobre los cuales se va a trabajar, y su procedimiento de cálculo tradicional.
- Recolectar y preparar datos de una Entidad Aseguradora.
- Definir las técnicas de Machine Learning y algoritmos de predicción a utilizar.
- Ejecutar experimentos y obtener el mejor modelo de predicción posible de cada técnica.
- Analizar resultados y comparar métricas.
- Concluir acerca del funcionamiento de cada mecanismo.

El estudio se encuentra diagramado de la siguiente manera. En la Sección 2 se presenta el estado del arte en donde se definen los conceptos, herramientas y terminología básica, tanto de la industria aseguradora como de Machine Learning, para luego profundizar en los trabajos relacionados en ésta temática de aplicación de algoritmos predictivos en la industria aseguradora. En la Sección 3 se sigue un proceso metodológico adaptado a Machine Learning basado en el estándar CRISP-DM, en donde se define concretamente el problema y los criterios de evaluación, se preparan datos, se entrenan modelos tanto para la predicción de reservas en pólizas individuales como la acumulada en forma mensual en una serie temporal, se analizan los errores y finalmente se ponen a prueba los modelos resultantes implementándolos en los sistemas productivos de la organización y contrastándolos con la realidad durante 6 meses. En la Sección 4 se describen las conclusiones del nivel de precisión de cada modelo para los diferentes escenarios, la problemática de implementar éste tipo de procesos en una empresa de seguros de magnitud, y las líneas futuras de investigación que se desprenden del presente trabajo.

1.1 MOTIVACIÓN, CONTRIBUCIONES Y LIMITACIONES

La industria del seguro se encuentra en una puja entre innovación y regulación, ya que por un lado emerge el concepto de "Insurtech" y su necesidad de aplicación de tecnología al negocio ante un mercado cada vez más competitivo, pero por otro lado se trata de una actividad muy regulada por los organismos gubernamentales, que establecen un marco común de aplicación de procedimientos y técnicas que las aseguradoras deben respetar.

En esa puja está el concepto de las Reservas que representan uno de los indicadores más importantes a tener en cuenta en la toma de decisiones porque establece el límite de la libre disponibilidad de fondos que la compañía puede invertir para obtener resultados financieros. Si bien las Reservas las define la ley, tanto en concepto como en procedimiento de cálculo, disponer de herramientas más simples y rápidas de predicción aumentará la certidumbre sobre las disponibilidades monetarias mucho tiempo antes de disponer los resultados finales de los cálculos y permitirá mejorar la toma de decisiones en la compañía.

Pero el elemento más innovador de esta presentación no es solamente la aplicación de algoritmos de predicción avanzados sobre el concepto de reservas en una empresa aseguradora, sino el enfoque hacia el proceso completo de implementación de éste tipo de herramientas en una compañía de magnitud. El estudio no solo se enfoca en los distintivos procesos de modelado, entrenamiento y testing de las técnicas de Machine Learning, sino que enfatiza las problemáticas que surgen en todas las etapas, desde las dificultades para entender el problema, conseguir y preparar los datos, hasta el mantenimiento posterior a la puesta en producción de los modelos.

En cuanto a las limitaciones, al ser una industria muy regulada, las empresas están obligadas a disponer de fuertes procedimientos de seguridad y confidencialidad sobre sus datos, por lo que si bien en el presente trabajo se utilizaron datos reales de una compañía de seguros de magnitud, solamente se presentarán los resultados resumidos sin mencionar la compañía ni exponer datos de los contratos individuales utilizados.

2 ESTADO DEL ARTE Y TRABAJO RELACIONADOS

2.1 ACTIVIDAD ASEGURADORA

2.1.1 Contrato del Seguro

El artículo 1° de la Ley Nro 17.418 denominada "***Ley de Seguros***" (1967), establece que habrá "***contrato de seguro***" cuando el "***asegurador***" se obligue, mediante el pago de una "***prima***" por parte del "***asegurado***", a resarcir un daño o cumplir la prestación convenida si ocurre el evento previsto, denominado "***siniestro***".

De ésta manera el "***asegurado***" es la persona que toma el seguro y sobre el que recae la cobertura para protegerlo del "***riesgo***", y el "***asegurador***", es quien asume la cobertura de dicho riesgo.

El "***riesgo***" es la posibilidad de ocurrencia de determinado hecho desfavorable o la medida del peligro de ocurrencia de un daño (Fernandez Dirube A, 2012). Aunque parezca impropio, esta palabra se ha transformado en el mundo del seguro y se la usa también como designación del bien real asegurado. En este sentido un automóvil, un buque, una casa son "riesgos" materiales, sujetos a su vez al "riesgo" aleatorio de incendio, avería o robo. Los riesgos (como peligro), en materia aseguradora deben ser futuros e inciertos en su aparición. La única excepción, si se trata de hechos de existencia necesaria como la muerte, la incertidumbre puede alcanzar sólo al aspecto temporal.

Para que exista seguro es indispensable que se produzca contractualmente una ***"transferencia***" del riesgo previsto, por parte del amenazado de daño. El seguro constituye en realidad esa transferencia, pero no del "riesgo" en sí mismo, sino de las consecuencias económicas del siniestro, provocado por el riesgo previsto, de las que se hace cargo el asegurador en la medida transferida por el asegurado, que puede ser parcial respecto del daño. Sin "transferencia" no existe seguro.

La celebración finalmente del contrato se origina en una "***propuesta***" de contrato de seguro que el asegurado hace conocer al asegurador (Meilij G, 1990). Esta propuesta no constituye un precontrato, y no resulta obligatorio ni para el asegurado ni el asegurador.

La aceptación de la propuesta sólo se produce por una manifestación positiva de parte del asegurador. Al aceptar la propuesta manifiesta formalmente su actitud mediante la emisión de una póliza que entrega al asegurado debidamente firmada, claramente redactada y fácilmente legible.

La "***póliza***" es la instrumentación por escrito del contrato de seguro y esencialmente contiene los nombres y domicilios de las partes contratantes, el interés o la persona asegurada, los riesgos asumidos, el momento inicial y el plazo de duración del contrato (vigencia de la póliza), la prima, la

suma asegurada y las condiciones generales, particulares y específicas que servirán de regla de conducta de las partes y determinarán sus derechos y obligaciones.

Puede resultar que el contenido específico o general de la póliza difiera con el de la propuesta que la generó, pues quizá el asegurador no ·ha aceptado los requisitos de la proposición tal como los planteó el asegurado. Si al asegurado no le conviniera el aseguramiento tal como lo ofrece el asegurador en la póliza que difiere de la propuesta, puede optar por rescindir el contrato a ese momento.

Por último, los "***certificados***" son los documentos por el que el asegurador da fe de la existencia de ciertas coberturas sobre un determinado objeto o persona (Mapfre, A., 1990). Normalmente el certificado sólo recoge las condiciones particulares del contrato y se emite a la póliza base previamente suscripta.

2.1.2 Tipos de Seguros

La amplitud del mundo del seguro, da lugar a diversas versiones o ***"ramos"*** de sus ofertas de coberturas y, consecuentemente, a diversas clasificaciones de su materia polifacética (Fernadez Dirube, 2012). El criterio más comúnmente utilizado es la diferenciación entre ***"Seguros Patrimoniales"*** y ***"Seguros de Personas"***.

A dicha clasificación, agregamos una tercera categoría que es un seguro de persona que en nuestro país tiene particularidades especiales, por lo que lo trataremos en una categoría aparte, y son los ***"Seguro de Riesgos del Trabajo"***. A continuación se describen los tipos de seguros definidos por la Superintendencia de Seguros de la Nación (SSN) y la Superintendencia de Riesgos del Trabajo (SRT):

- **Seguro Patrimoniales**: son los que aseguran un patrimonio (algo susceptible a tener valor), y se dividen entre los que aseguran cosas (vehículos, hogares, fábricas, etc.), y los seguros de responsabilidad civil, que protegen al asegurado en caso de que se produzcan reclamaciones por daños a terceros.
    - ***Seguro combinado familiar integral o Seguro de hogar:*** es una cobertura que abarca los daños causados por: incendio, efectos del agua, robo, responsabilidad civil y accidentes personales y del personal del servicio doméstico.
    - ***Seguro de Responsabilidad Civil:*** aquí el asegurador se compromete a indemnizar al asegurado del perjuicio patrimonial de su obligación de reparar los daños y perjuicios causados a terceros, por hechos de los cuales sea civilmente responsable. Para el caso de los propietarios de vehículos, éste seguro es obligatorio.
    - ***Seguro de Vehículos Automotores y/o Remolcados:*** la cobertura básica es el Seguro Obligatorio de Responsabilidad Civil hacia Terceros Transportados y no Transportados. Adicionalmente el Asegurado puede optar por una Voluntaria de Responsabilidad Civil como así también optar por la cobertura del casco del vehículo (Daños por Accidente, y/o Incendio, y/o Robo y/o Hurto).

- ***Seguro contra robo:*** otorga una indemnización al asegurado por daño o pérdida de los bienes asegurados, derivados de la sustracción ilegítima por parte de terceros efectuada con violencia y/o intimidación.

- ***Seguro de Desempleo Involuntario:*** el Seguro de Desempleo Involuntario paga al acreedor beneficiario, las cuotas correspondientes al servicio de la deuda del asegurado que no puedan ser pagadas a causa de cesantía involuntaria.

- ***Seguros Agrícola:*** tienen por objeto la cobertura de los riesgos que puedan afectar a las explotaciones agrícolas, ganaderas o forestales. Se utilizan para cubrir los daños producidos en los riesgos asegurables en función de la ubicación de la explotación, el cultivo, la hacienda, el rinde, etc., y algunas de las coberturas son: incendio, helada, lluvia, nieve, granizo, viento, etc..

- ***Seguro de Saldo Deudor:*** es un Seguro Colectivo orientado a cubrir a las entidades financieras a empresas cuya actividad sea vender a crédito, en cuotas o prestar dinero; y a sus clientes, especialmente de créditos hipotecarios, tarjetas de crédito, prendas, descubiertos en cuenta corriente, préstamos personales, entre otros. Es un beneficio para ambas partes, ya que en caso de fallecer el asegurado, la entidad se garantiza el abono total de la deuda y a la vez su familia queda liberada del pago de la misma.

- ***Seguro de Garantía:*** indemniza al asegurado por los daños patrimoniales sufridos dentro de los límites establecidos en la ley o en el contrato, en caso de que el tomador del seguro no cumpla con sus obligaciones legales o contractuales.

- ***Seguro de Incendio asociado a créditos hipotecarios:*** es un seguro exigido por las entidades crediticias que cubre los daños al inmueble dado en garantía hipotecaria en caso de incendio. Se pueden contratar coberturas adicionales tales como daños a causa de sismos, salida de mar, rotura de cañerías, etc.

- ***Seguro de Incendio para bienes inmuebles:*** paga una indemnización en caso de incendio con pérdida total del inmueble asegurado en la póliza. En caso de pérdida parcial, paga la reparación de dicho bien.

- ***Seguro de Transporte:*** otorga indemnizaciones a consecuencia de los daños sobrevenidos durante el transporte terrestre, marítimo y/o aéreo de mercaderías. Estos daños pueden afectar al objeto transportador (seguro de casco) o a las propias mercaderías transportadas.

- ***Seguros de Pérdidas Pecuniarias Diversas:*** indemnizan en el caso de que se produzca una pérdida de rendimiento económico que podría haberse obtenido si no se hubiera dado el siniestro reflejado en el contrato. Por ejemplo, una interrupción en la producción de una fábrica a consecuencia de una avería cubierta por la póliza de seguros. En este caso, el asegurador va a indemnizar al asegurado por la pérdida económica que dicha avería le haya ocasionado (pérdida de beneficios).

- ***Seguro de Todo Riesgo Operativo:*** los seguros de ingeniería cubren los riesgos derivados del funcionamiento, montaje o prueba de maquinaria o inherentes a la construcción de edificios y obras.
- ***Seguros Multirriesgos:*** cubren diversos riesgos, como indica su nombre, en una sola póliza. Por ejemplo, seguros de hogar que incluyen, entre otras coberturas: daños materiales, asistencia, efectos del agua, robo, accidentes personales, accidentes del personal doméstico y responsabilidad civil.
- ***Seguro de mascotas:*** asegura mascotas que viven en el seno familiar.
- ***Microseguros:*** se utilizar para proteger un micro-emprendimiento o personas excluidas de los sistemas formales de protección social, particularmente trabajadores de la economía informal y su familia.

- **Seguro de personas:** protegen a los asegurados de hechos donde se ve afectada la vida, salud y/o integridad de las personas. En éste tipo de seguros, el pago de la indemnización no guarda relación con el valor del daño producido por la ocurrencia del seguro, ya que una persona no es evaluable económicamente. Se dividen en:
  - ***Seguro de Vida:*** permite resguardar la economía de las familias ante distintas causas, para poder sostener el ingreso económico y mantener la calidad de vida en caso de fallecimiento por cualquier razón del asegurado. Como riesgos complementarios existen las coberturas que prevén una indemnización en caso de invalidez total y permanente debida a una enfermedad o accidente y ante la ocurrencia de eventos que afecten la salud.
  - ***Seguro Colectivo de Vida Obligatorio:*** cubre el riesgo de muerte e incluye el suicidio como hecho indemnizable, sin limitaciones de ninguna especie, de todo trabajador en relación de dependencia, cuyos empleadores se encuentren o no obligados con el Sistema Único de la Seguridad Social.
  - ***Seguro de Accidentes Personales***: otorga una indemnización al asegurado a consecuencia de las lesiones producidas por un accidente.
  - ***Seguro de Vida con Ahorro:*** otorga una indemnización a los beneficiarios en caso de fallecimiento del asegurado por una causa cubierta en la póliza, pero se distingue porque permite generar un ahorro de una suma de dinero, que puede ser retirada en caso de una anulación o fin de vigencia.
  - ***Seguros Mixtos:*** seguro mixto se garantiza el pago de una suma de dinero ya sea en caso de fallecimiento o de supervivencia. Es decir que, en caso de fallecimiento, la compañía aseguradora indemnizará a la familia beneficiaria; y si no, vencido el contrato abona la suma al asegurado. Estos seguros combinan dos características: un seguro de fallecimiento (en el que indemnizan a la familia) y uno de supervivencia (que sirve como un ahorro al asegurado).
  - ***Seguro de Retiro:*** a diferencia de los anteriores, éste seguro está enteramente dedicado al ahorro. Se caracteriza por tener dos etapas: ahorro y rentas. En la primera,

cuando el asegurado aún se desempeña laboralmente como activo, se genera un ahorro que se puede retirar de forma total o parcial en cualquier momento. Una vez iniciada la segunda etapa al momento del retiro, no se puede retirar el dinero aportado y acumulado, sino que el beneficiario comienza a recibir pagos (las rentas garantizadas) según la modalidad de cobro que se haya convenido durante la contratación. Se trata de un producto orientado a complementar la jubilación. ya que te permite disfrutar de sus beneficios una vez que dejaste de trabajar.

- ***Seguro de sepelio:*** brinda el servicio de sepelio o reembolsa los gastos incurridos por el sepelio del asegurado, según la modalidad contratada y hasta los límites establecidos en el contrato.
- ***Seguro de salud:*** es un seguro que brinda protección económica ante la ocurrencia de eventos que afecten los riesgos de salud específicamente previstos en el contrato y con el alcance que allí se especifique. Las coberturas mayormente ofrecidas por las entidades aseguradoras bajo esta clasificación suelen ser: cáncer, enfermedades graves, trasplantes de órganos, prótesis y ortesis, intervenciones quirúrgicas, entre otras.

- **Seguro de Riesgos del Trabajo:** en el año 1995 se estableció la ley Nro 24557, que generó el marco para la creación de un nuevo tipo de empresa aseguradora de carácter especial: "Aseguradoras de Riesgos del Trabajo" o ART. La ley tuvo el objetivo de reducir la siniestralidad laboral a través de la prevención de los riesgos laborales, basándose en la obligación de desarrollar planes de mejoramiento y de vigilar continuamente las condiciones y medio ambiente de trabajo, como asimismo la de monitorear el estado de salud de los trabajadores, derivado de la exposición a estos riesgos, a través de la realización de exámenes médicos. Define un listado cerrado de enfermedades profesionales, correlacionando el agente de riesgo, la actividad y la enfermedad. directa entre riesgo laboral y daño producido, lo que implica que las condiciones de trabajo no son adecuadas y dañan. La definición de enfermedad profesional implica un daño en la salud del trabajador expuesto a ciertos riesgos laborales. Es decir, hay una correlación directa entre riesgo laboral y daño producido, lo que implica que las condiciones de trabajo no son adecuadas y dañan. Adicionalmente se crea un organismo independiente para la regulación de éste nuevo tipo de aseguradoras, que es la Superintentencia de Riesgos del Trabajo.

2.1.3 Prima, Premio y Suma Asegurada

La relación obligacional emergente del contrato de seguro genera, desde la perspectiva del asegurado como sujeto pasivo o deudor, el deber jurídico de cumplimiento de una prestación principal, que tiene por objeto una suma de dinero (Stiglitz R, Stiglitz G, 1988). Tal suma de dinero constituye un precio, equivalente al valor de la prestación del asegurador que se halla condicionada a la verificación del siniestro.

La "***prima***" es el nombre con que se califica ese precio en el contrato de seguro (Di Giorgio, A., 2006). Desde el punto de vista técnico-económico, la prima es esencial porque el asegurador no puede cubrir el riesgo de ocurrencia del siniestro sin recaudar fondos suficientes, cuanto desde la óptica jurídica, el seguro es un contrato oneroso.

La prima constituye el precio del riesgo transferido, a partir de un triple ángulo: económico, jurídico y técnico (Fernandez Dirube A, 2012). Las primas de las tarifas de los diferentes ramos, representan el valor actual del riesgo futuro, medido de acuerdo con la experiencia estadística de la siniestralidad, teniendo en cuenta su frecuencia en el tiempo y su intensidad en los daños.

No sólo la prima es esencial para que exista el contrato de seguros, sino la propia actividad aseguradora, ya que el pago de la prima por parte del asegurado es lo que va a conformar el llamado ***"fondo de primas***", con el que luego el asegurador va a poder hacer frente a las indemnizaciones que deriven de los eventuales siniestros que ocurran en el futuro (López Saavedra D, 2003).

Es fundamental, para el asegurado y el asegurador, conocer y evaluar correctamente el valor de los bienes. De ello depende tanto la satisfacción y tranquilidad del asegurado como el equilibrio técnico del asegurador para ajustar sus cálculos y garantizar las coberturas. Al valor máximo de indemnización en caso de siniestro, previamente estipulado en las condiciones de la póliza, se lo denomina ***"suma asegurada"*** o ***"capital asegurado***".

Otro elemento esencial que integra el contrato de seguro es el "***interés asegurable***", el cual se manifiesta como el interés económico lícito de que un siniestro no ocurra (Traverso A, 2014). Este concepto es fundamentalmente jurídico, representando la relación de hecho o de derecho, que liga a una persona con un bien, y es susceptible de valoración patrimonial (Fernadenz Dirube, 2012). En seguros patrimoniales, el interés es la medida del daño indemnizable, rigiendo una regla proporcional, por la cual en caso de siniestro, la indemnización al asegurado guardará respecto del daño, la misma proporción que exista entre el valor asegurable del bien y la suma asegurada. El concepto de interés asegurable sólo rige respecto de los seguros patrimoniales y se aplica no sólo al valor de lo dañado, sino también a la situación jurídica del asegurado, respecto del bien afectado y de la medida de su interés asegurable respecto del mismo.

La prima queda sujeta al convenio de las partes, aunque en su determinación cobran marcado relieve ciertos elementos de juicio que hacen a la técnica aseguradora, y obligan en primer término al análisis de la distinción entre prima neta y prima bruta (Di Giorgio, A., 2006). La "***prima neta***" (o pura), apunta a la proporcionalidad del precio del seguro en relación con el riesgo de ocurrencia del siniestro, sin tomar en consideración cualquier tipo de recargos, comisiones, gastos, etc. En su cuantificación son elementos determinantes: la consideración del riesgo en una unidad de tiempo determinada; la suma asegurada; la duración del contrato y la tasa de interés que el asegurador calcula obtener de las sumas aportadas por los asegurados. Las primas netas, en su conjunto, deben conformar capitales necesarios para cubrir, según estimaciones técnicas y financieras, la totalidad de los siniestros a la postre verificados. La "***prima bruta***" (o comercial), comprende además del valor de riesgo (prima neta), otros factores externos a la probabilidad de los siniestros, pero que igualmente pesan sobre el presupuesto del asegurador, como los recargos, gastos de producción y administración, la carga fiscal que debe soportar el asegurador, reservas y comisiones. A la prima bruta también se la denomina "***premio***", y por tratarse del propio objeto de

la prestación, se debe encontrar determinado en el título de la obligación, o bien ser determinable mediante referencia a las tarifas del asegurador o de plaza.

La prima es en principio invariable, pero se puede modificar cuando la ley confiera facultades a las partes, o por variación del riesgo o del interés asegurable. Se puede proceder a un ajuste por disminución del riesgo, en cuyo caso el asegurado tiene derecho a hacer rectificar la prima por los períodos posteriores, de acuerdo con la tarifa aplicable al tiempo de la denuncia de la disminución. La misma facultad le compete en caso de haber denunciado erróneamente un riesgo más grave, de acuerdo con la tarifa al tiempo de celebración del contrato y por los tramos ulteriores a la denuncia del error (art. 34, Ley de Seguros). Esta situación configura las que se denominan ***"anulaciones***" que se instrumentan a través de endosos de devolución (o notas de crédito).

A su vez, en caso de agravación del riesgo, si el asegurador no opta por la rescisión del contrato, ésta es improcedente, corresponde el reajuste desde la denuncia y según la tarifa aplicable en ese momento (art. 34, Ley de Seguros). En este caso, se facturarán premios "***adicionales***" por endosos o suplementos de la póliza (o notas de débito).

La prima puede ser única o periódica. La prima única importa una sola prestación, que representa el valor total del riesgo, por la duración del seguro que normalmente suele celebrarse por períodos anuales. Esto no significa que no haya períodos más cortos al que tradicionalmente conocemos como de vigencia del contrato de seguro; períodos que tienen que ver con el tiempo que abarca cada una de las primas que se obliga a pagar el asegurado, aunque la Ley de Seguros presume que la vigencia del mismo es anual e incluso establece que los efectos comienzan a las doce horas del día fijado como punto de partida de la vigencia y terminan a las doce horas del día fijado para la finalización de la misma, salvo la existencia de pacto en contrario.

La prima periódica viene constituida por una serie de prestaciones sucesivas e independientes, que se hacen exigibles en los distintos períodos en que se desmiembra la duración del contrato (Fernandez Dirube, 2012). De todos modos, en ambos casos el pago se puede subdividir en cuotas, mediante acuerdo de partes. En éste caso, se le agregará a la prima un ***"adicional financiero"*** cobrado en cada póliza con pago diferido (en cuotas) de la prima respectiva. Esto significa que, en caso de pagarse al contado la prima, no corresponde cobrar el adicional financiero.

Sin perjuicio de ello, rige el principio según el cual la prima se debe pagar anticipadamente (Di Giorgio, A., 2006). Es decir el pago se tiene que hacer por adelantado. Tanto el de la prima única: al celebrarse el contrato; como el de la periódica: al inicio de cada etapa. La ley también establece cuando se produce la mora en el pago de la prima y sus efectos, indicando que si el pago de la prima periódica o de la prima única no se efectuara oportunamente, el asegurador no será responsable por el siniestro ocurrido antes del pago (Traverso A, 2014).

2.1.4 Otros conceptos

Otros conceptos de interés que pertenecen al glosario de la industria aseguradora (Femández Dirube A, 1993):

- **Coaseguro:** se define como la figura del reparto de la asunción de un riesgo entre varios aseguradores, tomando cada uno a su cargo una porción del valor asegurado total. Varios aseguradores eligen libremente qué parte cubrir de un determinado riesgo y confeccionan cada uno un contrato de seguro individual por dicho valor, aunque todos los contratos concertados por los coaseguradores se instrumenten por medio de una sola póliza. No existe solidaridad entre los coaseguradores, por lo que el asegurado, en caso de siniestro, debe cobrar individualmente de cada coasegurador la porción de indemnización que le corresponda por cada uno.
- **Reaseguro:** supone la posibilidad de que el asegurador cubra a su vez sus riesgos cuando superan determinado límite técnico con otro "asegurador" (el reasegurador). El reaseguro es un seguro de "segundo grado" pero no deja de ser parte del negocio asegurador. La compañía aseguradora establece una relación con el reasegurador para asegurar ese excedente de riesgo.
- **Retrocesión:** esta figura se da en el marco de compañías reaseguradoras. Una vez que el reasegurador ha establecido su capacidad en función al análisis cuantitativo y cualitativo de su cartera, procede a reasegurar los excedentes que se le hayan producido, a través de contratos generales de retrocesión.

2.1.5 Reservas

Las reservas técnicas se entienden como compromisos que deben ser reflejados en el Pasivo contable de la entidad aseguradora, en el rubro Deudas del balance de publicación, y se utilizan para hacer frente a las obligaciones y afrontar desviaciones, tomando en cuenta el valor del dinero en el tiempo, la inflación y los costos de administración y manejo de cartera, incluyendo el reaseguro (Di Giorgio, A., 2006). Cada país establece lo tipos y requisitos que debe cumplir el cálculo de cada una de las reservas que deben contemplarse en las situaciones patrimoniales de la empresa.

En nuestro país, la Superintendencia de Seguros de la Nación (SSN) establece en el Artículo 33 de su Reglamento General de la Actividad Aseguradora, los siguientes tipos de reservas: Riesgos en Curso, Insuficiencia de Primas, Siniestros Pendientes, Riesgos del Trabajo, y otras Reservas Especiales.

Uno de los tipos de reservas más importante es la ***"Reserva de Riesgos en Curso (RRC)"***, que representa la porción de la prima no devengada al cierre del ejercicio (Di Giorgio, A., 2006). Pensando en un contrato de seguros anual, constituye ésta reserva la parte no transcurrida dentro del ejercicio que pasa al próximo. Éste tipo de reserva tiene como objetivo hacer frente a los riesgos que permanecen en vigor al cierre contable de un ejercicio económico. Tiene su origen en el hecho de que las pólizas de seguro se renuevan anualmente en el mismo día y mes que se suscribió, y la aseguradora cobra dicha primas en los respectivos vencimientos anuales.

Éste sistema permite distribuir uniformemente los ingresos de la empresa a lo largo de todo el año, pero origina la necesidad de consumir al final de cada ejercicio una provisión con que hacer

frente a la posibles siniestros que ocurran en el año siguiente y que afecten a pólizas respecto a las cuales ya se ha satisfecho toda la prima de un año (Mapfre, 1990).

Su cálculo debe incluir el riesgo no corrido en el ejercicio, calculado en base a un sistema de diferimiento denominado "***póliza por póliza***". A tales efectos, se considera el monto de las primas por seguros directos emitidas, netas de anulaciones, por contratos de seguros cuyo vencimiento de vigencia opere con posterioridad a la fecha de cierre del ejercicio.

La ***"Reserva técnica por insuficiencia de primas"*** debe calcularse para cada ramo en que opere la empresa aseguradora, de acuerdo con la diferencia entre los importes correspondientes a seguros directos, reaseguros activos y/o retrocesiones.

Otro tipo de reserva importante es la ***"Reserva de Siniestros Pendientes"***, que refleja los siniestros impagos a la fecha de cierre del ejercicio (Di Giorgio, A., 2006). Asimismo, se incluyen en este concepto los pasivos constituidos con el propósito de integrar y ajustar los siniestros impagos a la fecha de cierre, como ser: desvíos de siniestralidad, contingencias, insuficiencia de valuación, siniestros pendientes y los siniestros ocurridos y no reportados, que implica la constitución de un pasivo correspondiente a futuros pagos por siniestros ocurridos y no denunciados o reportados y siniestros ocurridos y denunciados pero reservados en forma insuficiente.

De ésta manera, al cierre de cada ejercicio o período, las aseguradoras deben estimar los siniestros pendientes de pago a dicha fecha. A tales efectos, deben arbitrar todos los medios necesarios para que las carpetas de siniestros cuenten con todos los elementos indispensables para efectuar su correcta valuación (copia de la demanda y su contestación, informes periódicos de los asesores legales sobre el estado de los juicios pendientes, informes médicos sobre las posibles incapacidades en los siniestros pendientes de Accidentes del Trabajo, informes de inspectores de siniestros y presupuestos de talleres, informes de peritos tasadores, etc.).

Así como la SSN reglamenta las reservas generales, las aseguradoras que celebren contratos cuyo objeto sea la cobertura del riesgo definido en las Leyes Nº 24.557 y Nº 26.773 relacionados con los contratos de trabajo, la SRT indica que deben constituir adicionalmente reservas por siniestros pendientes, reservas por resultados negativos, reservas por contingencias y desvíos de siniestralidad, y reservas para la cobertura de las prestaciones dinerarias previstas en la legislación laboral para los casos de accidentes y enfermedades inculpables.

Por último, La SSN establece adicionalmente ***"Reservas especiales"*** como la Reserva por desvíos de siniestralidad para la cobertura de "Protección por Pérdida de Ingreso por Desempleo Involuntario o Invalidez Total y Temporaria", que tiene el objeto de hacer frente a resultados adversos que se produzcan específicamente por la operación de la cobertura en cuestión.

2.1.6 Estadística y Ciencia Actuarial

La estadística es una disciplina científica que se ocupa de la obtención, orden y análisis de un conjunto de datos con el fin de obtener explicaciones y predicciones sobre fenómenos observados (Roldán, 2017). Consiste en métodos, procedimientos y fórmulas que permiten recolectar información para luego analizarla y extraer de ella conclusiones relevantes. Se puede decir que es

la Ciencia de los Datos y que su principal objetivo es mejorar la comprensión de los hechos a partir de la información disponible.

Uno de los principales instrumentos que se utilizan en la estadística es la probabilidad (López, 2019). La teoría de la probabilidad es una herramienta matemática que establece un conjunto de reglas o principios útiles para calcular la ocurrencia o no ocurrencia de fenómenos aleatorios y procesos estocásticos. En otras palabras, está formada por un conjunto de técnicas que nos permiten asignar un número a la posibilidad de que un evento ocurra.

La ciencia actuarial utiliza éstas disciplinas para estudiar los riesgos financieros de las empresas, entre ellas las aseguradoras, a través de modelos matemáticos complejos y algoritmos que interpretan el funcionamiento de la economía a través de la probabilidad de ocurrencia de determinados sucesos (Vázquez Burguillo, 2016).

Se conoce con el nombre de actuario a la persona con título académico, profesionalmente capacitada para solucionar las cuestiones de índole financiera, técnica, matemática y estadística, relativas a las operaciones de seguros mediante la aplicación de la Ciencia Actuarial (Mapfre, 1990).

Los actuarios conocen perfectamente los sistemas informáticos y de gestión de riesgos, así como las variables que se incluyen en los modelos financieros y las pruebas que se realizan bajo situaciones de stress (Vázquez Burguillo, 2016). Dichos modelos tienen que ser realistas y con un porcentaje de acierto muy elevado. Además, se deben definir las formas de actuación, en caso falle el pronóstico, y los protocolos que permitan afrontar situaciones complejas fuera de los escenarios planteados.

La ciencia actuarial necesita estar en constante evolución. Esto, ya que las condiciones de mercado cambian con el paso de las experiencias sufridas en épocas de crisis. Por un lado, los modelos deben centrarse en la realidad económica y no solamente basarse en pruebas del pasado. Así, deben anticipar posibles resultados futuros y situaciones que puedan ocurrir en los próximos años para cubrirse y dotar las provisiones oportunas. Por otro lado, la analítica constante, los modelos deterministas, los test de stress y la cuantificación del volumen de riesgo con porcentajes de probabilidad son fundamentales para esta ciencia, así como la inversión en tecnología que permita acceder a información de redes complejas donde los modelos se componen de múltiples variables que hay tener en cuenta.

2.1.6.1 Estadísticas de una aseguradora

La actividad aseguradora ha tenido un origen puramente empírico, El seguro nació de la observación de determinados hechos que se producen con cierta frecuencia y grado de intensidad, dentro de una masa de personas o bienes, a quienes potencialmente los afectan (Fernandez Dirube A, 2012).

Pero el desarrollo de las ciencias matemáticas y estadísticas mencionado, fue dando fundamento científico a la observación empírica del comportamiento siniestral. A partir de

entonces, se va desarrollando el seguro sobre bases matemáticas y estadísticas, y se llegó así a medir la frecuencia e intensidad de determinados hechos dañosos, sobre grandes números de amenazados por determinado tipo de daño.

A partir de estos principios científicos, se define la base matemática y estadística del seguro, estableciendo que determinados hechos que provocan cierto tipo de daños, se producen con determinada frecuencia en el tiempo y medible intensidad en sus efectos dañosos, en un gran número de casos. Se trata de la denominada "Ley de los grandes números", que es la base matemática y estadística del seguro.

Algunos instrumentos estadísticos utilizadas en las aseguradoras para el desarrollo de su actividad (Mapfre, 1990):

- **Masa Asegurable:** el volumen de los riesgos asegurados debe ser lo suficientemente amplio para dar solidez técnico-actuarial a su actividad aseguradora.
- **Bases técnicas:** representan los cálculos actuariales que dan origen a la determinación de las primas y recargos que va a aplicar la aseguradora. Normalmente la primar está integrada por el índice de siniestralidad (frecuencia más costo promedio de los siniestros), los índices de gastos de administración y producción, los factores de corrección y seguridad, y el beneficio de explotación.
- **Tablas de mortalidad:** reflejan las posibilidades de fallecimiento de una colectividad de personas en función de los diferentes tipos de edades y el período de vida prolongado que se considere. Éste instrumento será la base para la fijación de las primas de las diferentes modalidades de seguro de vida.
- **Resultado técnico:** es el que proviene propia y exclusivamente del ejercicio de la actividad aseguradora, sin tener en cuenta otros ingresos y gastos que pueda tener la empresa ajenos a ésta actividad, como puede ser su gestión financiera. El resultado técnico es la diferencia entre las primas recaudadas y el importe de los gastos habidos por siniestros (pagados o pendientes de pago). Entre dichos gastos hay que incluir también comisiones, gastos de administración, reaseguro, etc.
- **Índice de Frecuencia**: es la cifra o coeficiente que refleja el promedio del número de siniestros que una póliza de seguros tiene durante un año completo o el promedio de siniestros por año de todo un conjunto o cartera de pólizas.
- **Índice de Intensidad:** es el coste promedio de los siniestros producidos respecto a un asegurado o conjunto de asegurados o con relación a una determinada cartera de pólizas.
- **Índice de Siniestralidad:** es el coeficiente o porcentaje que refleja la proporción existente entre el coste de los siniestros producidos en un conjunto o cartera determinada de pólizas y el volumen global de las primas que han devengado en el mismo período tales operaciones.
- **Siniestralidad:** en sentido amplio, es la valoración conjunta de los siniestros producidos (pendientes y liquidados) con cargo a una entidad aseguradora. En sentido más estricto, equivale a la proporción entre el importe total de los siniestros y las primas recaudadas por

una entidad aseguradora, proporción que se mide mediante el antes referido índice de Siniestralidad.

- **Siniestralidad esperada:** importe de los siniestros que, de acuerdo con experiencias anteriores, se calcula que deberán ser satisfechos.

En general las variables antes citadas son de gran utilidad para determinar si las tarifas son o no técnicamente suficientes, y proporcionan los datos oportunos para, si resulta necesario, proceder a la corrección de las mismas.

Como se mencionó anteriormente, la base fundamental del seguro es el riesgo, el cual se define en diferentes grados y la clave es saber mensurarlo (Fratta, 2016). No se evidencia en la práctica de la actividad aseguradora claridad sobre conceptos como: “lo probable y lo posible”. No se analiza la probabilidad próxima y remota, sino que se observa, en los diferentes estudios de riesgos, en donde algunos apuntan a una irremediable ocurrencia de siniestro, y otros hacen hincapié en medidas de seguridad que, de darse, no darían lugar a un siniestro.

2.2 MACHINE LEARNING

Desde que las primeras computadoras programables fueron concebidas, las personas se preguntaron si tendrían la capacidad de pensar, de aprender y de convertirse en "máquinas inteligentes" (Kramer, 2002). El campo de la ciencia que se encarga de resolver este interrogante se denomina inteligencia artificial. Se trata de un área multidisciplinaria, que a través de ciencias como las ciencias de la computación, la matemática, la lógica y la filosofía, estudia la creación y diseño de sistemas capaces de resolver problemas cotidianos por sí mismos, utilizando como paradigma la inteligencia humana.

Para que una maquina pueda comportarse de manera inteligente debería ser capaz de resolver problemas de la manera en que lo hacen los humanos (Aggarwal, 2015), es decir, en base a la experiencia y el conocimiento. Esto implica que debería ser capaz de modificar su comportamiento en base a cuan precisos son los resultados obtenidos comparados con los esperados.

A esa generación de resultados, basados principalmente en hallar estructuras y patrones en un compendio de registros digitales, se le denomina "***ciencia de los datos***" y, más concretamente, cuando de ello se derivan resultados aplicables a las decisiones empresariales, se habla de "**aprendizaje automatizado**" o "***machine learning***" (Michalski et al., 2013). Machine Learning de este modo es un área de la inteligencia artificial que engloba un conjunto de técnicas que hacen posible el aprendizaje automático a través del entrenamiento con grandes volúmenes de datos (Russo C et al, 2016). Hoy en día existen diferentes modelos que utilizan esta técnica y consiguen una precisión incluso superior a la de los humanos en las mismas áreas. La construcción de modelos de Machine Learning requiere adaptaciones propias debido a la naturaleza de los datos o a la problemática a la que se aplica. Así, surge la necesidad de investigar las diferentes técnicas que permitan obtener resultados precisos y confiables en un tiempo razonable.

Machine Learning está muy relacionado con otras tecnologías como "***Minería de Datos***" o "***Data Mining***", ya que ambas están arraigadas en la ciencia de la información, entrecruzándose y confundiéndose entre sí (Mayorga Muñoz, 2019). En realidad, la Minería de Datos y el Aprendizaje Automático se apoyan sobre la misma base, pero de diferentes maneras. Data Mining utiliza datos extraídos de la información existente para buscar patrones emergentes que puedan ayudar a moldear los procesos de toma de decisiones, en cambio Machine Learning, por otro lado, puede realmente aprender de los datos existentes y proporcionar la base necesaria para que una máquina se enseñe a sí misma. Si bien se puede configurar Data Mining para buscar automáticamente tipos específicos de datos y parámetros, no aprenden ni aplican el conocimiento por sí mismos sin la interacción humana. La minería de datos tampoco puede ver automáticamente la relación entre las piezas de datos existentes con la misma profundidad que el Aprendizaje Automático.

Por último, otro concepto relacionado es "***BigData***", que hace referencia a un conjunto masivo de datos que por sus características (volumen, velocidad y variedad) sólo pueden ser procesados con técnicas y herramientas muy específicas (Lazcano, 2019). BigData es útil en muchos casos para aplicar Machine Learning, porque para "entrenar" una Inteligencia Artificial hace falta una base de datos lo suficientemente grande como para que el programa pueda encontrar los patrones de

comportamiento subyacentes y hacer sus predicciones. Big Data puede generar entonces el conjunto masivo de datos para que Machine Learning los utilice para poner foco en el futuro, y centrarse en la predicción de tendencias.

2.2.1 Fases del proceso de Machine Learning:

A la hora de usar Machine Learning, es conveniente seguir un proceso para obtener buenos resultados y hacer un uso eficiente del tiempo. Además, representa una orientación sobre qué es lo que se debe hacer en caso de que los resultados no sean los esperados. El diagrama de la ***Figura 3.1*** está basado en el estándar CRISP-DM que describe un modelo estándar y abierto del proceso de desarrollo de un proceso de Machine Learning (Martínez Heras, 2019). Muestra las distintas fases del proceso y cómo interaccionan entre sí, y una ponderación del grado de dificultad en su realización.

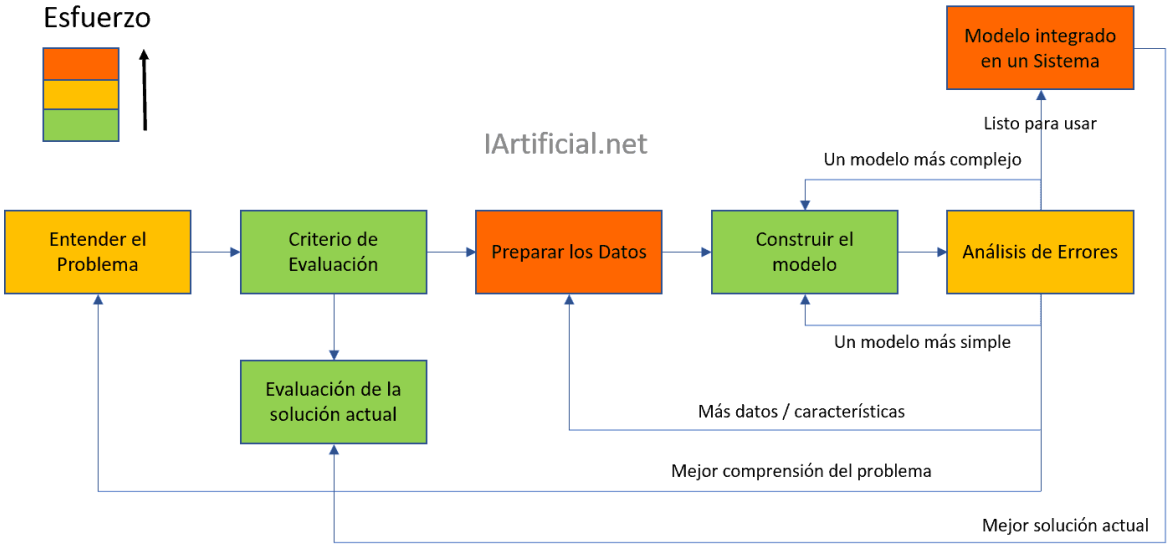


Figura 3.1 - Fases del Proceso de Machine Learning. Fuente: https://iartificial.net/fases-del-proceso-de-machine-learning/ (Martínez Heras, 2019)

- **Entender el Problema**: es muy importante entender el problema que se requiere resolver, el cual lleva tiempo sobre todo si proviene de un sector en el que el analista tiene pocos conocimientos. Ésta fase no solo incluye el entendimiento del problema, sino también los datos que se tienen disponibles. Es común hacer un análisis exploratorio de datos para familiarizarnos con ellos. En el análisis exploratorio se suelen hacer gráficos, correlaciones y estadísticas descriptivas para comprenderlos mejor, y estimar si son suficientes y relevantes para construir un modelo.

- **Definir un Criterio de Evaluación**: antes de pensar en utilizar un modelo de Machine Learning, es necesario definir cómo se va a evaluar, tratándose normalmente de una medida de error. (Martínez Heras, 2019). Típicamente en los problemas de regresión, en donde las predicciones son valores numéricos continuos, se usan las siguientes medidas (Channabasava Gola, 2018):

- ***Error Cuadrático Medio***: mide la diferencia cuadrática entre cada dato y su correspondiente predicción, para luego calcular su promedio en el indicador ***"MSE",*** y la raíz cuadrada de ese promedio en el indicador ***"RMSE",*** obteniendo así una medida de error del mismo orden de los datos. Ésta métrica si bien es muy útil porque potencia los errores en una escala cuadrática, al tener una naturaleza absoluta puede generar dificultades a la hora de sacar conclusiones cuando se comparan diferentes modelos.

- ***Error Logarítmico Medio***: es básicamente el RMSE pero calculado en base a una escala logarítmica obteniendo el indicador ***"RMSLE"***.

- ***R Cuadrado:*** calcula la relación entre el MSE y otro similar utilizando la media en lugar de la predicción. A diferencia de los anteriores, éste indicador tienen naturaleza relativa ya que realiza una comparación de las predicciones con una determinada línea base.

- ***Error Absoluto Medio:*** mide la diferencia absoluta entre cada dato y su correspondiente predicción, para luego calcular su promedio en el indicador ***"MAE"*** y su porcentaje para el indicador ***"MAPE".***

Y en problemas de clasificación, donde las predicciones son clases discretas, se utiliza el concepto de ***"Matriz de confusión"*** en donde se contabilizan las clasificaciones Verdaderas negativas, Falsas negativas, Verdaderas positivas, y Falsas positivas, para luego extraer las siguientes medidas:

- ***Exactitud:*** representa el porcentaje de clasificaciones pronosticadas correctamente, tanto positivamente como negativamente, o sea, la suma de los Verdaderos positivos más Verdaderos negativos sobre el total.

- ***Precisión:*** representa el porcentaje sólo de las clasificaciones Verdaderas pronosticadas correctamente, o sea, los Verdaderos positivos sobre el total de positivos.

- ***Exhautividad:*** representa el porcentaje de las clasificaciones Verdaderas positivas sobre el total de clasificaciones correctas, o sea, los Verdaderos positivos sobre la suma de Verdaderos positivos más Falsos negativos.

- ***AUC ROC:*** representa el área debajo de la curva ROC, la cual se obtiene trazando las tasas de Verdadero positivo (eje y) y Falso positivo (eje x).

- **Evaluación de la solución actual:** probablemente el problema que se intenta resolver con Machine Learning, ya se esté resolviendo de alguna forma. El hecho de medir el rendimiento de la solución actual (con el criterio de evaluación elegido), es una buena base para luego compararlo con el rendimiento del modelo de machine learning. Si no hay ninguna solución actual, una alternativa útil es definir una solución simple y muy fácil de implementar.

- **Preparar los datos:** la preparación de datos es de las fases del machine learning que supone los mayores esfuerzos. Es la más laboriosa y "time consulting" dado que con frecuencia los datos provendrán de distintas fuentes o bases de datos, y sus criterios de almacenamiento

también variarán en función de la propia evolución a la que se hayan visto sometidos éstos sistemas en la compañía (Alarcón Madrid L, 2018). En ésta fase los principales desafíos son:

- ***Datos Incompletos:*** es normal que no se tengan todos los datos que el analista les gustaría tener (campos vacíos, registros o archivos faltantes, etc.), por lo que se deberá tomar una decisión como puede ser: eliminarlos y dejar solo los datos completos, imputarlos con un valor razonable (un simple promedio por ej.), imputarlos con un modelo de Machine Learning que se use soporte para los casos más sofisticados, o incluso no hacer nada y usar alguna técnica de machine learning que pueda manejar datos incompletos.
- ***Combinar datos de varias fuentes:*** es muy común la necesidad de combinar datos de diferentes fuentes, ya que algunos datos pueden venir de una base de datos, otros de una hoja de cálculo, de ficheros, etc.. La combinación es necesaria para que los algoritmos de machine learning puedan considerar toda la información.
- ***Darle el formato adecuado a los datos:*** se requiere darle el formato que requieren las librerías de Machine Learning que se estén utilizando. En general, estas librerías esperan que los datos tengan forma de matriz o de tensor. Un tensor es una generalización de una matriz.
- ***Calcular características relevantes (features):*** los algoritmos de machine learning funcionan mucho mejor si se les ofrece características relevantes en lugar de los datos puros. De la misma forma, es muy útil transformar los datos para hacer la tarea de aprendizaje más fácil. Ésta fase incluye la tarea de pensar en qué características van a ser más relevantes para solucionar el problema y probarlo.
- ***Normalización de datos:*** en muchos casos es útil normalizar los datos, que implica poner a todos los datos en una escala similar. De ésta manera los algoritmos pueden funcionar mejor y más rápido (generalmente trabajan mejor con valores chicos).
- ***Estandarización de datos (manejo de outliers):*** un valor atípico es un punto de observación que está distante del resto de las observaciones, y surge debido a cambios en el comportamiento del sistema, comportamiento fraudulento, error humano, error del instrumento o simplemente a través de desviaciones naturales de las poblaciones (Swalin, 2018). Una muestra puede haberse contaminado con elementos externos de la población que se está examinando. Muchos modelos de aprendizaje automático, como la regresión lineal y logística, se ven fácilmente afectados por los valores atípicos. Para solucionar este problema, se puede cambiar el modelo, las métricas, o se pueden hacer algunos cambios en los datos para manejarlos. Existen muchas técnicas que se pueden aplicar en Machine Learning para detectar los outliers como BoxPlot, Distancia de Cook, Z-Core, y otras para tratarlos como la simple eliminación, winsorizing, o transformaciones Log-Scale.

- **Construir el modelo:** la fase de construir un modelo de machine learning, una vez que se tengan los datos preparados, requiere sorprendentemente poco esfuerzo. Esto se justifica en que existen muchas librerías de machine learning disponibles, y muchas de ellas son gratuitas y de código abierto. Durante esta fase se tiene que elegir qué tipo de técnica de machine

learning se va a usar. El algoritmo de machine learning aprenderá automáticamente a obtener los resultados adecuados con los datos históricos que hemos preparado, desprendiéndose un determinado nivel de error (Martínez Heras, 2019). Una vez determinados los algoritmos y librerías a utilizar, el siguiente paso es el ***"entrenamiento"***, que implica ejecutar los algoritmos inicializados con algunos valores que pueden ser aleatorios, lo que genera el modelo matemático de inferencia (generalmente es una fórmula con coeficientes y variables) que tendrá la misión de predecir una salida a partir de esos valores iniciales. Al principio el error será grande, pero a través de la comparación de la predicción con los resultados correctos, el modelo es capaz de ajustar los pesos, coeficientes y sesgos hasta conseguir un buen modelo de predicción. El proceso se repite iteración tras iteración, y en cada una de ella el nivel de error se va acercando al ideal (Roman, 2019). Durante el entrenamiento el conjunto de datos se divide en tres grupos:

  - ***Conjunto de Entrenamiento:*** son los datos con los que se construye el modelo. Normalmente representa entre el 50% y el 70% de la muestra.
  - ***Conjunto de Validación:*** es una porción de datos que se usa para validar el modelo y calcular el error durante el proceso. Representa entre el 15% y el 25% de la muestra.
  - ***Conjunto de Prueba:*** es una última porción que se mantiene aparte y sobre la cual se evalúa el modelo una vez entrenado. Usualmente se reporta con ellos las métricas finales. Representa entre el 15% y el 25% de la muestra.

- **Análisis de Errores:** la fase del análisis de errores requiere un esfuerzo relativo medio y es importante para entender qué nivel de efectividad tiene el modelo y cuál es el siguiente paso para mejorar los resultados. Aquí se utilizan las métricas y los criterios de evaluación definidos en la fase 2 del proceso. Para mejorar resultados se pueden tomar algunas de éstas decisiones: usar un modelo más complejo (con más variables), usar un modelo más simple (simplificación o regulación), aumentar o mejorar los datos, buscar nuevas características, o desarrollar una mejor comprensión del problema. El objetivo de ésta fase es asegurar que el modelo sea capaz de generalizar, que es la capacidad que tienen los modelos de machine learning de producir buenos resultados cuando se usan datos nuevos. Si no se consiguen resultados satisfactorios, se itera sobre las fases anteriores las veces que sea necesario.

- **Modelo integrado en un Sistema:** una vez que se esté satisfecho con el error y el nuevo modelo es lo suficientemente mejor que el proceso actual, el último paso es integrarlo a los sistemas empresariales del resto de la compañía. Ésta tarea de integrar un modelo de machine learning en un sistema requiere un esfuerzo relativo mayor, necesitándose llevar adelante tareas como: repetir de forma automática las fases de preparación de datos, hacer que el modelo de machine learning se comunique con otras partes del sistema tanto para la obtención de datos como para el uso de los resultados, monitorear automáticamente los errores del modelo y comunicar si los errores crecen con el tiempo para re-construirlo y reiniciar las fases anteriores. El mayor esfuerzo radica en la sistematización de esas interfaces de datos que tanto tiempo consumió en la etapa de preparación, pero es esencial para que el machine learning y la inteligencia artificial sean útiles.

2.2.2 Clasificación de Algoritmos de Machine Learning

Uno de los criterios más generales por el que se pueden clasificar los algoritmos de Machine Learning es si van a resolver problemas de ***"regresión"*** o de ***"clasificación"***, cuya diferencia está en el tipo de resultado que se obtiene de la técnica de machine learning.

- **Regresión:** el resultado de ésta técnica es un número, es decir, un valor numérico dentro de un conjunto infinito de posibles resultados. Se puede utilizar por ejemplo para predecir por cuánto se va a vender una propiedad inmobiliaria, predecir cuánto tiempo va a permanecer un empleado en una empresa, estimar cuánto tiempo va a tardar un vehículo en llegar a su destino, o estimar cuántos productos se van a vender.

- **Clasificación:** el resultado de ésta técnica es una clase, entre un número limitado de clases disponibles. Las clases se refieren a las categorías arbitrarias definidas según el tipo de problema. Se puede utilizar en clasificaciones binarias donde sólo hay 2 clases como por ejemplo detectar si un correo es spam o no (spam o no-spam), si un cliente comprará un producto (sí, no), determinar un tipo de tumor (maligno, benigno), si subirá el índice bursátil mañana (sí, no), o en clasificaciones con más clases como un reloj que determine qué deporte está haciendo la persona que lo porta (caminar, correr, bicicleta, nadar), o reconocer objetos en imágenes digitalizadas (perro, gato, coche, avión, globo, manzana). Muchos algoritmos de machine learning dan los resultados de clasificación con probabilidades, por ejemplo, que un correo es spam con una probabilidad del 89%, o que una imagen tiene un 67% de probabilidades ser un perro, un 18% de ser un gato, un 9% de ser una oveja, etc..

Otro criterio para clasificar los algoritmos de Machine Learning es dependiendo del tipo de datos que requiere para funcionar (Marsland, 2015):

- **Algoritmos supervisados**: estos algoritmos utilizan un conjunto de datos de entrenamiento etiquetados (preclasificados), los cuales procesan para realizar predicciones sobre los mismos, corrigiéndolas cuando son incorrectas. El proceso de entrenamiento continúa hasta que el modelo alcanza un nivel deseado de precisión.

- **Algoritmos semi-supervisados:** combinan tanto datos etiquetados como no etiquetados para generar una función deseada o clasificador. Este tipo de modelos deben aprender las estructuras para organizar los datos así como también realizar predicciones.

- **Algoritmos no supervisados:** el conjunto de datos no se encuentra etiquetado y no se tiene un resultado conocido. Por ello deben deducir las estructuras presentes en los datos de entrada, lo puede conseguir a través de un proceso matemático para reducir la redundancia sistemáticamente u organizando los datos por similitud.

Por último, otro criterio para clasificar los algoritmos es dependiendo del tipo de técnica que utiliza para funcionar, y entre los más relevantes encontramos:

- **Regresión Lineal:** es el método clásico para datos en los que se desea explicar una variable dependiente en función del resto. Su principal característica es la simplicidad e interpretabilidad (Guillen M, Pesantez Narvaez, 2018). Su objetivo es minimizar la suma de

errores cuadrados considerando las desviaciones de los valores medios y tomados estos como la diferencia entre la observación y una combinación lineal de factores. La inferencia de este modelo se realiza en base a suponer que la variable respuesta sigue una distribución normal.

- **Regresión Logística:** opera de manera similar a la regresión lineal al encontrar los valores de los coeficientes que multiplican al valor de cada variable explicativa. A diferencia de la regresión lineal, la predicción de la respuesta se transforma utilizando una función no lineal, de modo que los resultados se pueden interpretar como una puntuación o una probabilidad estimada. Es una técnica fundamental en muchas áreas aplicadas, tales como pruebas de medicamentos, calificación crediticia, análisis de fraude y un gran número de situaciones en los que interviene una clasificación. Hoy en día, es el método de referencia para problemas de clasificación binaria. Existen variaciones del algoritmo estándar de Regresión Logísticas como Ridge y Lasso (Pereira, Basto, Ferreira da Silva, 2015).

- **Support Vector Machines (SVM):** busca la maximización de la distancia entre la recta o el plano (denominado hiperplano) y las muestras que se encuentran a un lado u otro, y encuentra los coeficientes que dan como resultado una mejor separación. La distancia entre el hiperplano y los puntos de datos más cercanos se denomina margen. El hiperplano mejor u óptimo que puede separar las dos clases es la línea que tiene el margen más grande. A los puntos más cercanos se les llama vectores de soporte, ya que apoyan o definen el hiperplano. En el caso que las muestras no sean linealmente separables se utiliza una transformación llamada kernel (Burges, 1998) (Xindong Wu et al, 2008). Éste tipo de algoritmos se realizan a través del análisis del pasado, tratando de reproducir la respuesta o cómo anticipar lo sucedido (Kotsiantis, 2007).

- **Árboles de decisión (clasificadores y de regresión):** los árboles son algoritmos que actúan como modelos predictivos en machine learning y admiten una representación gráfica del modelo de árbol de decisión, que muestra cómo los datos se dividen paso a paso, en base a qué factor explicativo y cuál es el orden secuencial de la ordenación organizada (Guillen M, Pesantez Narvaez, 2018). Cada nodo representa una sola variable de entrada y un punto de división en esa variable cuando la variable es cuantitativa. Los nodos de la hoja del árbol contienen una variable de salida que se usa para hacer la predicción.

- **Naive Bayes:** este procedimiento funciona como un procedimiento directo de modelización predictiva. El modelo se compone de dos tipos de probabilidades que se pueden calcular directamente a partir de una muestra de entrenamiento: 1) La probabilidad de pertenencia a cada una de las clases y 2) la probabilidad condicional de ocurrencia del suceso de interés para cada clase, que da lugar a un valor predictivo. Una vez calculado, el modelo de probabilidad se puede usar para hacer predicciones para nuevos datos utilizando el Teorema de Bayes. Asumir que cada variable de entrada es independiente y normalmente distribuida es una suposición fuerte y es bastante inviable para datos reales, sin embargo, la técnica es muy efectiva en una amplia gama de problemas complejos.

- **K-Nearest Neighbors:** las predicciones sobre éste método se hacen buscando a través de todo el conjunto de entrenamiento cuáles son los K casos más similares (los vecinos) y resumiendo la variable de salida para esos casos. Para los problemas de regresión, esta podría ser la media

de la variable de salida, para los problemas de clasificación este podría ser el valor más común. En este método es fundamental determinar qué medida se empleará para medir la distancia entre las características de los datos. El enfoque más simple es encontrar la distancia euclídea, pero esto puede requerir una gran cantidad de memoria o espacio para almacenar todas las similitudes de datos y debe actualizarse cuando entre un nuevo caso. La idea de distancia o cercanía acaba provocando muchas dificultades en dimensiones muy altas (muchas variables de entrada) que pueden afectar negativamente el rendimiento del algoritmo. Esto se conoce como la maldición de la dimensionalidad, que sugiere usar aquellas variables de entrada que son más relevantes para predecir la variable de salida.

- **Bootstrap y Bagging:** es un algoritmo de machine learning potente que se basa en Bootstrap Aggregation o bagging. Bootstrap toma muestras de los datos, calcula el valor de interés (o variable de respuesta) y luego promedia todos los valores para obtener una mejor estimación del valor real. El método bagging tiene el mismo enfoque, pero se usa para estimar modelos estadísticos completos en lugar de valores únicos, como lo hacen comúnmente los árboles de decisión. Los modelos creados para cada muestra de datos son, por lo tanto, más diferentes de lo que serían de otra manera, pero igual de precisos. La combinación de sus predicciones da como resultado una mejor estimación del verdadero valor de salida.

- **Bosques aleatorios (Random Forest):** un Random Forest es un conjunto (ensemble) de árboles de decisión combinados con bagging (Martínez Heras, 2019). Al usar bagging distintos árboles ven distintas porciones de los datos. Ningún árbol ve todos los datos de entrenamiento, lo que hace que cada uno se entrene con distintas muestras de datos para un mismo problema. De esta forma, al combinar sus resultados, unos errores se compensan con otros y se obtiene una predicción que generaliza mejor.

- **Boosting y AdaBoost:** Boosting es una técnica que intenta crear un clasificador fuerte a partir de una serie de clasificadores débiles mediante la construcción de un modelo a partir de los datos de entrenamiento, y luego la creación de un segundo modelo que intenta corregir los errores del primer modelo (Guillen M, Pesantez Narvaez, 2018). Los modelos se agregan hasta que el conjunto de entrenamiento se predice perfectamente o se agrega un número máximo de modelos. AdaBoost y Discrete Adaboost fueron los primeros algoritmos realmente exitosos desarrollados para la clasificación binaria. Para el caso de GBM (generalized boosting regression method), la función link es logit y se introduce para predecir variables de tipo binario.

- **Redes Neuronales:** Una red neuronal se puede definir como un sistema que permite establecer una relación entre entradas y salidas inspiradas en el sistema nervioso y diferenciándose de la computación tradicional, ya que estos no utilizan una algoritmia secuencial (Acevedo et al., 2017). Las redes neuronales artificiales se comportan como un cerebro humano, en donde se procesa la información en paralelo, con la posibilidad de aprender y generalizar situaciones no incluidas en procesos de entrenamiento, y realizar predicciones en sistemas relacionales no lineales. Según su naturaleza, existen muchos tipos de redes neuronales: monocapa, multicapa, convulsionales, recurrentes, radiales, etc.. Existen redes que utilizan tanto modelos de aprendizajes supervisados (por corrección de errores y estocásticos) como no supervisados (hebbiano, competitivo, comparativo y por refuerzo). Por

otro lado existen redes neuronales especializadas, que tienen como objetivo aplicarse en situaciones bien concretas, como por ejemplo las Redes “Long Short-Term Memory” (LSTM), que se especializan en problemas de análisis de secuencias de datos, como por ejemplo las series temporales (Brownlee, 2017). En éste último caso de estudio, deben manejar cuestiones como tendencia y estacionalidad, y sus resultados y capacidades de predictivas se contrastan con técnicas estadísticas tradicionales como los modelos ARIMA (Autoregressive Integrated Moving Average) (Lopez Briega, 2016).

- **Deep Learning:** se trata de un tipo complejo de red neuronal, en donde las neuronas artificiales se entrelazan en muchos y profundos niveles de jerarquía (Bengio, Goodfellow, Courville, 2017). Éstos algoritmos son capaces de hacer representaciones abstractas de la información, consiguiendo el aprendizaje automático no supervisado.
- **Active Learning:** es un caso especial de aprendizaje semi-supervisado donde el algoritmo de aprendizaje puede interactuar con un usuario u otra fuente de información para obtener los resultados deseados (Burr, 2014).

2.3 APLICACIONES DE MACHINE LEARNING EN LA INDUSTRIA ASEGURADORA

La creciente disponibilidad de grandes bases de datos ha fomentado el desarrollo de algoritmos y procedimientos dedicados al análisis de volúmenes de información que hace unas décadas eran inimaginables (Guillen M, Pesantez Narvaez, 2018). El entorno asegurador se ha visto involucrado en este proceso, dado que no solo ha aumentado la capacidad de almacenaje y procesamiento de los registros sobre los asegurados, sino que además se han creado nuevas metodologías a fin de extraer conocimiento de dichos datos de una forma sistemática.

2.3.1 Un sistema interactivo de detección de fraudes y abusos en seguros de salud, basado Machine Learning

En el año 2015, lker Kosea, Mehmet Gokturkb y Kemal Kilicc presentaron un trabajo sobre la detección de casos fraudulentos y abusivos en la atención médica. El propósito de su estudio fue, sin acoplarse a los actores principales de los sistemas de salud como médicos, pacientes o enfermedades específicas, implementar y evaluar un modelo novedoso de detección de fraudes independientemente de esos actores y productos involucrados en los reclamos, y plantearon una estructura extensible para introducir nuevos tipos de fraudes y abusos.

La utilización del concepto de Machine Learning Interactivo (IML), que se caracteriza por la incorporación de personas expertas durante el proceso de desarrollo de los modelos, le permitió incorporar conocimiento experto en un entorno no supervisado. Otros métodos que utilizaron en su proyecto fue el de comparación por pares del procesamiento jerárquico analítico (AHP), que es una herramienta que ayuda al proceso decisorio y que permiten elegir una entre múltiples alternativas, utilizado por los autores para ponderar a los actores y atributos. Otra técnica utilizada fue Maximización de Expectativas (EM), algoritmo usado en estadística para encontrar estimadores de máxima verosimilitud de parámetros en modelos probabilísticos que dependen de variables no observables, utilizados para agrupar actores similares. También incorporaron herramientas de Datawarehouse para cálculos de riesgo proactivos, de visualización para un análisis efectivo, y de estandarización z-cores para calcular los riesgos.

Los expertos participaron en todas las fases del estudio, y produjeron seis tipos diferentes de comportamientos anormales utilizando guiones gráficos. El modelo propuesto se evaluó con datos de elaboración de recetas en la vida real, cubriendo todos los actores y productos relevantes.

El modelo desarrollado se caracterizó por ser independiente del actor y de los productos básicos, siendo configurable y fácilmente adaptable en el entorno dinámico de fraude y conductas abusivas, ya que tuvo la característica de manejar de manera efectiva la naturaleza fragmentada de las conductas anormales.

Su propuesta combinó el análisis proactivo y retrospectivo con una herramienta de visualización mejorada que redujo significativamente los requisitos de tiempo para el proceso de búsqueda de hechos después de que el sistema detectó reclamos riesgosos, para después ser utilizado para producir informes mensuales que sean evaluados por compañías de seguros.

Por último, incorporaron un motor que también consideró las relaciones entre los actores, es decir, el riesgo de red de los actores (los actores que están relacionados con actores más riesgosos se vuelven más riesgosos y los relacionados con actores menos riesgosos se vuelven menos riesgosos) siendo considerado como una extensión adicional del modelo.

Los autores manifestaron que el modelo desarrollado fue la única aplicación que incluyó a casi todos los actores y productos básicos en el dominio del seguro de atención médica, y proporcionó una solución para satisfacer la demanda de una herramienta de soporte de decisiones que asigne medidas de riesgo para reclamar transacciones.

2.3.2 Machine Learning y Modelización Predictiva para la Tarifación en el Seguro de Automóviles

Montserrat Guillen y Jessica Pesantez-Narvaez en el año 2018 publicaron un artículo donde exploraron diversas aproximaciones a la predicción de la siniestralidad y las primas del ramo del automóvil, comparando su implementación en una muestra real, dividida aleatoriamente en muestras de entrenamiento y test. En su trabajo propusieron medidas para ayudar en la valoración de los métodos y su implicación práctica para la predicción de eventos pocos frecuentes y el cálculo de primas.

En dicho artículo abordaron qué potencial de aplicación tienen los métodos estándares del machine learning en la tarificación de seguros. Eligiendo una muestra real de pólizas del automóvil, analizaron la capacidad predictiva de distintas aproximaciones, y estudiaron cómo se pueden conseguir mejoras tanto en la precisión como en la robustez de la tarificación. Precisión, en el sentido de poder calcular una prima más acertada para cada asegurado, con un intervalo de confianza más reducido y menores tasas de error. Robustez, para que dichas predicciones sean lo más estables posibles cuando se produzcan modificaciones en la cartera, tales como algún siniestro de coste muy elevado, por ejemplo.

De los resultados obtenidos, corroboraron que los modelos predictivos clásicos tienen la ventaja de proporcionar una interpretabilidad directa e intuitiva, es decir, se puede cuantificar fácilmente el impacto de cada factor de riesgo en el cálculo de la prima y justificar si cada uno de los factores juega a favor o en contra de un precio final. Sin embargo, pese a que tradicionalmente se acepta que los métodos de la inteligencia artificial son cajas negras, algunas de las nuevas metodologías no resultan ser tan crípticas como se creía en sus inicios y pueden suponer una nueva forma, mucho más ajustada, de obtener tarificadores en el entorno del machine learning (Witten et al., 2016).

El elemento más innovador de su presentación es el proporcionar medidas de comparación de la capacidad predictiva de los métodos, viendo además cómo abordar el cálculo de la prima bajo diferentes perspectivas y señalando las diferencias según el método de cálculo de la prima pura que se haya empleado. Además, mediante un análisis comparativo, se obtienen resultados sobre la estabilidad de la capacidad predictiva y el impacto en el precio final.

Utilizaron un total de nueve algoritmos diferentes: Regresión Lineal, Regresión Logística, Árbol de Decisión, SVM (Support Vector Machine), Naive Bayes, kNN (K-Nearest Neighbors), Random Forest y dos algoritmos de Gradient Boosting (el GBM o Generalized Boosted Regression Modelling y el Discrete Adaboost). Luego compararon el resultado predictivo que proporciona cada uno utilizando como medida el RMSE (root mean squared error). En la comparativa de los métodos, valoraron si cada uno de los procedimientos logra mejor o peor el objetivo de predecir si se produce o no un siniestro.

Una de sus principales conclusiones, fue que existen grandes similitudes entre los diferentes métodos en cuanto a la capacidad de predecir la siniestralidad en una muestra de pólizas del seguro de automóviles, y que la decisión sobre cuáles implementar depende más del deseo de interpretabilidad, reproducibilidad y elementos de la supervisión, que de una ganancia excesiva en la precisión y robustez de cada uno de ellos al implementarlos en la muestra de test.

En cuanto a las primas, concluyeron que la dispersión de precios, y concretamente la diferencia entre la prima pura máxima y mínima, puede llegar a ser muy diferente según el método predictivo utilizado, por lo tanto la decisión de qué método utilizar puede incidir en las políticas comerciales que quieran emprenderse.

2.3.3 Estimación de la rentabilidad del cliente utilizando Big Data: un estudio de caso de la industria de seguros

En el año 2016 Kuangnan Fang, Yefei Jiang, y Malin Song publicaron un artículo en donde propusieron un nuevo método de estimación de rentabilidad de clientes para la industria del seguro.

Su trabajo partió desde el reconocimiento general de que mantener los clientes es más rentable que buscar nuevos. Sin embargo, no todas las relaciones con clientes valen la pena mantener, y en lugar de tratar todos los clientes por igual, resulta más efectivo desarrollar estrategias específicas para clientes específicos.

La rentabilidad del cliente es una de los más importantes factores de segmentación para distinguir clientes valiosos y no valiosos definida como la contribución monetaria realizada por un cliente a una organización (Mulhern, 1999).

Con la llegada de la rentabilidad del cliente, en la práctica del marketing tradicional se ha reconsiderado al cliente como un tipo de activo análogo a otras unidades económicas. Por lo tanto, las decisiones de marketing son las mismas que las decisiones de inversión en las que los ingresos esperados son evaluados. La conocida regla del 80/20 indica que para la mayoría de las empresas, las ganancias provienen principalmente de un pequeño conjunto de clientes. En consecuencia, el conocimiento de la rentabilidad del cliente puede mejorar la toma de decisiones en marketing y proporcionar una métrica para la asignación de fuentes de comercialización a clientes y mercado segmentados. Es posible que las empresas inviertan en los clientes que son valiosos pero minimicen las inversiones en los que no lo son.

En una industria orientada al cliente como el motor y la base del desarrollo, la investigación sobre los clientes en la industria del seguro es de gran importancia ya que, en un mercado competitivo, son libres de elegir sus aseguradoras (Peng et al., 2007). Se necesita saber qué clientes se van y por qué, y si merece los esfuerzos para retenerlos o no. Afortunadamente, la disponibilidad de los datos efectivos que registra detalles de cada transacción financiera y siniestro permite a los tomadores de decisiones obtener la valiosa información a nivel del cliente. Con información sobre la rentabilidad del cliente, el comportamiento del cliente y sus preferencias, los gerentes pueden tomar decisiones a largo plazo para producir una mezcla de clientes que generará la mayor rentabilidad.

En la industria de seguros, en donde las primas de los clientes son las principales fuentes de ganancias y la liquidación de sus siniestros son los principales gastos, los clientes se pueden dividir en cuatro tipos diferentes: alto beneficio y alto riesgo, alto beneficio y bajo riesgo, bajo beneficio y alto riesgo también como bajo beneficio y bajo riesgo. Para las compañías de seguros, el mejor cliente es el de tipo de alto beneficio y bajo riesgo. Sin embargo, dividiendo clientes con solo un aspecto de ganancia o riesgo inevitablemente conducirá a algunos sesgos, por lo que los autores en dicho trabajo introdujeron el concepto de fondos de reserva de rentabilidad para construir una rentabilidad del cliente de seguros (ICP) dirigida a la industria de seguros basado en la rentabilidad tradicional del cliente (CP). El tradicional CP refleja las primas y los eventos de siniestros incurridos, sin embargo, el ICP propuesto en su trabajo no solo usa la información de CP sino también considera los futuros eventos de siniestros.

En ese contexto, si bien el cálculo de la rentabilidad histórica ha sido un concepto estudiado mucho, los autores consideraron que es raro predecir la rentabilidad futura y vieron en esa tarea un verdadero desafío. Considerando las características de la industria de seguros, su trabajo propuso un nuevo método de cálculo de la rentabilidad del cliente para industria de seguros, con consideración de la reserva de rentabilidad definida anteriormente. Su objetivo fue desarrollar un modelo para medir la contribución real del cliente de seguro de manera efectiva.

El trabajo realizó un análisis empírico y estableció un modelo predictivo con los datos de una compañía de seguros de Taiwán. Aplicaron la técnica de Random Forecast Regression (RF), y la compararon con otros métodos como Regresión Lineal, Árboles de decisión, SVM y Generalized Boosted Model. El modelo de regresión RF obtuvo el mejor rendimiento de predicción, con el RMSE más bajo y pseudo R-cuadrados más altos.

El estudio concluyó en que el modelo puede predecir el valor del cliente utilizando solo un pequeño subconjunto de variables, y que la región, la edad, el estado del seguro y el sexo son los más importantes factores para predecir su rentabilidad en la industria aseguradora.

2.3.4 Algoritmo Ensemble Random Forest para Análisis de Big Data en Industria Aseguradora

En el año 2017, un grupo de científicos de la Universidad Tecnológica del sur de China, Weiwei Lin, Ziming Wu, Longxin Lin y Angzhan Wen, en conjunto con el profesor Jin Li de la Universidad de Guangzhou, trabajaron sobre la problemática de la distribución desequilibrada de los datos

comerciales en empresas de seguros, y propusieron un algoritmo tipo Ensemble Random Forest para utilizar en modelos predictivos de clasificación.

Los autores incluyeron en su trabajo la tecnología de Big Data, cuya llegada al mundo abrió un nuevo capítulo en empresa como bancos, seguros y otros sectores financieros tradicionales, para entrar en una nueva era de ciencia y competencia tecnológica. La tecnología de Big Data no solo crea un valor significativo sino también promueve el cambio y progreso de las industrias tradicionales.

El método tradicional de comercialización en la venta de seguros está principalmente basado en negocios de ventas fuera de línea, en donde vendedores de seguros ofrecen los productos de la empresa llamando o visitando a los clientes. Esta forma de marketing ciego ha logrado buenos resultados en el pasado, que mantuvo el rendimiento de ventas de la empresa durante mucho tiempo. Con expansión de la industria de seguros, una gran cantidad de empresas nuevas ingresaron al mercado lo que forma una competencia saludable y promueve constantemente la reforma de la industria.

Por otro lado, la voluntad de las personas de comprar seguros aumentó gradualmente, y los potenciales clientes se fueron expandiendo rápidamente. Según las estadísticas, la tasa de éxito de la venta telefónica tradicional es inferior a una milésima, y la tasa de venta de seguros de un vendedor senior puede llegar al dos por ciento, siendo esto, obviamente muy ineficiente. Por lo tanto, el entender mejor y con mayor precisión la intención de compra de los usuarios se ha convertido en una necesidad muy urgente de una compañía de seguros.

Debido a la falta de propósito e innovación de los métodos tradicionales de comercialización, los datos de negocios de seguros están mal organizados y están oscuras las características de compra de los clientes, lo que conducen directamente a una serie de desequilibrios en la categorización de datos de los productos, generando dificultades para la clasificación de usuarios y la recomendación de productos a generar.

La clasificación en esos conjuntos de datos desequilibrados ha desconcertado a muchos investigadores, inclusos los autores del artículo, utilizando datos reales, no pudieron obtener la distribución de datos esperada, especialmente en los escenarios comerciales más sensibles al costo. En ese contexto de datos desequilibrados, recomendaron elegir algunos métodos de remuestreo que sacrifiquen algunas características para construir conjuntos de datos de entrenamiento relativamente equilibrados. Otra alternativa que evaluaron es la de construir muestras virtuales con el fin de equilibrar la distribución de datos. Como resultado, mejoraron la tasa de reconocimiento de clases, pero sacrificaron la precisión del modelo de clasificación.

El objetivo principal de su trabajo fue proporcionar un modelo de clasificación para la base de datos de negocios de seguros tradicionales, combinada con las tecnologías de Big Data, no solo proporcionando una buena estrategia para la orientación de marketing preciso de productos de seguros, sino también una muy buena referencia para la clasificación de conjuntos de datos desequilibrados.

Durante el desarrollo de su experimentación, exploraron una muestra a gran escala de datos de negocios de seguros, y propusieron un algoritmo Ensemble Random Forest que usó la capacidad de computación paralela y memoria caché optimizado por la herramienta para BigData Apache

Spark. Utilizaron los indicadores F-Measure y G-mean para evaluar el rendimiento del modelo. Adicionalmente, se utilizaron técnicas de bootstrap sobre la muestra para preprocesar los desequilibrios en los algoritmos de clasificación.

Como resultado de su experimentación, el algoritmo Ensemble Random Forest superó a SVM y a otros algoritmos de clasificación en tanto rendimiento como precisión dentro de un conjunto de datos desequilibrados, permitiendo una mejor clasificación de productos de seguros y análisis de clientes potenciales, resultando útil para mejorar los análisis en materia de marketing en comparación con enfoques tradicionales.

2.3.5 Oportunidades Estratégicas de Insurtech en Seguros Personales

Luis Enrique Alarcón Madrid, en su trabajo final del Master en Dirección Aseguradora Profesional de la Universidad Pontificia de Salamanca en 2018, trabajó sobre el paradigma de "Insurtech" que manifiesta un empoderamiento del cliente en una actividad aseguradora, en donde el asegurado comienza a tomar decisiones de manera constante e informada. Dicho contexto genera una movilidad permanente y una pérdida de fidelidad con las aseguradoras, y se manifiesta siempre en movimiento en busca de mejores condiciones y el mejor servicio, en función de sus circunstancias, personalidad, y momento del ciclo vital en el que se encuentra.

Dicho concepto de "Customer Centrity" se ha elevado a un nivel en el que nunca antes había estado, en donde la personalización de la oferta se convierte en vital para ser competitivo, y en un futuro, para la supervivencia en el mercado. Si antes un incremento en la gama de productos a disposición del cliente era un factor que se analizaba desde el punto de vista de los incrementos de los costos, ahora es necesario valorarlo desde el punto de vista de clientes y necesidades que es capaz de satisfacer, y el grado de flexibilidad de la cadena productiva para captar nuevos clientes y para mantenerlos en el tiempo.

En el rubro de los servicios, donde la gestión se realiza fundamentalmente a través de sistemas informáticos integrados, la estandarización y control necesarios para cada proceso han alcanzado una rigidez difícil de transformar de una manera disruptiva, flexibilidad que se requiere para atender las necesidades de los clientes en éste nuevo contexto. Especialmente en el rubro del sector asegurador y financiero, en donde aún predominan sistemas basados en los antiguos pero robustos "mainframes", tienen una complejidad especial para afrontar éstos cambios que ahora la tecnología permite, especialmente porque puede aprovechar la actual y creciente capacidad de captación (BigData) y análisis de datos (Analítica Avanzada). Se hace patente por consiguiente la necesidad de un cambio en la interacción con los clientes, que ha dado origen a la aparición de nuevos competidores directos en el sector capaces de apalancar su ventaja competitiva en la relación ofrecida al cliente mediante el uso de nuevas tecnologías (InsurTech), lo que ha hecho imperativa para las empresas del rubro introducir éste punto en su agenda directiva.

En ese contexto, su trabajo se concentró en generar modelos de predicción utilizando Machine Learning entrenados sobre datos históricos de seguros de Incapacidad Laboral Temporal. Dichos

modelos intentaron predecir por un lado la probabilidad de ocurrencia de un siniestro, y por otro el número de días a indemnizar (costo siniestral) en caso de ocurrencia del siniestro.

En el primer modelo utilizó la técnica de clasificación de árboles de decisión para determinar un valor binomial (siniestro/no siniestro), y en el segundo escogió la técnica Enet Blender para determinar el total de días a indemnizar, presentando la predicción del costo siniestral como la multiplicación de esos días estimados por la garantía diaria de indemnización controlada.

El autor concluyó que la aplicación de Machine Learning al proceso de tarifación del seguro de Incapacidad Laboral Tempral permite obtener un resultado específico, pero también conclusiones generalizables en la aplicación de nuevas tecnologías a procesos de negocio.

La aportación de valor de los dos modelos combinados para la tarifación del seguro permitió obtener una tarifa “dinámica” en función de diversas variables asociadas a la propia vida de la póliza y del asegurado, frente a una tarifa únicamente ligada a la condición “edad”, sobre la cual el asegurado no tiene ningún tipo de control, ya que por ejemplo personas con un estilo de vida saludable se verían penalizadas por el simple hecho de cumplir años. Éste hecho de que la tarifa pueda individualizarse en función del riesgo real de cada asegurado, hace el seguro más atractivo para aquellos que a su vez son más atractivos para la compañía por su perfil de bajo riesgo.

3 MACHINE LEARNIG EN LA ESTIMACIÓN DE RESERVAS EN UNA ASEGURADORA

Como se mencionó en el apartado donde se desarrollaron los conceptos de la industria aseguradora, un punto importante que rige su actividad es el monto que deben inmovilizar en concepto de reservas para hacer frente a los posibles siniestros que pueden ocurrir durante la vigencia de sus pólizas.

Por un lado, se trata de un tema completamente regulado, tanto desde el punto de vista conceptual como en los procedimientos técnicos detallados que se deben ejecutar para calcularlas, que finalmente serán persistidas en los estados contables y balances que deben presentar las aseguradas ante las entidades reguladoras en forma periódica.

Por otro lado, representan montos que no tendrán libre disponibilidad por parte de la empresa para realizar inversiones que le permitan compensar el posible déficit de sus resultados técnicos, y maximizar sus resultados financieros que establece finalmente el lucro de su actividad. Llevar un control y una estimación permanente de dichos indicadores le permite a la empresa planificar tanto sus inversiones como los precios de los productos con un grado menor de incertidumbre que si se limitaría a esperar su cálculo concreto en forma mensual, trimestral o anual.

Si bien los tipos de reservas tratadas en secciones anteriores tienen diferentes formas de cálculo, todos coinciden en procesos que van de lo particular que son las pólizas individuales, hasta lo general que son los montos acumulados en forma mensual. En este contexto, se pueden identificar dos etapas principales en el procesamiento: a nivel de pólizas individuales, y a nivel agregado tanto en general como por los diferentes ramos que maneja la aseguradora.

La determinación de la reserva de una póliza individual es un proceso complejo y muy heterogéneo de acuerdo del ramo a la cual pertenece. Es muy diferente estimar la reserva si el bien asegurado es un automóvil, una persona o un domicilio. Incluso dentro de cada ramo existen disparidades muy marcadas, porque el proceso es diferente si el bien asegurado es un automóvil de uso particular, o forma parte de una flota de uso empresarial.

Por otro lado, cada proceso individual incluye etapas que pueden ir desde búsquedas en tabulaciones estándares de mercado, cálculos entre variables, tablas de decisión basadas en reglas de negocio dinámicas, e incluso peritajes que realizan inspectores en forma presencial, lo que incorpora en algún punto valoraciones sujetas a determinados niveles de subjetividad.

Por último, dentro de los procedimientos que utilizan cada una de las etapas mencionadas anteriormente, participan un sin número de variables como la prima, el premio, la suma asegurada, el período de cobertura, la ubicación geográfica, características del cliente como la edad, estado civil, sexo, y características del bien asegurado como si un automóvil se guarda en un garaje, o la cantidad de cristales de un domicilio.

Una vez calculada la reserva individual de una póliza para todo el período de su cobertura, la aseguradora debe seguir su evolución en el tiempo, y separar la reserva ejecutada y la reserva

pendiente. Su diferencia radica en que la primera representa la porción de la reserva que ya ha generado erogaciones en conceptos de siniestros, mientras que la segunda representará el respaldo ante los futuros siniestros que tenga el contrato hasta el fin de su cobertura.

Al final de la cadena, las aseguradoras deben ejecutar complejos y pesados procesos batchs mensuales, trimestrales y anuales, que permiten calcular el monto general en concepto de reservas que se deben mostrar en los balances y situaciones patrimoniales de la empresa. Esos procesos incluyen sumarizaciones y discriminaciones tanto de las reservas totales y pendientes de las pólizas, como otros tipos de reservas requeridas por ley para cada uno de los ramos que comercialice. Adicionalmente, dado la sensibilidad del asunto, tanto los procesos como sus resultados intermedios, son constantemente monitoreados tanto por personal técnico del área de IT, como por representantes del negocio, porque es muy común la necesidad de ajustes y reprocesamiento hasta conseguir resultados satisfactorios para todas las partes.

El presente trabajo se desarrolla en torno a esa problemática, partiendo de la necesidad de tener una estimación más concreta de los niveles de reserva que se tendrán tanto a nivel individual como agregado, que permita planificar y tomar decisiones con anterioridad a la finalización de los procesos batch que realizan los cálculos definitivos al final de cada ciclo contable.

La utilización de técnicas de Machine Learning para la estimación de éste tipo de reservas, le puede proporcionar a las aseguradoras un mecanismo rápido y fácil de mantener, que le permita disponer y evolucionar las estimaciones a futuro, no solamente al inicio de cada período sino también ir ajustándolas a medida que transcurre el mes.

Para el caso de las reservas de pólizas individuales, se puede disponer de una estimación rápida incluso en el mismo momento de la cotización, que pueda originar decisiones que pueden llegar hasta el rechazo del negocio por parte de la aseguradora. Para el caso de las reservas generales mensuales, puede disponer de una curva con la estimación de la reserva general tanto para el mes en curso como para los próximos meses, que permita tomar decisiones relacionadas con los instrumentos de inversión de corto y mediano plazo, sin estar acoplados a la espera de los largos procesos de cálculo de reserva mencionados anteriormente.

3.1 Entendimiento del Problema

El primer paso de todo proceso de aplicación de Machine Learning es el entendimiento del problema a resolver, que incluye una nivelación conceptual y el reconocimiento de las estructuras de datos con las que se va a trabajar.

Como los datos es uno de los activos más cuidados en las empresas, el acceso a los mismos normalmente incluye una serie de procedimientos y trámites administrativos que involucran diferentes sectores de la empresa como Auditoría y Seguridad Informática, y llevan tiempos y esfuerzo que muchas veces no son tenidos en cuenta en la planificación de los proyectos. Esos procedimientos incluyen normalmente tareas como la firma de convenios de confidencialidad,

solicitud de permisos sobre bases de datos, análisis de riesgos por parte de un analista de seguridad, creación de usuarios, y pueden incluso requerir el desarrollo y ejecución de procesos de ofuscación de los datos que la empresa no está dispuesta a compartir. Las políticas son más rigurosas cuando los equipos de trabajo son externos a la empresa.

Para el caso del presente trabajo, aproveché mi carrera como profesional dentro de uno de los grupos aseguradores más importante del país, que posee además presencia internacional en países limítrofes, y que comercializa productos en todos los rubros de la industria aseguradora durante más de 75 años de presencia en el mercado.

Si bien mi situación de empleado en relación de dependencia de más de 15 años en la empresa me facilitó muchas cosas, igual tuve que cumplir esas políticas de seguridad y el primer paso fue conseguir la habilitación para la utilización de los datos de la empresa, con el compromiso de no divulgación de información de pólizas particulares ni nombres.

Una vez otorgada dicha autorización de los directivos de la empresa, el siguiente paso fue conseguir los permisos concretos en los mecanismos de seguridad de las bases de datos y seguir las formalidades y trámites al respecto. En éste punto es muy importante conocer en términos generales lo que se está buscando y dónde está ubicado, porque la misión de las áreas de seguridad que brindan las habilitaciones y los permisos es la de analizar y mitigar el riesgo de fuga y manipulación incorrecta de la información. Si no se tienen en claro esos temas, situación que puede ser originada por ejemplo en un relevamiento incompleto de la necesidad, comienza un ciclo de averiguaciones y justificaciones, que termina recién cuando se completa toda la información requerida para la habilitación.

Una vez que ya se dispone los permisos y habilitaciones, sigue la tarea de conseguir el conocimiento de las estructuras, modelos de datos, y conceptos técnicos y funcionales del negocio, que incluye no sólo revisar documentación sino también reconocer y entrevistar a los recursos humanos claves del departamento de IT que puedan asesorar sobre la información requerida.

Lamentablemente a la hora de implementar procesos de Business Intelligence en las grandes empresas, el analista se encuentra con un sinfín de modelos y tecnologías operacionales que vienen como “legados” de diferentes generaciones, que están a cargo de personas de distintas áreas del departamento de sistemas. A ésta problemática de la “variedad” de las estructuras de los datos, se le suma la heterogeneidad en la disponibilidad de documentación técnica al respecto, tanto en cantidad como en calidad. Es común encontrarse con componentes con muy buena documentación, pero otros con documentación poco clara, desactualizada, o directamente inexistente.

Para los casos en donde la documentación no alcanza para satisfacer las necesidades de información requerida, el único camino es gestionar una entrevista con algún integrante de los equipos técnicos que dan gobierno a los sistemas operacionales. Esa gestión también puede llevar tiempo, porque esas áreas tienen prioridades que son impuestas por la vertical de negocio específica de la empresa a la que le dan servicio, y si el requerimiento de información del analista no es para resolver una solicitud de la misma área, comienza un proceso de gestión de la demanda y priorización cruzada entre diferentes verticales sólo para conseguir los tiempos para la

realización de las entrevistas. Si bien es una situación común en éste tipo de empresa de gran magnitud, en la industria aseguradora se ve agravada porque se trata de una actividad muy regulada, y los cambios normativos no siempre son requeridos por las autoridades con un margen de tiempo prudencial para realizarlos, generando que la disponibilidad de los técnicos, sea reducida.

Por último, éstos procesos administrativos mencionados no terminan una vez que ya se consigue un set de datos para trabajar, sino es muy común que en el medio del proceso surja la necesidad de expandirlo y enriquecerlo con otros dados provenientes de otras ubicaciones, situación que dispara un nuevo ciclo de justificación, permiso, documentación, entrevista y extracción, que se va a repetir cada vez que se requiera ese tipo de enriquecimiento.

Todos esos tiempos y recursos que están asociados a ésta primera etapa del proceso de Machine Learning hay que considerarlos adecuadamente al inicio y tenerlos en cuenta en la planificación de los proyectos. De lo contrario va a implicar necesariamente replanificaciones y demoras que pueden originar sentimientos de impotencia en los especialistas en Machine Learning, porque ven que los tiempos se estiran por cuestiones que no tienen que ver con su especialización técnica.

3.2 SOLUCIÓN ACTUAL Y CRITERIO DE EVALUACIÓN: RESERVAS VS. SINIESTROS

Cumplido el primer paso de todo proceso de aplicación de Machine Learning que es el entendimiento del problema a resolver, sigue la determinación de un criterio de evaluación que permita contrastar los resultados del proceso actual contra los estimados.

Como se menciona en en el **Capítulo 3.1.5** del presente trabajo, los diferentes tipos de reservas y su proceso de cálculo están definidos de forma normativa en la legislación que emiten los organismos de control de cada país, que en el caso de Argentina se trata de la Superintendencia de Seguros de la Nación para el caso de los seguros generales, y la Superintendencia de Riesgos del Trabajo para el caso de los seguros relacionados con una ART (Aseguradora de Riesgos del Trabajo). En otros países como por ejemplo Uruguay, la actividad está regulada directamente por el Banco Central.

Si bien no puede considerarse como un criterio de evaluación porque las reservas hay que calcularlas a inmovilizarlas igual tal cual dice la ley, vale la pena mencionar que la contrastación natural de un proceso de cálculo de reservas en una compañía aseguradora es su comparación con los pagos realizados en conceptos de siniestros reales ocurridos durante el período de cobertura evaluado.

En ese sentido una de las estadísticas más importantes que tiene una aseguradora es la ***"Estadística Siniestral"***, que como se menciona en el **Capítulo 3.1.6** del presente trabajo, se utiliza para llevar el control de lo que se va gastando en cada póliza, ramo, producto, zona, etc., y poder tomar decisiones al respecto cuando detecta un escalamiento del nivel de siniestralidad que no haya sido estimado previamente. Éste indicador se calcula con largos procesos batchs periódicos,

que obtienen para cada póliza la reserva y los gastos en concepto de siniestros acumulados en forma mensual, trimestral y anual. A continuación en el ***Gráfico 4.1*** se muestra la dispersión de reservas y siniestros de la Estadística Siniestral de Noviembre de 2019, para el ramo de automotores.

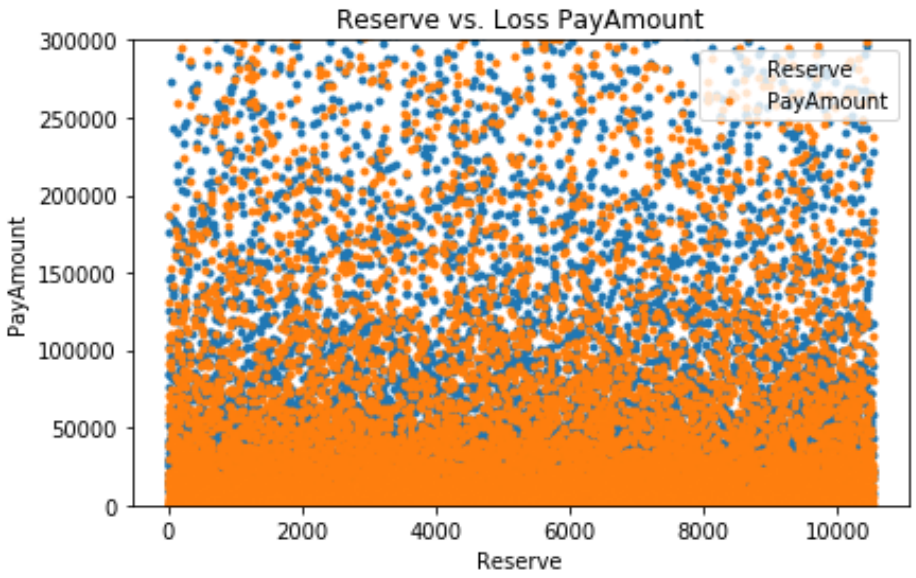


Gráfico 4.1 – Estadística Siniestral: dispersión de las Reservas y Siniestros acumulados a nivel de pólizas individuales para el ramo de automotores.

Como se mencionó en la introducción del presente capítulo, en la aseguradora que representa nuestro objeto de estudio no hay un proceso formal de estimación de reservas sino que ese es justamente el problema que se necesita resolver. Los resultados de los largos procesos batchs que realizan los cálculos, se van compartiendo a representantes del negocio quienes tienen una "idea" de lo que "debería" dar. Si la magnitud de los resultados están muy alejados de lo que se esperaba, se procede a revisar datos y procesos para detectar errores e inconsistencias en el cálculo.

Uno de los tipos de reservas más importante que siguen ese proceso es la "***Reserva Riesgos en Curso"***, que representa los fondos que se deben inmovilizar como respaldo de posibles siniestros futuros de los contratos vigentes. Éste tipo de reservas también se los suele comparar con la siniestralidad, para tener un indicador de los fondos que se reservaron de más o de menos en cada período. Por más que la naturaleza normativa del proceso de cálculo impida la reinversión de los fondos reservados de más, es importante tanto para las áreas financieras como de producto ir llevando el control de esa diferencia. A continuación en el ***Gráfico 4.2***, se presentan las curvas de reserva y siniestralidad, correspondiente a los indicadores generales calculados en forma mensual, tomando las pólizas vigentes de todos los ramos comercializados en argentina, expresando sus valores en pesos argentinos.

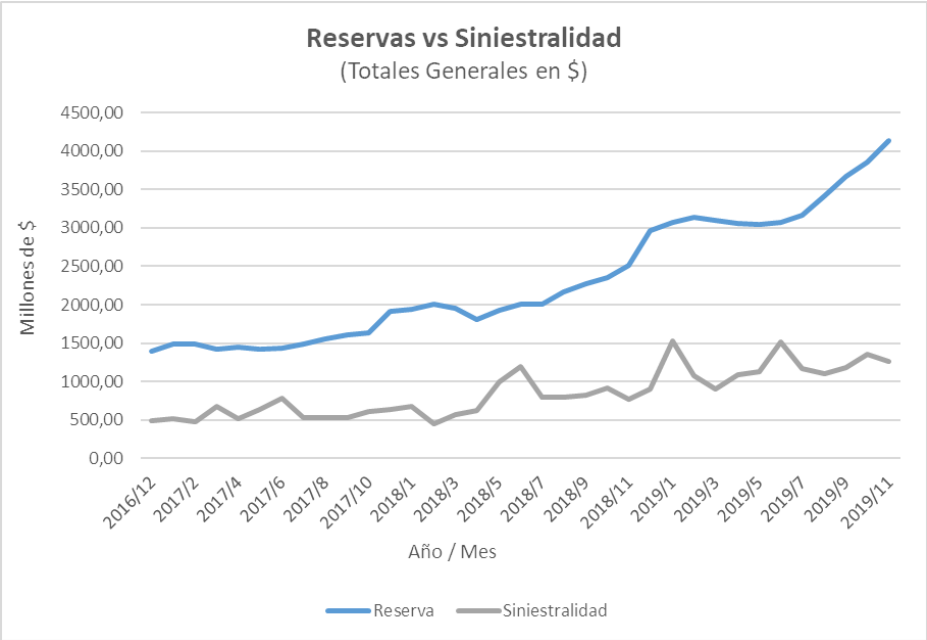


Gráfico 4.2 – Reserva Riesgo en Curso general mensual, contrastada con la siniestralidad general en los mismos períodos

La curva de siniestralidad por naturaleza tiene una variabilidad importante entre un mes y otro, y normalmente denota un componente estacional en los meses de verano e invierno. En cambio la curva de reservas es mucho más suave sin manifestar cambios tan bruscos entre meses contiguos. El punto más representativo de la gráfica, es que la curva de reserva si bien sigue la tendencia de los siniestros, se posiciona muy por encima de ellos, indicando que la empresa está inmovilizando una cantidad de fondos mucho más grande de los que realmente necesita para cumplir sus obligaciones.

Esa realidad que se manifiesta en las sumatorias generales, no siempre es la misma en los diferentes ramos que comercializa la empresa. A continuación en los ***Gráficos 4.3*** se muestran los indicadores de reserva y siniestralidad desagregadas en los ramos de automotores, vida colectivo y combinado familiar respectivamente. Se puede apreciar disparidad en la brecha entre las curvas de reservas y siniestros, respondiendo a las particularidades que ha tenido cada tipo de seguro en los períodos evaluados.

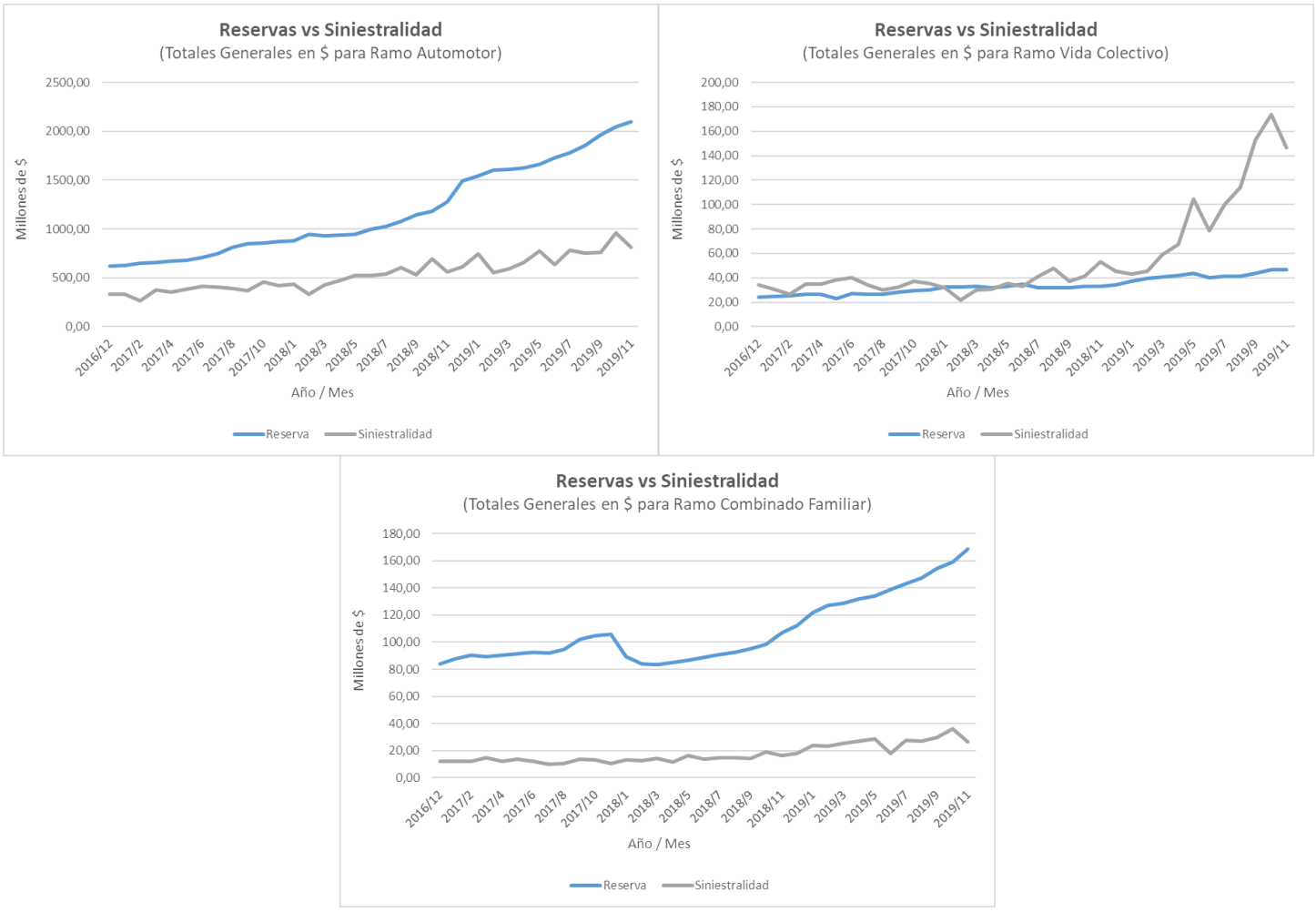


Gráfico 4.3 – Reserva Riesgo en Curso mensual, contrastada con la siniestralidad en los mismos períodos, para los ramos de: Automotores, Vida Colectivo y Combinado Familiar.

Por último, la tendencia ascendente de éste tipo de indicadores expresados en moneda nacional puede estar condicionado por cuestiones inflacionarias y por las variaciones del tipo de cambio, sobre todo en los meses en que se produjeron devaluaciones bruscas. En los ***Gráficos 4.4*** se muestra la evolución del tipo de cambio en el período evaluado, y la comparativa de las curvas de reservas y siniestros expresadas en dólares estadounidenses, convertidas a moneda extranjera utilizando el tipo de cambio oficial del primer día hábil de cada mes.

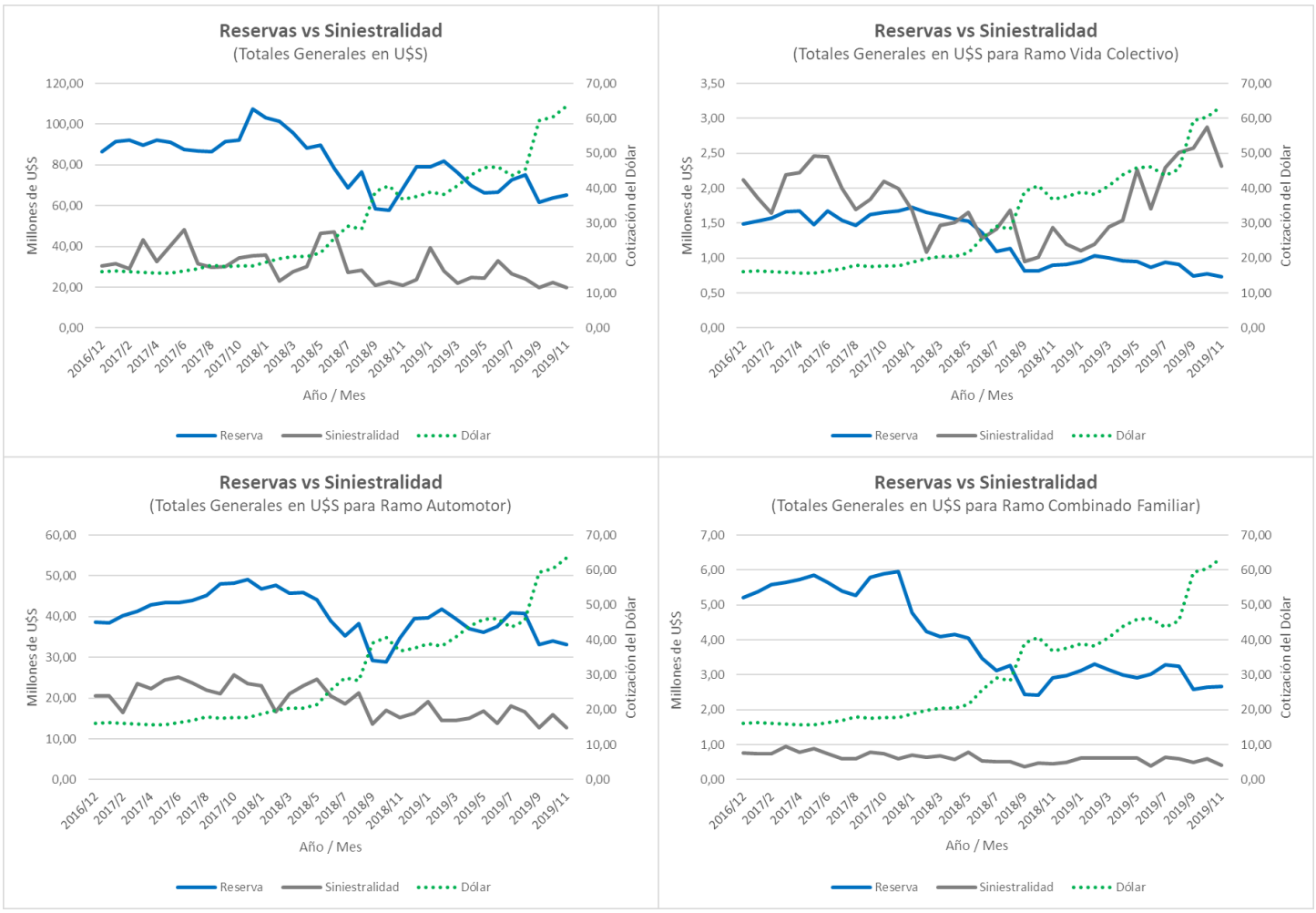


Gráfico 4.4 – Reserva Riesgo en Curso mensual expresada en dólares, contrastada con la siniestralidad en los mismos períodos, y con la evolución del tipo de cambio en los períodos evaluados.

Comparando las gráficas expresadas en moneda nacional (***Gráfico 4.3***) y las expresadas en moneda extranjera (***Gráfico 4.4***), se manifiesta una curva de siniestralidad ya sin tendencias ascendentes sostenidas, indicando que estaba influida por las variaciones del tipo de cambio y su correspondiente impacto en la inflación. Por su parte la curva de reservas expresada en moneda extranjera demuestra una fuerte caída en los meses de devaluación brusca, haciendo que se reduzca la brecha con la curva de siniestralidad. Esto se explica en que los procesos de cálculo de reservas suavizan la tendencia evitando cambios bruscos en los montos que deben inmovilizar financieramente, protegiendo a las aseguradoras tanto de la volatilidad del tipo de cambio como de eventos que puedan generar siniestros de gran magnitud en un período corto de tiempo.

Finalizando el apartado y como el objetivo del presente trabajo es la estimación de reservas, cerramos la fase definiendo el criterio de evaluación para los modelos que se generen en las fases subsiguientes: que tengan la capacidad de predecir la tendencia de los datos de reserva manteniendo niveles de error no mayores al 20% de MAPE (Error de Porcentaje Cuadrático Medio).

3.3 PREPARACIÓN DE LOS DATOS:

Una vez que se logró un entendimiento del problema a resolver y se dispuso un criterio claro de evaluación de resultados, el siguiente paso en el proceso de Machine Learning es la preparación de datos para entrenamiento. Como lo indica la bibliografía mencionada en apartados anteriores, se trata de un proceso largo, complejo y lleno de imprevistos, y el presente trabajo no fue la excepción, ya que se requirió invertir alrededor del 60% del tiempo total del proceso de Machine Learning en tareas de búsqueda, filtrado y preparación de datos.

Al hablar de reservas y montos pagados en conceptos de siniestros, además de existir diferentes tipos con diferentes procesos de cálculo dependiendo del ramo, existen también diferentes ubicaciones en donde quedan persistidos los resultados en las bases de datos de las empresas aseguradoras. En ésta investigación se identificaron datos de reservas y pagos de siniestros en las siguientes locaciones:

- Sistema Core de Emisión de Seguros: gestiona todas las pólizas, productos, y clientes de la empresa. Su modelo de datos contiene la información de los diferentes ramos, cotizaciones, primas, sumas aseguradas, países, monedas y condiciones de cada contrato de seguro emitido.
- Sistema Core de Siniestros: gestiona los reclamos, los siniestros y sus pagos. Adicionalmente su modelo de datos contiene información sobre el cálculo de las reservas pendientes y ejecutadas de cada contrato al momento de los siniestros.
- Datawarehouse: contiene los modelos de datos dimensionales con información agregada en base a diferentes temas como los ramos, zonas, clientes, productos, etc.
- Históricos: contiene información histórica de los procesos batchs principales de la empresa relacionada con productividad, rentabilidad, contabilidad, reservas, etc..

El punto de partida fue la identificación de las tablas de datos resultantes de los dos procesos batchs más importantes de la empresa que se encargan de realizar operaciones estadísticas de reservas y siniestros, que son, como se mencionó en el punto anterior, los procesos relacionados con ***"Estadística Siniestral"*** y con ***"Reserva Riesgo en Curso".*** El primer conjunto de datos contiene información de las pólizas individuales y sus respectivos cálculos de reserva total, reserva pendiente y el monto real siniestrado para cada período, preparados en forma acumulativa por mes, por año e histórico. En cambio el segundo conjunto contiene las pólizas individuales y sus correspondientes reservas calculadas de acuerdo a como lo dicta la ley, pero sólo los contratos vigentes de cada mes sin realizar agregaciones y sin considerar siniestros.

Si bien ambos sets de datos representaron el punto de partida, fue necesario enriquecerlos con información proveniente del resto de los modelos de datos mencionados, para identificar todas las posibles características relevantes que puedan ser de utilidad en el armado de los algoritmos

como ser los datos de los clientes, la descripción de los ramos, los valores de las primas, de las sumas aseguradas y de los pagos individuales de siniestros.

Una problemática relacionada con la normalización de los datos es el manejo de las diferentes monedas en las que están expresados los montos tanto de los contratos como la de los pagos. Por un lado se debe normalizar las monedas de las primas y sumas aseguradas de las pólizas correspondientes a los diferentes países en los que tiene presencia la aseguradora (Argentina, Uruguay, Paraguay y Brasil), y por otro lado los pagos que se pudieron realizar utilizando monedas extranjeras (dólares). Dicha normalización implica realizar la conversión a una moneda común (pesos argentinos) utilizando la cotización del momento de la firma del contrato y del pago del siniestro.

Un especial foco de análisis tuvo lugar ante el reconocimiento de ***"outliers"***. Ambos sets de datos mostraban la presencia de puntos extremos que correspondían a contratos grandes que posee la aseguradora, en donde una única póliza consolida todos los certificados de bienes y personas de grandes empresas. Dichos valores no pueden ser filtrados de ninguna manera ya que si bien son minoría en el universo de datos preparados, su repercusión en los montos totales de reservas y siniestros es considerable, por lo que la única alternativa es trabajar con algoritmos que puedan manipularlos como tales. Para minimizar ese desbalanceo en la distribución de los datos y ajustarlos en magnitudes manejables por los algoritmos de Machine Learning, se procedió a transformar la unidad de medida en base monetaria a números entre 0 y 1, utilizando cocientes fijos definidos arbitrariamente en valores algo mayores a los máximos.

Por último, se prepararon archivos tomando las pólizas vigentes de Argentina entre diciembre de 2016 y noviembre de 2019 (3 años), tanto a nivel general para todos los ramos y productos comercializados, como a nivel de los ramos particulares de Automotores, Vida Colectivo y Combinado familiar. Por otro lado se prepararon datos tanto a nivel de pólizas individuales como agrupados por mes/año.

A continuación se presentan ejemplos de las estadísticas de algunos de los tensores generados. La ***Tabla 4.1*** representa los datos extraídos de pólizas individuales de la Estadística Siniestral para el ramo Automotores, la ***Tabla 4.2*** los generados a partir de la Reserva Riesgo en Curso, y en la ***Tabla 4.3*** las diferentes agregaciones de Reservas Riesgos en Curso por mes para diferentes ramos (tanto en pesos como en dólares).

	Ramo	Producto	Prima	Reserva	Pagos
Cantidad	168093	168093	168093	168093	168093
Promedio	387,3	90,5	139821,5	148563,8	96180,1
Desv.Estand.	467,6	85	5786695,8	3955662	3114640,7
Mínimo	200	1	0	0	0
25%	200	21	7140,2	8000	0
50%	200	99	25097,2	25331,7	6360
75%	200	99	54161,9	75802	41762,9
Máximo	3100	950	1230189623	768421887,9	650468208,6

Tabla 4.2 – Tensor de pólizas individuales extraídas de las tablas resultantes del proceso de "Estadística Siniestral"

	Ramo	Producto	Prima	Capital Asegurado	Reserva
Cantidad	603534	603534	603534	603534	603534
Promedio	200	99	7026,5	1885167,8	3425,7
Desv.Estand.	0	0	97195,6	43773230,5	54730,1
Mínimo	200	99	0	1	0
25%	200	99	1229,2	240000	316,8
50%	200	99	2147,6	398000	804,3
75%	200	99	3923	644000	1862
Máximo	200	99	18822000,5	3361943518	9351362,9

Tabla 4.2 – Tensor de pólizas individuales extraídas de las tablas resultantes del proceso de "Reserva Riesgo en Curso"

	Dolar	AnioMes	Reserva Total $	Reserva Total U$S	Reserva Auto$	Reserva Auto U$S	Reserva VidaColectivo $	Reserva Combinado $
Cantidad	35	35	35	35	35	35	35	35
Promedio	30,4	17	2312396580	81109988,5	1174426651	40322700	33564904,6	108357911,4
Desv.Estand.	14,6	10,2	807631270,8	13137279,7	463386076,4	5251499,1	6749310,5	24866858,4
Mínimo	15,7	0	1390440447	57827025,9	622497998,2	28927583,7	23168169	83621846,7
25%	17,7	8,5	1579958466	69325217,1	827031098	37314605,7	27903385,7	89805422
50%	25,6	17	2008775380	81917728,9	997788592,9	40265376,2	32628696,1	95004678,6
75%	40,7	25,5	3063154290	91260588,6	1605432990	44066622,6	39796950,7	127666141,3
Máximo	63,5	34	4140962468	107436830,9	2099492780	49015434	46742788,9	168658972,8

Tabla 4.3 – Tensor con agregaciones por mes/año, extraídas de las tablas resultantes del proceso de "Reserva Riesgo en Curso"

3.4 CONSTRUCCIÓN DEL MODELO

Contando con los sets de datos preparados, el paso que sigue es la elección de los algoritmos de Machine Learning y la ejecución de experimentos. Todo experimento requiere de ambientes de ejecución que consiste básicamente en HW, SW y conectividad.

El ambiente utilizado para el presente trabajo constó de las siguientes características:

- Computadora personal con procesador Intel Core i7 2.7 GHz (4 Cores HT), 16 GB RAM DDR3 1600 MHz, 512 SSD, y GPU NVidia 1024 MB.

- Sistema Operativo Host OS X con producto de virtualización Parallels.
- Máquina virtual configurada con 8 CPU, 8 GB RAM, y 20 GB Disco.
- Sistema Operativo Linux Ubuntu 18.04 LTS.
- Entorno para Machine Learning Jupyter Notebook con Python 3.
- Subcripción de Microsoft Office 365 para productos de oficina y repositorio de archivos.

Analizando la distribución de los datos, la primera moción fue utilizar una de las técnicas de Machine Learning más rápidas, simples, y menos costosa, que es la ***"Regresión Lineal"***, para luego incursionar en técnicas más específicas, fundamentalmente especializadas en análisis y predicción de datos en series de tiempo como las ***"Redes Neuronales LSTM"*** y los métodos estadísticos **"ARIMA"**. El código fuente completo de los experimentos se presenta en el **Anexo I**, donde se pueden visualizar detalles más concretos de los algoritmos y las librerías utilizadas.

Cada experimento cuyos resultados se presentarán a continuación, requirió de muchas ejecuciones con diferentes parámetros técnicos de configuración, principalmente tasa de entrenamiento, iteraciones, tamaño de bloque y magnitud de las variables, presentándose los resultados que mejor precisión y menor margen de error se consiguieron en cada corrida.

En ese contexto, se realizaron experimentos siguiendo dos enfoques distintos: uno tratando de generar modelos de predicción para reservas de pólizas individuales, y otro para predecir la reserva acumulada a nivel mensual.

3.4.1 Predicción de Reserva para una Póliza individual

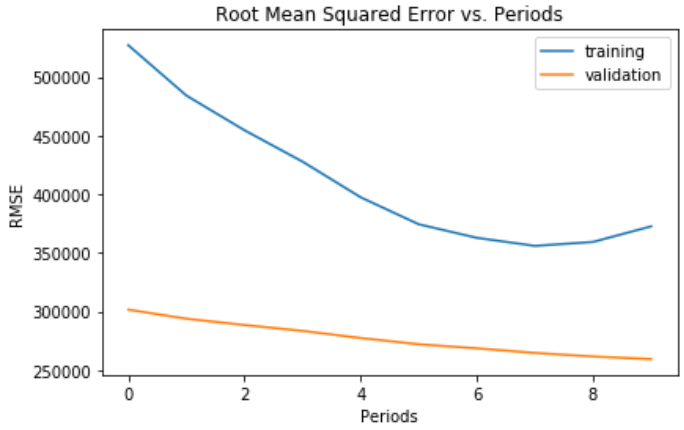
Tomando como punto de inicio la configuración de un modelo de regresión lineal, la decisión más importante es la elección de las características de los datos que se van a usar como variables independientes, y cuál va a ser la característica que se desea obtener como resultado de la estimación. Para el presente trabajo, como el objetivo es la predicción de la reserva y que la aseguradora tenga una previsión de lo que va a tener que inmovilizar por ley a fin de mes, está claro que la característica a utilizarse como ***"target***" en la configuración del algoritmo es el indicador de reserva. Se podría haber seleccionado en su lugar la siniestralidad, pero tendría otro fin la estimación, ligado a la predicción de la siniestralidad. Si bien es muy interesante predecir siniestralidad, los montos de reservas van a tener que ser inmovilizados de todas maneras por cuestiones normativas.

Los primeros experimentos se ejecutaron sobre el set de datos desagregados a nivel de pólizas individuales obtenidos a partir de la Estadística Siniestral, utilizando una única variable independiente que es una de las características que más poder tiene sobre la estimación de la reserva que es la prima de la póliza. Los juegos de datos se dividieron en tres grupos: el 70% para entrenamiento, el 15% para validación y el otro 15% para testing,

En el ***Gráfico 4.5*** que se presenta a continuación, se muestra la corrida del entrenamiento con mejor RMSE del algoritmo de regresión lineal conseguido, tomando como única variable

independiente la prima, y como dependiente la reserva, utilizando únicamente los datos de estadística siniestral de las pólizas del ramo automotores individuales del mes de noviembre de 2019.

Training model… learning_rate=0.001,steps=100,batch_size=5
Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)
00 - 0.56 - 527109.87 - 301559.91
01 - 0.56 - 484163.08 - 293835.69
02 - 0.53 - 454590.71 - 288450.89
03 - 0.54 - 427803.86 - 283438.42
04 - 0.53 - 397387.40 - 277368.96
05 - 0.54 - 374366.02 - 272043.75
06 - 0.53 - 362817.87 - 268565.13
07 - 0.53 - 355952.25 - 264647.13
08 - 0.54 - 359304.57 - 261585.63
09 - 0.54 - 372599.15 - 259410.95
Model training finished.
Total Minutes (ElapsedTime): 5.3944045901



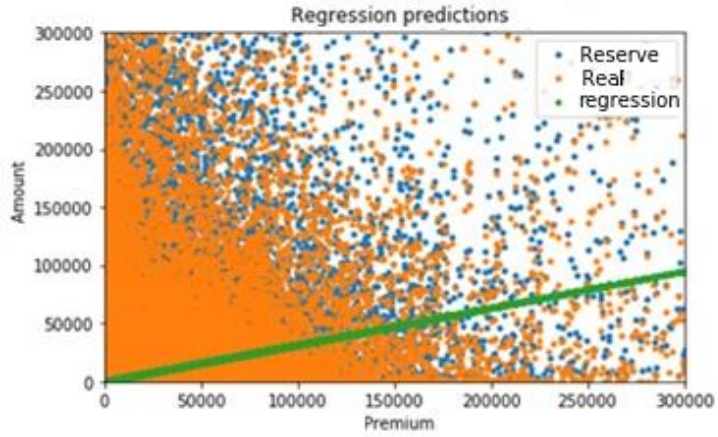


Gráfico 4.5: Entrenamiento de un modelo de regresión lineal con datos extraídos del proceso de Estadística Siniestral

En el primer cuadro de texto se muestra la salida del proceso de entrenamiento, figurando los parámetros de configuración del algoritmo, el proceso iterativo de entrenamiento con los períodos, error de entrenamiento, validación y tiempo de cada iteración, y finalmente el resultado final indicando el tiempo total incurrido en la corrida. En el segundo cuadro se muestra la evolución del error cuadrático de entrenamiento y validación durante el proceso.

Por último, en el gráfico de dispersión final se muestran los datos originales y las predicciones realizadas por el modelo ya entrenado. Si bien el modelo de entrenó con los datos de reserva, en el gráfico se incluyó adicionalmente los valores de siniestralidad real de cada póliza, quedando graficado sobre un EJE X la prima de las pólizas individuales, y en el EJE Y los montos tanto de la reserva de cada póliza (puntos azules) como de los pagos de siniestros reales (puntos naranja), obteniéndose en verde la recta de regresión que mejor ajuste se consiguió.

Claramente una regresión lineal en éste modelo no brinda resultados útiles, fundamentalmente porque para diferentes pólizas con un mismo valor de prima, existen diferentes resultados posibles de reservas y montos siniestrados, sin ningún tipo de linealidad, representando una mala elección de atributos o de algoritmo. En éste punto es donde se necesitó volver al primer paso del proceso de Machine Learning que es el entendimiento de los datos, produciéndole la primera iteración en las fases del proceso.

El problema finalmente estuvo en el origen de los datos, porque la Estadística Siniestral si bien realiza cálculos estadísticos sobre reservas, está enfocada fundamentalmente en la producción que ha generado siniestros en el período, dejando de lado muchos casos de pólizas vigentes que no han tenido siniestros, y realizando un proceso de acumulación especial en el cálculo para diferentes períodos (un mes, un año, histórico) que hace que finalmente los datos tengan una distribución no lineal. Para realizar predicciones en éste contexto, se debe pensar en otros tipos de algoritmos de machine learning.

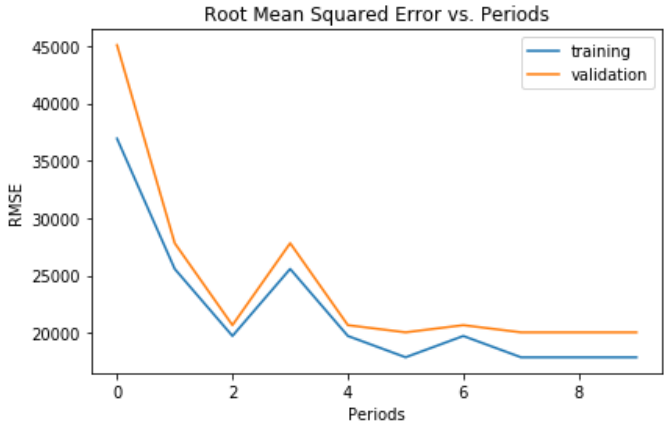
Como el objetivo del trabajo es la predicción de reservas, se descartó el set de datos originados en la Estadística Siniestral y se enfocaron los experimentos en el originado en la Reserva Riesgo en Curso. A continuación, en el ***Gráfico 4.6*** se muestran los resultados obtenidos del entrenamiento con el set de datos extraídos del proceso de Reserva Riesgo en Curso, en donde se volvió a intentar procesar una regresión lineal sobre los datos de las pólizas de automotores individuales del mes de noviembre de 2019, utilizando como variable independiente la prima y como variable dependiente la reserva.

Como el conjunto de datos es más grande, los tiempos de entrenamiento fueron sensiblemente superiores al anterior, pero el error que se iba calculando durante las iteraciones fue muy inferior y mucho más cercano si se compara el error de entrenamiento y el de validación, situación que se ve reflejada muy claramente en el gráfico de dispersión final con una recta de regresión mucho más cercana a los datos. A diferencia de la corrida anterior, en éste gráfico de dispersión no se incluyó la siniestralidad real porque no se disponía de ese dato a nivel de pólizas individuales sin acumular. Como mencionamos anteriormente, a diferencia de la Estadística Siniestral, el proceso de Reserva Riesgo en Curso se centra en el cálculo de reservas y no tiene en cuenta los siniestros.

Training model... learning_rate=0.01,steps=100,batch_size=100
Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)

00 - 4.30 - 36938.27 - 45088.55
01 - 4.27 - 25579.09 - 27822.24
02 - 4.26 - 19734.86 - 20677.87
03 - 4.29 - 25580.25 - 27823.90
04 - 4.44 - 19734.86 - 20677.87
05 - 4.35 - 17877.22 - 20052.40
06 - 4.31 - 19734.86 - 20677.87
07 - 4.27 - 17877.21 - 20052.40
08 - 4.27 - 17877.21 - 20052.40
09 - 4.49 - 17877.21 - 20052.40

Model training finished.
Total Minutes (ElapsedTime): 43.251089759



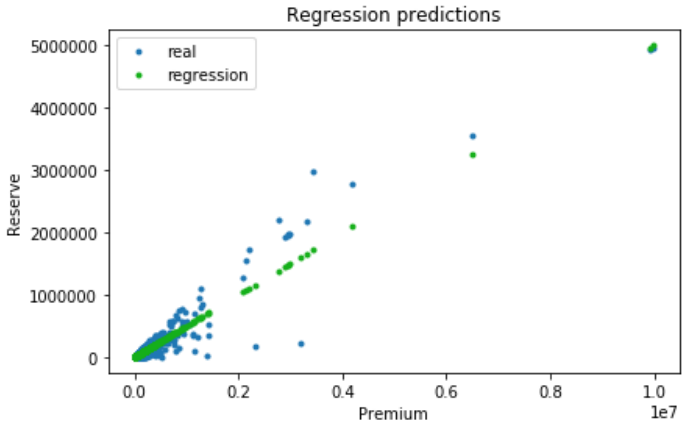


Gráfico 4.6: Entrenamiento de un modelo de regresión lineal a datos extraídos del proceso de Reserva Riesgo en Curso

En éste caso, se obtuvieron mejores resultados de ajuste, indicando que si bien el proceso de Reserva Riesgo en Curso es complejo y participan muchas variables y condiciones, existe una tendencia lineal si lo comparamos únicamente con los valores de la prima de cada póliza.

En éste punto, una vez que se empiezan a tener resultados esperanzadores con las predicciones de los modelos entrenados, es donde empiezan a surgir las diferentes alternativas para continuar con el trabajo, que tiene como objetivo conseguir modelos con mejor ajuste y capacidades de predicción.

Como se mencionó en el **Capítulo 3.2.1**, la naturaleza del proceso de Machine Learning es iterativa, y en esa búsqueda de mejores resultados se empiezan a disparar las transiciones de la ***Figura 3.1***, que implica iterar entre las fases de **Análisis de Errores** y **Construcción del Modelo** para probar diferentes alternativas de parámetros e incluso diferentes tipos de algoritmos. También implica volver a la **Preparación de los datos** para incluir más características de los datos que puedan resultar relevantes, o incluso volver a la fase de **Entendimiento del Problema** para para encontrar la causa de los buenos o malos resultados que se van teniendo, como ocurrió en éste caso para descartar el set de datos originados en la Estadística Siniestral que se mencionó anteriormente.

Como las alternativas son muchas y muy variadas, en el presente trabajo se intentó volver a las preparación de los datos e incorporar alguna característica adicional que pueda ser relevante y que mejore las predicciones. En ese sentido, se procedió a enriquecer los datos incorporando el ***"Capital Asegurado"*** como una nueva variable del algoritmo de regresión lineal.

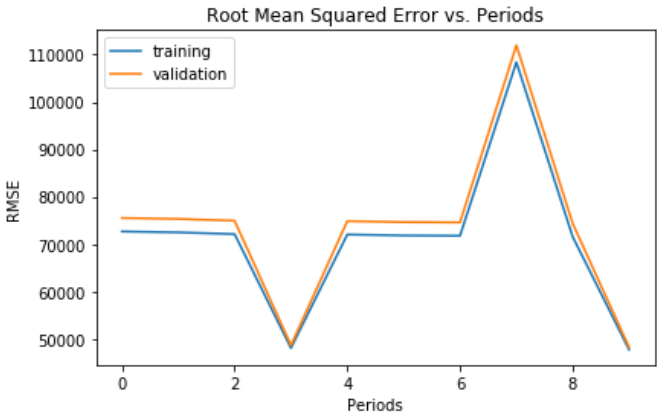
En el ***Gráfico 4.7*** se observa el proceso de entrenamiento y dispersión de la mejor corrida conseguida sobre el conjunto de datos de pólizas de automotores individuales del mes de noviembre de 2019 originados en la Reserva Riesgo en Curso, agregando una variable independiente adicional al algoritmo de regresión antes mencionado, resultando un modelo con dos variables independientes: prima y capital asegurado, y una dependiente: reserva.

Si bien el error durante el proceso de entrenamiento del modelo con dos variables finalmente quedó bastante cercano al conseguido en el modelo con una única variable, se necesitaron ejecutar más cantidad de corridas jugando con diferentes parámetros de ejecución (tasa de entrenamiento, iteraciones y tamaño de bloque) hasta conseguir un resultado estable y que el error cuadrático RMSE no termine divergiendo durante el proceso. Esto ocurre cuando en el proceso de entrenamiento el error en lugar de achicarse a medida que ocurren las iteraciones se va agrandando y termina disparándose. En esos casos se requiere cortar el proceso, y probar con otros parámetros. Cuando más complejo es el modelo, más complejo va a ser el procedimiento de entrenamiento.

Training model... learning_rate=0.0001,steps=100,batch_size=50
Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)

00 - 3.52 - 72694.75 - 75541.94
01 - 3.52 - 72506.75 - 75347.93
02 - 3.57 - 72137.40 - 74965.86
03 - 3.50 - 48108.96 - 48678.81
04 - 3.51 - 72056.96 - 74882.86
05 - 3.74 - 71853.00 - 74671.86
06 - 3.60 - 71789.84 - 74606.75
07 - 3.52 - 108405.51 - 111954.92
08 - 3.52 - 71551.62 - 74360.42
09 - 3.62 - 47783.49 - 48263.33

Model training finished.
Total Minutes (ElapsedTime): 35.6133875012



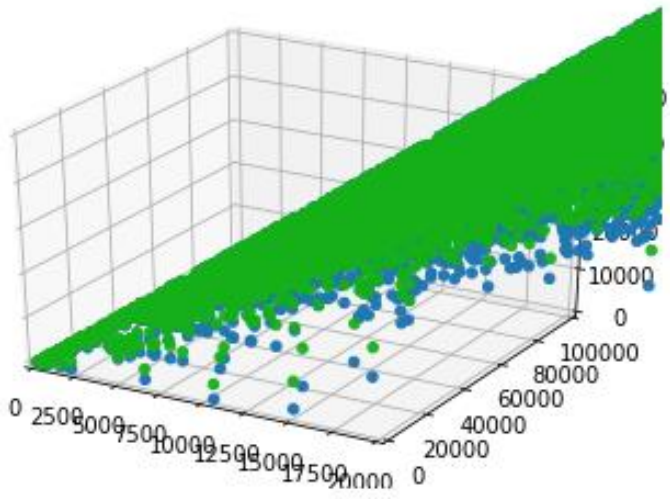


Gráfico 4.7: Entrenamiento de un modelo de regresión lineal a datos extraídos del proceso de Reserva Riesgo en Curso, con dos variables independientes: prima y capital asegurado, y una dependiente: reserva

3.4.2 Predicción de Reserva acumulada mensual

Luego de trabajar con regresiones a nivel de pólizas individuales, la propuesta era trabajar con la reserva agregada a nivel mensual, en donde la situación es muy diferente porque existe una única variable independiente, que es un mes / año en particular, representando en una Serie Temporal. De ésta forma la cantidad de datos de entrenamiento se reduce a un único valor por mes, que en comparación a los modelos anteriores, al tener pocos ejemplos los tiempos de procesamiento y los parámetros de tasa de entrenamiento, tamaño de bloque e iteraciones fueron muy diferentes.

Se prepararon tres modelos, uno para cada tipo de técnica seleccionada (Regresión Lineal, ARIMA y LSTM), que se entrenaron sobre una base de ejemplos históricos de 3 años (desde diciembre de 2016 a noviembre de 2019), generados para estimar la reserva de riesgos en curso general tomando en cuenta todos los ramos comercializados en Argentina. Cada modelo se entrenó con dos juegos de datos diferentes, uno con los datos expresados en moneda nacional (pesos argentinos), y otro con los datos expresados en dólares estadounidenses, utilizando el tipo de cambio oficial del mes para cada dato.

De ésta manera se obtuvieron seis modelos entrenados (dos por cada técnica), a los que se le agregó un séptimo que es una combinación de los dos modelos de Regresión Lineal, el cual consiste en tomar la estimación realizada por el modelo entrenado en pesos, y convertir ese resultado a dólares luego de la predicción. El objetivo fue agregar una alternativa que intente romper la linealidad de los modelos de regresión lineal para obtener una curva que acompañe mejor la trayectoria de los datos.

Dichos procesos de entrenamiento tuvieron algunas particularidades en relación a las corridas anteriores sobre pólizas individuales, que implicaron cambios y ajustes adicionales tanto en los datos como en la configuración de los algoritmos:

- La variable independiente no es continua sino discreta (Mes / Año)
- Sólo se disponían de 37 ejemplos de entrenamiento (un valor por mes/año).
- Por la cantidad de datos reducidas, todos los datos se usaron para entrenamiento y no se hizo la división en conjuntos diferentes para entrenamiento, validación y testing.
- A excepción de la regresión, los algoritmos especializados en series temporales (ARIMA y LSTM) requieren parametrizaciones especiales que tienen que ver con la incidencia de la temporalidad de los datos.

En relación a éste último punto, a continuación en ***el Gráfico 4.8*** se muestran las curvas de los datos originales de Reserva Riesgo en Curso general, expresados tanto en pesos como en dólares, extrayendo a través de técnicas estadísticas por un lado la tendencia y por otro lado la estacionalidad. Se denota claramente una tendencia fuertemente ascendente para el caso de la curva en pesos, y levemente descendente en la curva en dólares, reafirmando lo mencionado en el **Capítulo 4.2** del presente trabajo sobre los procesos inflacionarios y devaluatorios. En cuanto a la

estacionalidad, se reconocen claramente tres ciclos en ambos casos, indicando una estacionalidad anual.

Reserva Total en Pesos Argentinos **Reserva Total en Dólares**

Gráfico 4.8: Tendencia y estacionalidad extraídas de las curvas de datos de Reserva Riesgo en Curso expresadas tanto en Pesos Argentinos como en Dólares Estadounidenses.

Con diferentes intensidades, pero el mismo diagnóstico cabe para las curvas de los datos de los ramos particulares de Automotores, Vida Colectivo y Combinado Familiar, expresados a continuación en los ***Gráficos 4.9, 4.10 y 4.11*** respectivamente.

Reserva Automotores en Pesos Argentinos **Reserva Automotores en Dólares**

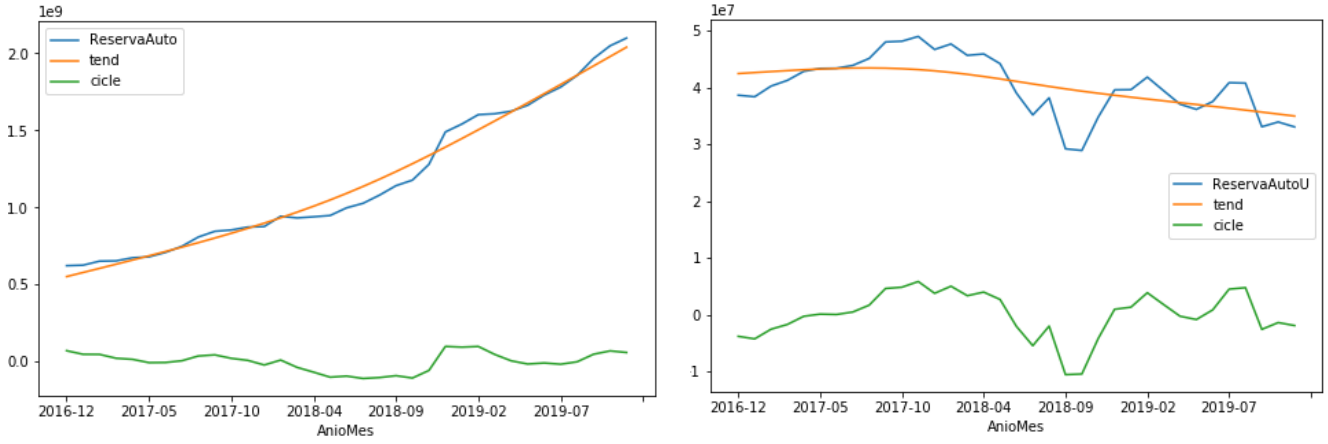


Gráfico 4.9: Tendencia y estacionalidad extraídas de las curvas de datos de Reserva Riesgo en Curso del ramo Automotores, expresadas tanto en Pesos Argentinos como en Dólares Estadounidenses.

Reserva Vida Colectivo en Pesos Argentinos **Reserva Vida Colectivo en Dólares**

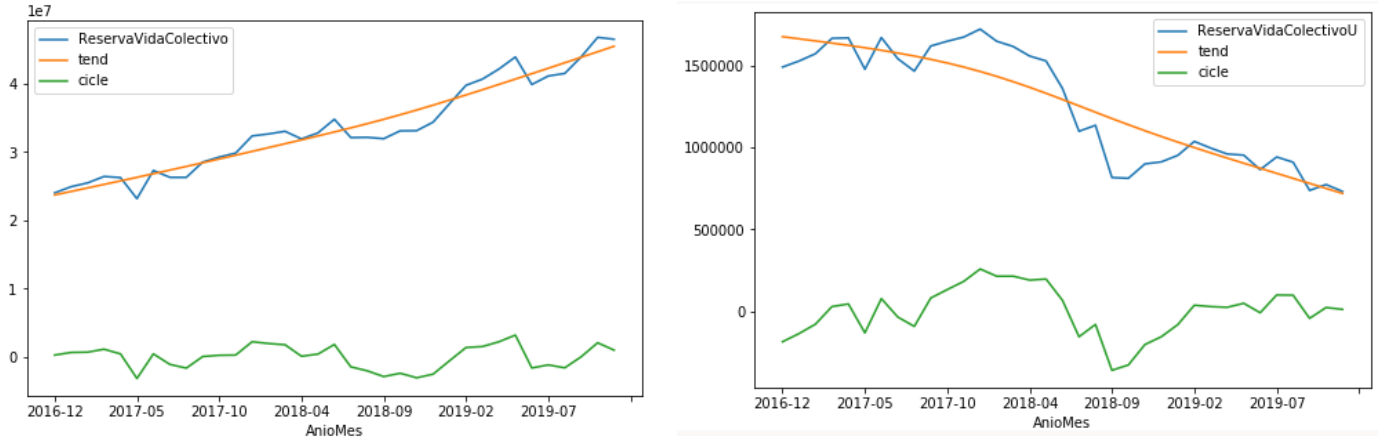


Gráfico 4.10: Tendencia y estacionalidad extraídas de las curvas de datos de Reserva Riesgo en Curso del ramo Vida Colectivo, expresadas tanto en Pesos Argentinos como en Dólares Estadounidenses.

Reserva Combinado Familiar en Pesos Argentinos **Reserva Combinado Familiar en Dólares**

Gráfico 4.11: Tendencia y estacionalidad extraídas de las curvas de datos de Reserva Riesgo en Curso del ramo Combinado Familiar, expresadas tanto en Pesos Argentinos como en Dólares Estadounidenses.

3.4.2.1 Entrenamiento de modelos de Regresión Lineal:

En los ***Gráficos 4.12, 4.13, 4.14 y 4.15*** se muestra el resumen del proceso entrenamiento para los modelos de Regresión Lineal de cada uno de los sets de datos (Reserva Total, Automotores, Vida Colectivo y Combinado Familiar). Pese a la poca cantidad de datos de entrenamiento, los algoritmos lograron eficientemente conseguir curvas que reproducen la tendencia de la curva de reserva y que minimiza el error entre las predicciones y datos de entrenamiento.

Regresión Lineal para Reserva Total en Pesos

Training model...
learning_rate=0.01,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.03 - 0.43
01 - 0.02 - 0.38
02 - 0.02 - 0.33
03 - 0.02 - 0.27
04 - 0.02 - 0.22
05 - 0.02 - 0.17
06 - 0.02 - 0.13
07 - 0.02 - 0.10
08 - 0.02 - 0.09
09 - 0.02 - 0.09
Model training finished.
Total Minutes (ElapsedTime): 0.2135057648

Regresión Lineal para Reserva Total en Dólares

Training model...
learning_rate=0.03,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.58
01 - 0.02 - 0.43
02 - 0.02 - 0.29
03 - 0.02 - 0.19
04 - 0.02 - 0.18
05 - 0.02 - 0.16
06 - 0.02 - 0.16
07 - 0.02 - 0.15
08 - 0.02 - 0.14
09 - 0.02 - 0.14
Model training finished.
Total Minutes (ElapsedTime): 0.2106004993

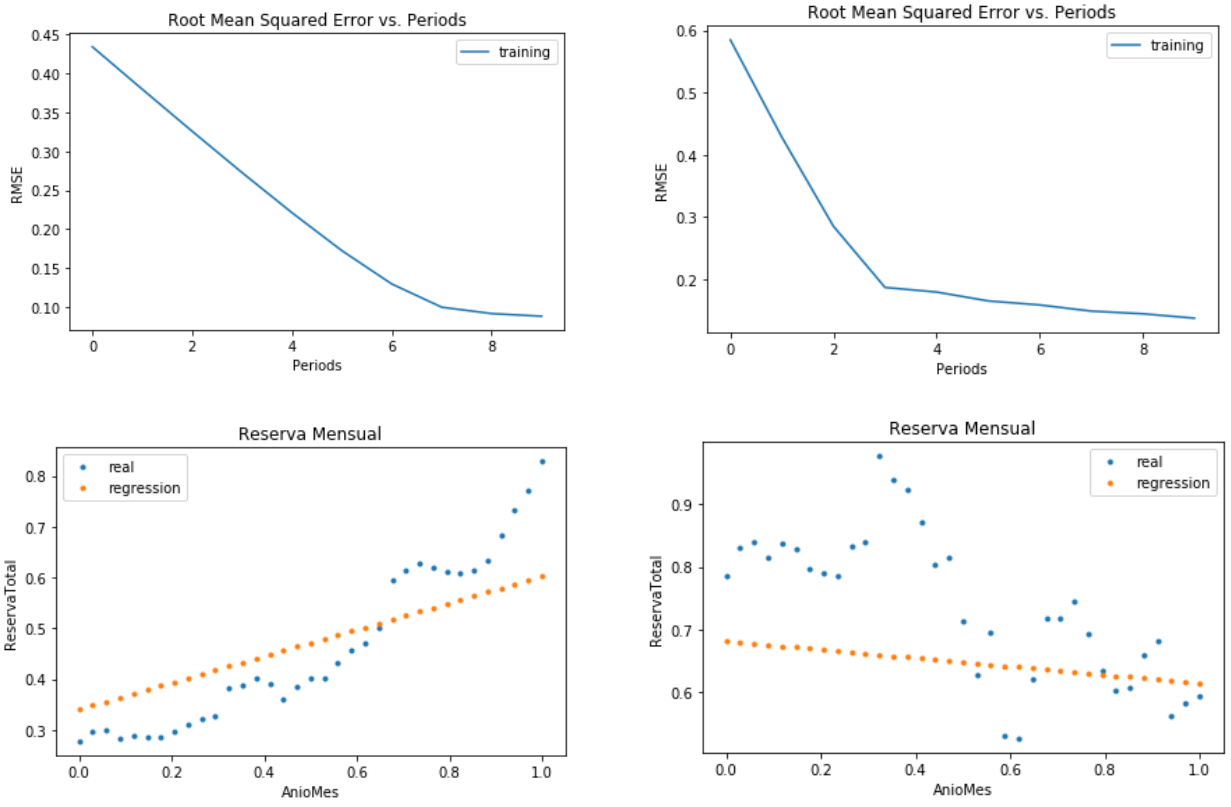


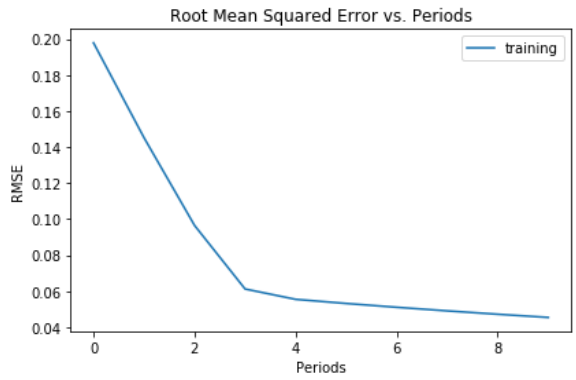
Gráfico 4.12: Entrenamiento de los modelos de regresión lineal con datos de Reserva Riesgo en Curso total acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

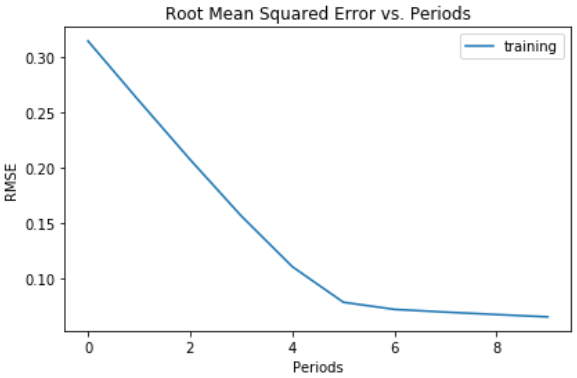
Regresión Lineal para Reserva Automotores en Pesos

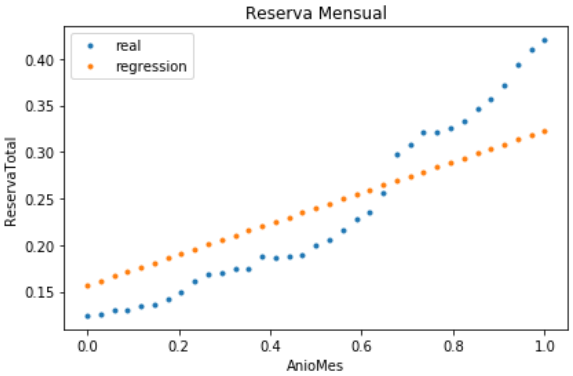
Training model...
learning_rate=0.01,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.20
01 - 0.02 - 0.15
02 - 0.02 - 0.10
03 - 0.02 - 0.06
04 - 0.02 - 0.06
05 - 0.02 - 0.05
06 - 0.02 - 0.05
07 - 0.02 - 0.05
08 - 0.02 - 0.05
09 - 0.02 - 0.05
Model training finished.
Total Minutes (ElapsedTime): 0.2177311699

Regresión Lineal para Reserva Automotores en Dólares

Training model...
learning_rate=0.01,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.31
01 - 0.02 - 0.26
02 - 0.02 - 0.21
03 - 0.02 - 0.16
04 - 0.02 - 0.11
05 - 0.02 - 0.08
06 - 0.02 - 0.07
07 - 0.02 - 0.07
08 - 0.02 - 0.07
09 - 0.02 - 0.07
Model training finished.
Total Minutes (ElapsedTime): 0.2151371479







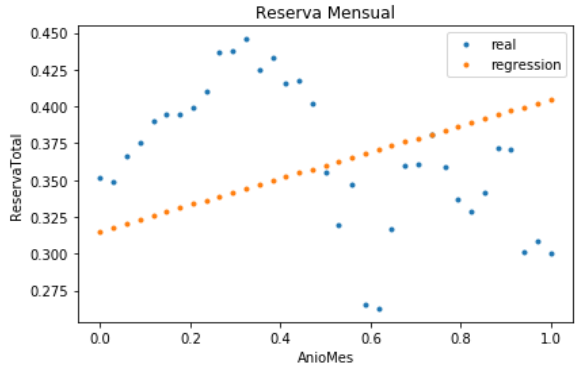


Gráfico 4.13: Entrenamiento de los modelos de regresión lineal con datos de Reserva Riesgo en Curso del ramo Automotores acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

Regresión Lineal para Reserva Vida Colectivo en Pesos

Training model...
learning_rate=0.01,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.00094
01 - 0.02 - 0.00055
02 - 0.02 - 0.00053
03 - 0.02 - 0.00051
04 - 0.02 - 0.00050
05 - 0.02 - 0.00049
06 - 0.02 - 0.00048
07 - 0.03 - 0.00047
08 - 0.03 - 0.00047
09 - 0.06 - 0.00046
Model training finished.
Total Minutes (ElapsedTime): 0.2707012653

Regresión Lineal para Reserva Vida Colectivo en Dólares

Training model...
learning_rate=0.02,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.00946
01 - 0.02 - 0.00761
02 - 0.02 - 0.00619
03 - 0.02 - 0.00510
04 - 0.02 - 0.00427
05 - 0.02 - 0.00364
06 - 0.02 - 0.00316
07 - 0.02 - 0.00280
08 - 0.02 - 0.00253
09 - 0.02 - 0.00232
Model training finished.
Total Minutes (ElapsedTime): 0.2176881909

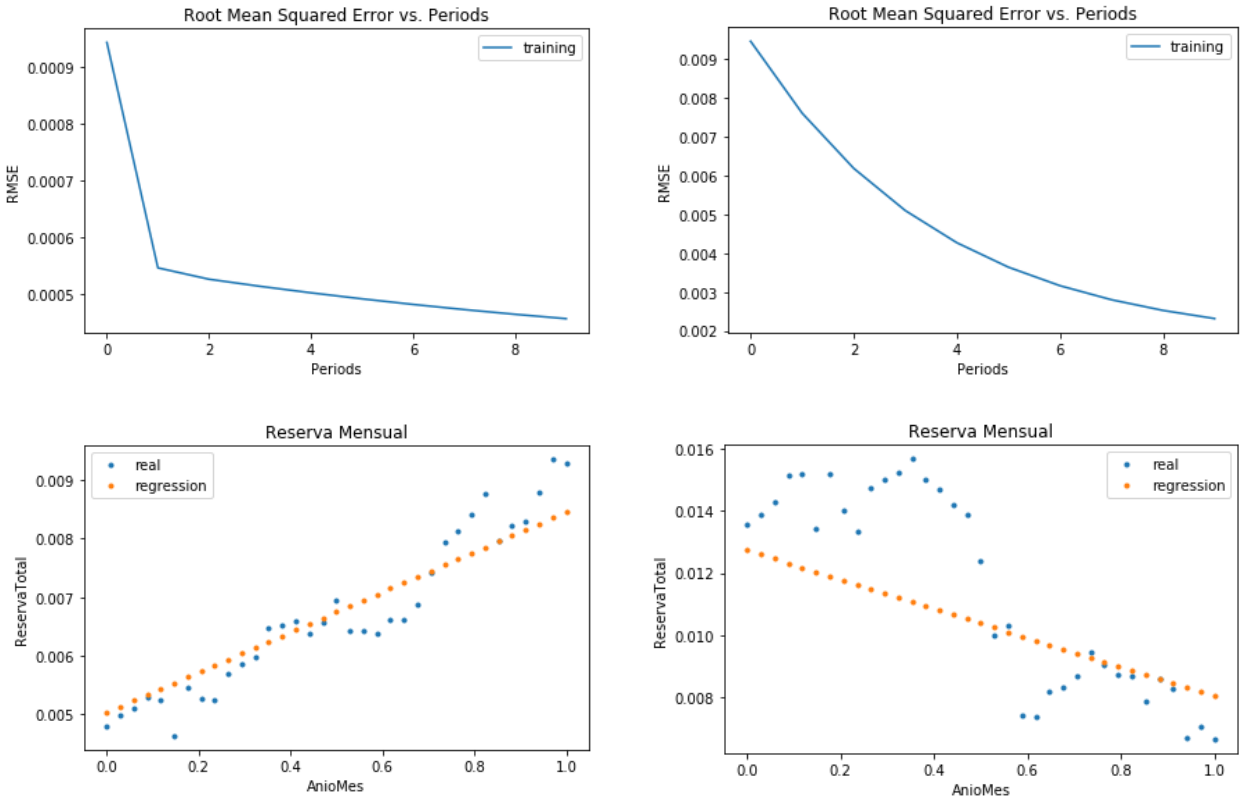


Gráfico 4.14: Entrenamiento de los modelos de regresión lineal con datos de Reserva Riesgo en Curso del ramo Vida Colectivo acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

Regresión Lineal para Reserva Combinado Familiar en Pesos

Training model...
learning_rate=0.01,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.00298
01 - 0.02 - 0.00290
02 - 0.02 - 0.00287
03 - 0.02 - 0.00285
04 - 0.02 - 0.00282
05 - 0.02 - 0.00280
06 - 0.02 - 0.00278
07 - 0.02 - 0.00277
08 - 0.02 - 0.00275
09 - 0.02 - 0.00274
Model training finished.
Total Minutes (ElapsedTime): 0.2213627696

Regresión Lineal para Reserva Combinado Familiar en Dólares

Training model...
learning_rate=0.02,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
00 - 0.02 - 0.02813
01 - 0.02 - 0.02118
02 - 0.02 - 0.01661
03 - 0.02 - 0.01362
04 - 0.02 - 0.01164
05 - 0.02 - 0.01029
06 - 0.02 - 0.00933
07 - 0.02 - 0.00861
08 - 0.02 - 0.00804
09 - 0.02 - 0.00758
Model training finished.
Total Minutes (ElapsedTime): 0.2182341456

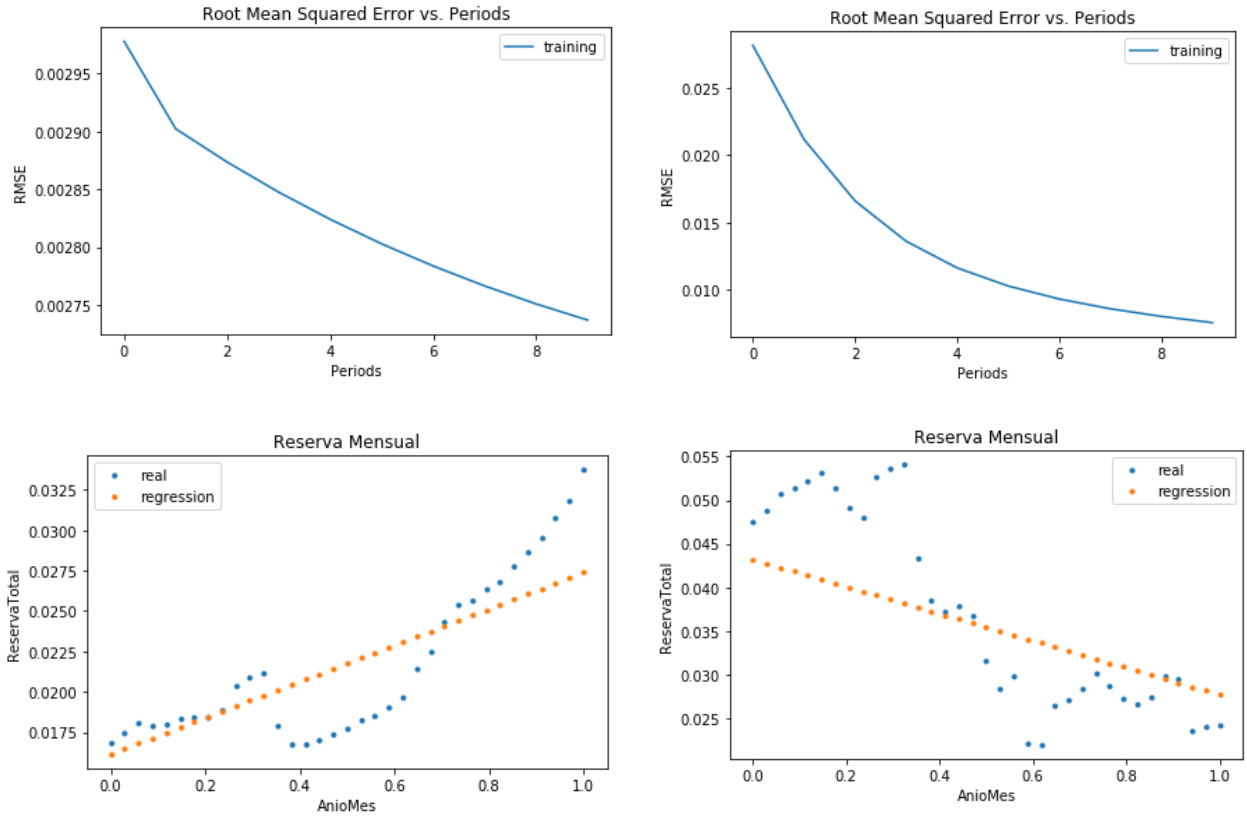


Gráfico 4.15: Entrenamiento de los modelos de regresión lineal con datos de Reserva Riesgo en Curso del ramo Combinado Familiar acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

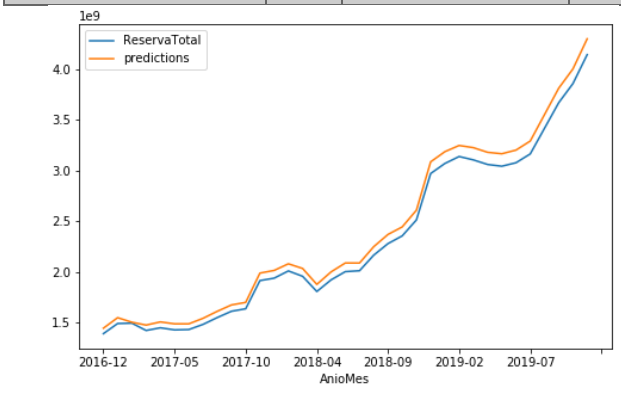
3.4.2.2 Entrenamiento de modelos ARIMA:

Como se mencionó en el **Capítulo 3.2.2**, en los mecanismos ARIMA no se utilizan los conceptos generales de los algoritmos de Machine Learning sino que está basada en técnicas estadísticas que se han especializado para trabajar y ajustarse a series temporales, con el objetivo de tener la capacidad de realizar predicciones. A continuación en el ***Gráfico 4.16*** se muestra el resumen del ajuste del mecanismo para los datos de Reserva Total tanto en Pesos como en Dólares.

ARIMA para Reserva Total en Pesos

Dep. Variable:	**ReservaTotal**		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Tue, 29 Sep 2020		
Time:	06:47:58		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-619.262		
AIC	1244.524		
BIC	1248.826		
HQIC	1245.926		
	coef	**std err**	**z**
ar.L1	1.0387	0.008	136.950
ar.S.L3	-0.0346	0.182	-0.190
sigma2	1.263e+16	3.65e-18	3.46e+33
	P>|z|	**[0.025**	**0.975]**
ar.L1	0.000	1.024	1.054
ar.S.L3	0.849	-0.392	0.323
sigma2	0.000	1.26e+16	1.26e+16

Ljung-Box (L1) (Q):	2.54	**Jarque-Bera (JB):**	4.95
Prob(Q):	0.11	**Prob(JB):**	0.08
Heteroskedasticity (H):	2.14	**Skew:**	-0.82
Prob(H) (two-sided):	0.24	**Kurtosis:**	4.07



ARIMA para Reserva Total en Dólares

Dep. Variable:	**ReservaTotalU**		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Wed, 30 Sep 2020		
Time:	10:06:32		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-532.410		
AIC	1070.821		
BIC	1075.123		
HQIC	1072.223		
	coef	**std err**	**z**
ar.L1	0.9865	0.016	60.328
ar.S.L3	0.0525	0.148	0.356
sigma2	4.543e+13	nan	nan
	P>|z|	**[0.025**	**0.975]**
ar.L1	0.000	0.954	1.019
ar.S.L3	0.722	-0.237	0.342
sigma2	nan	nan	nan

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	0.15
Prob(Q):	0.99	**Prob(JB):**	0.93
Heteroskedasticity (H):	0.90	**Skew:**	-0.09
Prob(H) (two-sided):	0.88	**Kurtosis:**	3.29

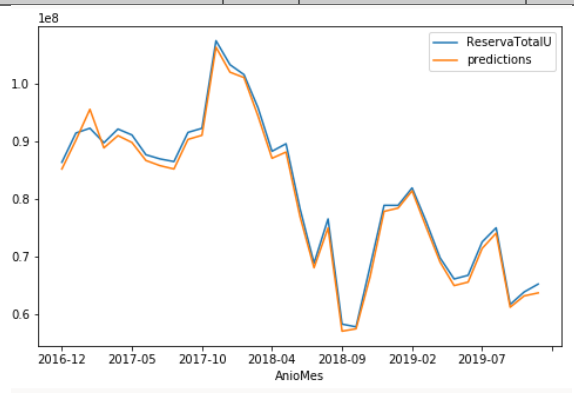


Gráfico 4.16: Entrenamiento de los modelos ARIMA con datos de Reserva Riesgo en Curso acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

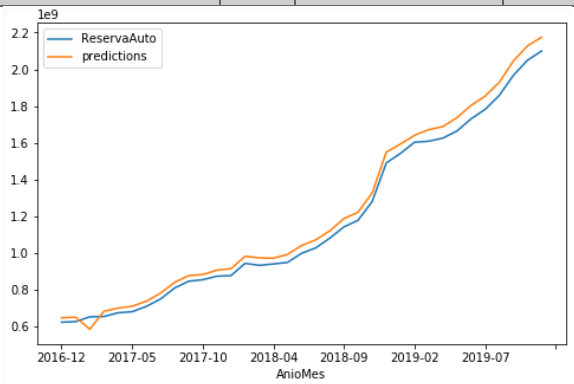
Visualmente se denota un ajuste del modelo a los datos mucho más cercano que los modelos de regresión lineal, pudiendo reproducir en el modelo no solamente en la tendencia como era el caso anterior, sino también en las fluctuaciones que sufren los datos fundamentalmente relacionada a la estacionalidad.

El mismo proceso se hizo a continuación con los datos filtrados por ramos particulares de Seguros de Automotores, Vida Colectivo y Combinado familiar, mostrándose los resultados en los ***Gráficos 4.17, 4.18 y 4.19*** respectivamente.

ARIMA para Reserva Automotores en Pesos

Dep. Variable:	**ReservaAuto**		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Wed, 30 Sep 2020		
Time:	10:49:42		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-585.361		
AIC	1176.722		
BIC	1181.024		
HQIC	1178.124		
	coef	**std err**	**z**
ar.L1	1.0386	0.005	200.885
ar.S.L3	-0.1496	0.246	-0.609
sigma2	1.401e+15	1.78e-16	7.86e+30
	P>|z|	**[0.025**	**0.975]**
ar.L1	0.000	1.028	1.049
ar.S.L3	0.543	-0.631	0.332
sigma2	0.000	1.4e+15	1.4e+15

Ljung-Box (L1) (Q):	2.05	**Jarque-Bera (JB):**	116.81
Prob(Q):	0.15	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.76	**Skew:**	2.45
Prob(H) (two-sided):	0.39	**Kurtosis:**	11.15



ARIMA para Reserva Automotores en Dólares

Dep. Variable:	**ReservaAutoU**		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Wed, 30 Sep 2020		
Time:	11:59:49		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-507.632		
AIC	1021.263		
BIC	1025.565		
HQIC	1022.665		
	coef	**std err**	**z**
ar.L1	0.9911	0.014	69.903
ar.S.L3	0.0066	0.136	0.048
sigma2	9.093e+12	1.85e-15	4.93e+27
	P>|z|	**[0.025**	**0.975]**
ar.L1	0.000	0.963	1.019
ar.S.L3	0.961	-0.260	0.273
sigma2	0.000	9.09e+12	9.09e+12

Ljung-Box (L1) (Q):	0.08	**Jarque-Bera (JB):**	4.95
Prob(Q):	0.77	**Prob(JB):**	0.08
Heteroskedasticity (H):	3.30	**Skew:**	-0.82
Prob(H) (two-sided):	0.07	**Kurtosis:**	4.07

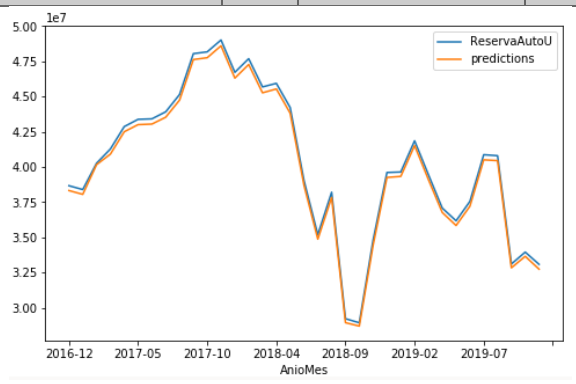
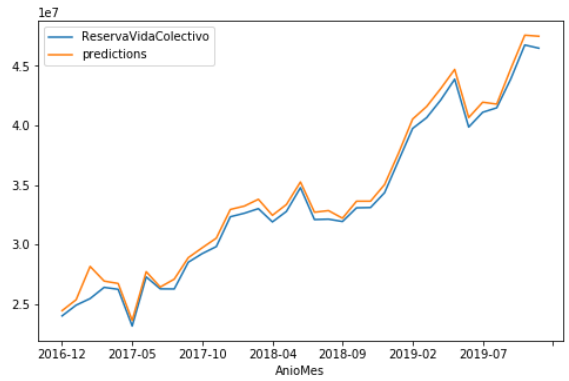


Gráfico 4.17: Entrenamiento de los modelos ARIMA con datos de Reserva Riesgo en Curso del ramo Automotores acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

ARIMA para Reserva Vida Colectivo en Pesos

Dep. Variable:	ReservaVidaColectivo		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Wed, 30 Sep 2020		
Time:	13:01:19		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-489.723		
AIC	985.447		
BIC	989.749		
HQIC	986.849		
	coef	std err	z
ar.L1	1.0183	0.009	112.685
ar.S.L3	0.0929	0.198	0.470
sigma2	2.837e+12	3.27e-16	8.68e+27
	P>|z|	[0.025	0.975]
ar.L1	0.000	1.001	1.036
ar.S.L3	0.638	-0.294	0.480
sigma2	0.000	2.84e+12	2.84e+12

Ljung-Box (L1) (Q):	1.22	Jarque-Bera (JB):	3.22
Prob(Q):	0.27	Prob(JB):	0.20
Heteroskedasticity (H):	1.13	Skew:	-0.68
Prob(H) (two-sided):	0.85	Kurtosis:	3.79



ARIMA para Reserva Vida Colectivo en Dólares

Dep. Variable:	ReservaVidaColectivoU		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Mon, 05 Oct 2020		
Time:	06:44:30		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-401.330		
AIC	808.661		
BIC	812.963		
HQIC	1072.223		
	coef	std err	z
ar.L1	0.9633	0.027	35.139
ar.S.L3	0.3845	0.162	2.373
sigma2	1.162e+10	1.78e-12	6.51e+21
	P>|z|	[0.025	0.975]
ar.L1	0.000	0.910	1.017
ar.S.L3	0.018	0.067	0.702
sigma2	0.000	1.16e+10	1.16e+10

Ljung-Box (L1) (Q):	0.38	Jarque-Bera (JB):	1.57
Prob(Q):	0.54	Prob(JB):	0.46
Heteroskedasticity (H):	0.33	Skew:	-0.55
Prob(H) (two-sided):	0.09	Kurtosis:	2.86

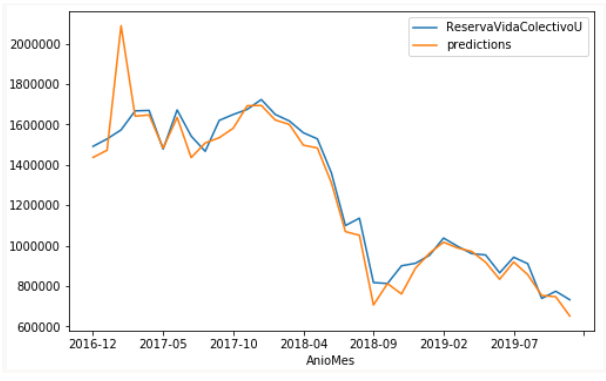
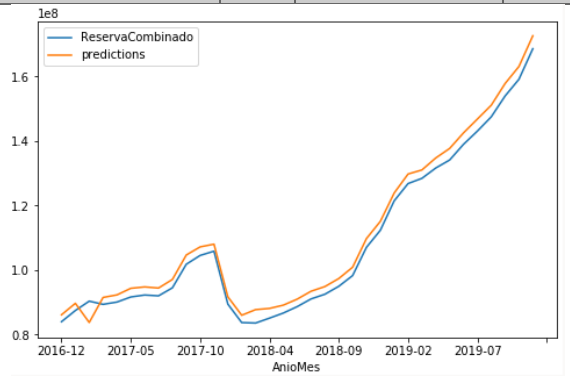


Gráfico 4.18: Entrenamiento de los modelos ARIMA con datos de Reserva Riesgo en Curso del ramo Vida Colectivo acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

ARIMA para Reserva Combinado Familiar en Pesos

Dep. Variable:	ReservaCombinado		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Mon, 05 Oct 2020		
Time:	07:10:35		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-518.305		
AIC	1042.611		
BIC	1046.913		
HQIC	1044.013		
	coef	std err	z
ar.L1	1.0255	0.008	128.756
ar.S.L3	-0.1061	0.215	-0.493
sigma2	1.832e+13	1.22e-14	1.5e+27
	P>|z|	[0.025	0.975]
ar.L1	0.000	1.010	1.041
ar.S.L3	0.622	-0.528	0.315
sigma2	0.000	1.83e+13	1.83e+13

Ljung-Box (L1) (Q):	6.99	Jarque-Bera (JB):	95.89
Prob(Q):	0.01	Prob(JB):	0.00
Heteroskedasticity (H):	0.12	Skew:	-2.15
Prob(H) (two-sided):	0.00	Kurtosis:	10.47



ARIMA para Reserva Combinado Familiar en Dólares

Dep. Variable:	ReservaCombinadoU		
Model:	SARIMAX(1, 0, 0)x(1, 0, 0, 3)		
Date:	Mon, 05 Oct 2020		
Time:	07:34:35		
Sample:	0		
	- 35		
Covariance Type:	opg		
No. Observations:	35		
Log Likelihood	-439.367		
AIC	884.733		
BIC	889.035		
HQIC	886.135		
	coef	std err	z
ar.L1	0.9751	0.013	72.537
ar.S.L3	-0.1374	0.169	-0.812
sigma2	1.177e+11	1.3e-14	9.06e+24
	P>|z|	[0.025	0.975]
ar.L1	0.000	0.949	1.001
ar.S.L3	0.417	-0.469	0.194
sigma2	0.000	1.18e+11	1.18e+11

Ljung-Box (L1) (Q):	0.50	Jarque-Bera (JB):	5.75
Prob(Q):	0.48	Prob(JB):	0.06
Heteroskedasticity (H):	0.35	Skew:	-0.93
Prob(H) (two-sided):	0.11	Kurtosis:	4.00

Gráfico 4.19: Entrenamiento de los modelos ARIMA con datos de Reserva Riesgo en Curso del ramo Combinado Familiar acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

3.4.2.3 Entrenamiento de modelos LSTM:

Por último, se realizó el entrenamiento de una Red Neuronal tipo LSTM. Como la cantidad de datos de entrenamiento de la serie de tiempo era reducida, a medida que se realizaban pruebas con diferente número de neuronas, los tiempos de procesamiento no aumentaban

significativamente, por lo que se optó por agregar una cantidad de neuronas lo suficientemente alta que permita minimizar el error, resultando modelos de 100 nodos aproximadamente.

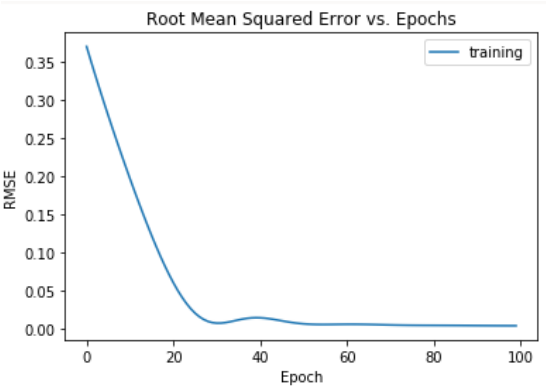
En los ***Gráficos 4.20, 4.21, 4.22 y 4.23*** se muestra el resumen del proceso entrenamiento para los modelos de redes LSTM de cada uno de los sets de datos (Reserva Total, Automotores, Vida Colectivo y Combinado Familiar).

LSTM para Reserva Total en Pesos

```
Training model...
Epoch 1/100 - 1s 32ms/step - loss: 0.3706 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.3516 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.3331 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.3150 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.2973 - acc: 0.0286
Epoch 6/100 - 0s 3ms/step - loss: 0.2799 - acc: 0.0286
...
Epoch 98/100 - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Epoch 99/100 - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Model training finished.
```

LSTM para Reserva Total en Dólares

```
Training model...
Epoch 1/100 - 1s 33ms/step - loss: 0.5791 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.5528 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.5272 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.5021 - acc: 0.0286
Epoch 5/100 - 0s 4ms/step - loss: 0.4775 - acc: 0.0286
Epoch 6/100 - 0s 4ms/step - loss: 0.4535 - acc: 0.0286
...
Epoch 98/100 - 0s 3ms/step - loss: 0.0221 - acc: 0.0286
Epoch 99/100 - 0s 3ms/step - loss: 0.0221 - acc: 0.0286
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0221 - acc: 0.0286
Model training finished.
```



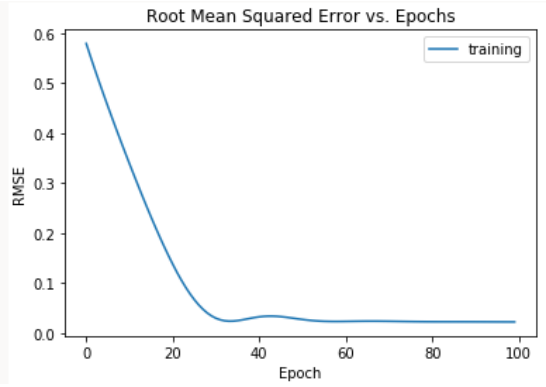


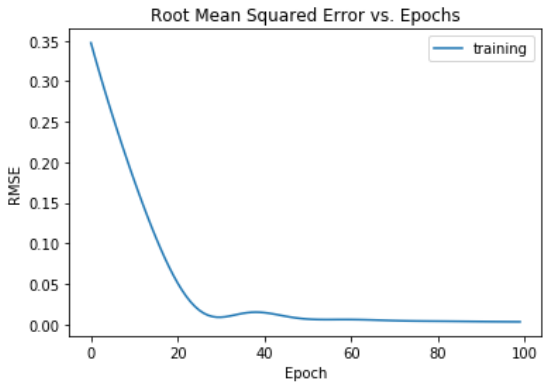
Gráfico 4.20: Entrenamiento de los modelos de redes LSTM con datos de Reserva Riesgo en Curso Total acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

LSTM para Reserva Automotores en Pesos

```
Training model...
Epoch 1/100 - 1s 27ms/step - loss: 0.3475 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.3284 - acc: 0.0286
Epoch 3/100 - 0s 2ms/step - loss: 0.3099 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.2919 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.2744 - acc: 0.0286
Epoch 6/100 - 0s 3ms/step - loss: 0.2573 - acc: 0.0286
...
Epoch 98/100 - 0s 3ms/step - loss: 0.0032 - acc: 0.0571
Epoch 99/100 - 0s 3ms/step - loss: 0.0031 - acc: 0.0571
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0031 - acc: 0.0571
Model training finished.
```

LSTM para Reserva Automotores en Dólares

```
Training model...
Epoch 1/100 - 1s 31ms/step - loss: 0.6683 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.6370 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.6066 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.5768 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.5477 - acc: 0.0286
Epoch 6/100 - 0s 3ms/step - loss: 0.5191 - acc: 0.0286
...
Epoch 98/100 - 0s 4ms/step - loss: 0.0224 - acc: 0.0286
Epoch 99/100 - 0s 3ms/step - loss: 0.0224 - acc: 0.0286
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0223 - acc: 0.0286
Model training finished.
```



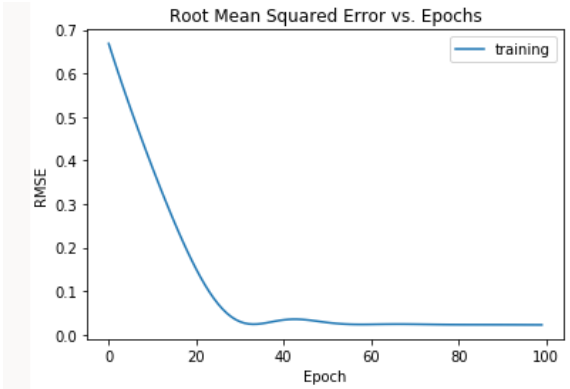


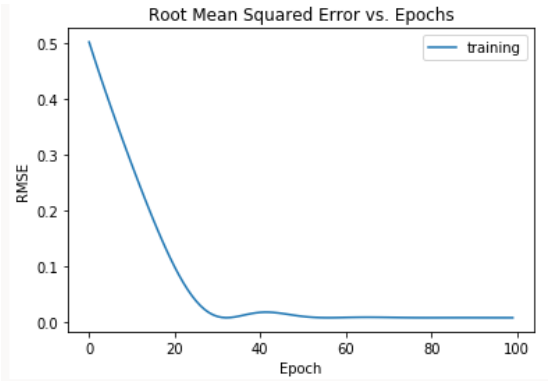
Gráfico 4.21: Entrenamiento de los modelos de redes LSTM con datos de Reserva Riesgo en Curso del ramo Automotores acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

LSTM para Reserva Vida Colectivo en Pesos

```
Training model...
Epoch 1/100 - 1s 35ms/step - loss: 0.5030 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.4791 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.4558 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.4329 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.4104 - acc: 0.0286
Epoch 6/100 - 0s 3ms/step - loss: 0.3883 - acc: 0.0286
...
Epoch 98/100 - 0s 3ms/step - loss: 0.0081 - acc: 0.0571
Epoch 99/100 - 0s 2ms/step - loss: 0.0081 - acc: 0.0571
Epoch 100/100/35 - 0s 2ms/step - loss: 0.0081 - acc: 0.0571
Model training finished.
```

LSTM para Reserva Vida Colectivo en Dólares

```
Training model...
Epoch 1/100 - 1s 32ms/step - loss: 0.5629 - acc: 0.0286
Epoch 2/100 - 0s 2ms/step - loss: 0.5351 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.5080 - acc: 0.0286
Epoch 4/100 - 0s 2ms/step - loss: 0.4816 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.4558 - acc: 0.0286
Epoch 6/100 - 0s 3ms/step - loss: 0.4306 - acc: 0.0286
...
Epoch 98/100 - 0s 3ms/step - loss: 0.0243 - acc: 0.0286
Epoch 99/100 - 0s 3ms/step - loss: 0.0243 - acc: 0.0286
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0243 - acc: 0.0286
Model training finished.
```



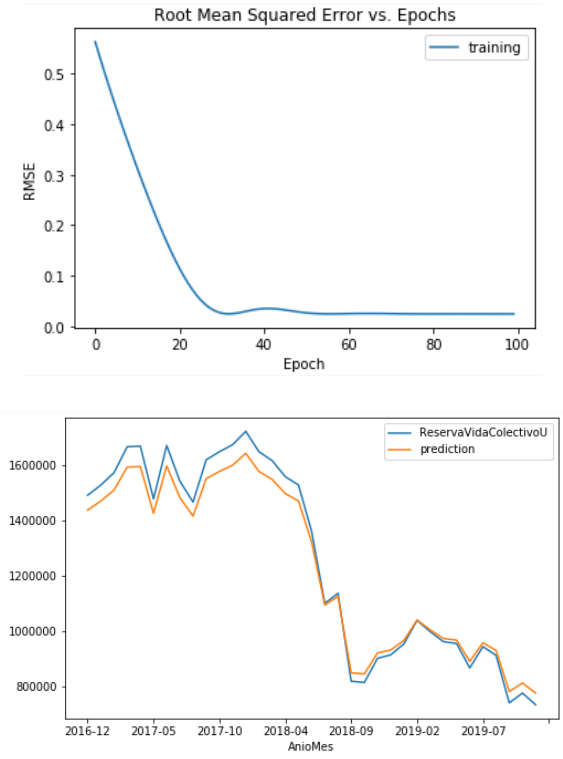
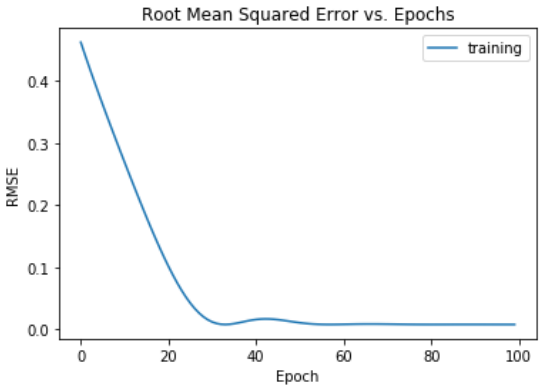
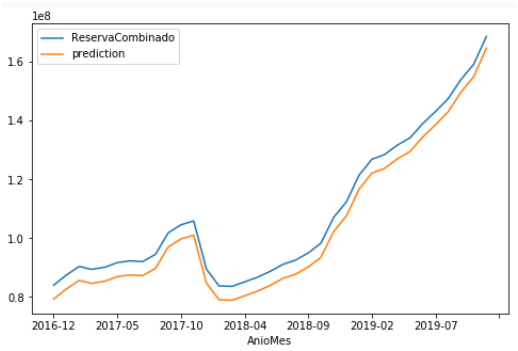


Gráfico 4.22: Entrenamiento de los modelos de redes LSTM con datos de Reserva Riesgo en Curso del ramo Vida Colectivo acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

LSTM para Reserva Combinado Familiar en Pesos

```
Training model...
Epoch 1/100 - 1s 26ms/step - loss: 0.4616 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.4407 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.4203 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.4003 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.3807 - acc: 0.0286
Epoch 6/100 - 0s 2ms/step - loss: 0.3614 - acc: 0.0286
...
Epoch 98/100 - 0s 3ms/step - loss: 0.0081 - acc: 0.0286
Epoch 99/100 - 0s 4ms/step - loss: 0.0081 - acc: 0.0571
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0081 - acc: 0.0571
Model training finished.
```





LSTM para Reserva Combinado Familiar en Dólares

```
Training model...
Epoch 1/100 - 1s 29ms/step - loss: 0.5085 - acc: 0.0286
Epoch 2/100 - 0s 3ms/step - loss: 0.4840 - acc: 0.0286
Epoch 3/100 - 0s 3ms/step - loss: 0.4603 - acc: 0.0286
Epoch 4/100 - 0s 3ms/step - loss: 0.4371 - acc: 0.0286
Epoch 5/100 - 0s 3ms/step - loss: 0.4145 - acc: 0.0286
Epoch 6/100 - 0s 3ms/step - loss: 0.3924 - acc: 0.0286
...
Epoch 98/100 - 0s 4ms/step - loss: 0.0242 - acc: 0.0286
Epoch 99/100 - 0s 3ms/step - loss: 0.0242 - acc: 0.0286
Epoch 100/100/35 - 0s 3ms/step - loss: 0.0242 - acc: 0.0286
Model training finished.
```

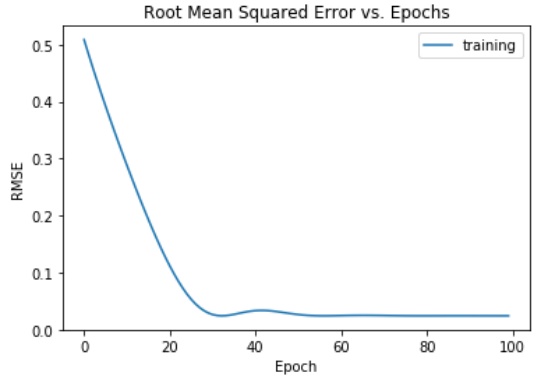


Gráfico 4.23: Entrenamiento de los modelos de redes LSTM con datos de Reserva Riesgo en Curso del ramo Combinado Familiar acumulada en forma mensual. Un modelo fue entrenado con datos en pesos argentino y el otro con datos convertidos a dólares estadounidenses.

3.4.2.4 Comparativas entre modelos

Para finalizar la etapa de entrenamientos, y antes de iniciar el análisis de los diferentes errores que va a permitir una comparación cuantitativa de los modelos, se plantea realizar una comparación cualitativa a través de una exploración visual de los resultados obtenidos en un mismo espacio de gráficas.

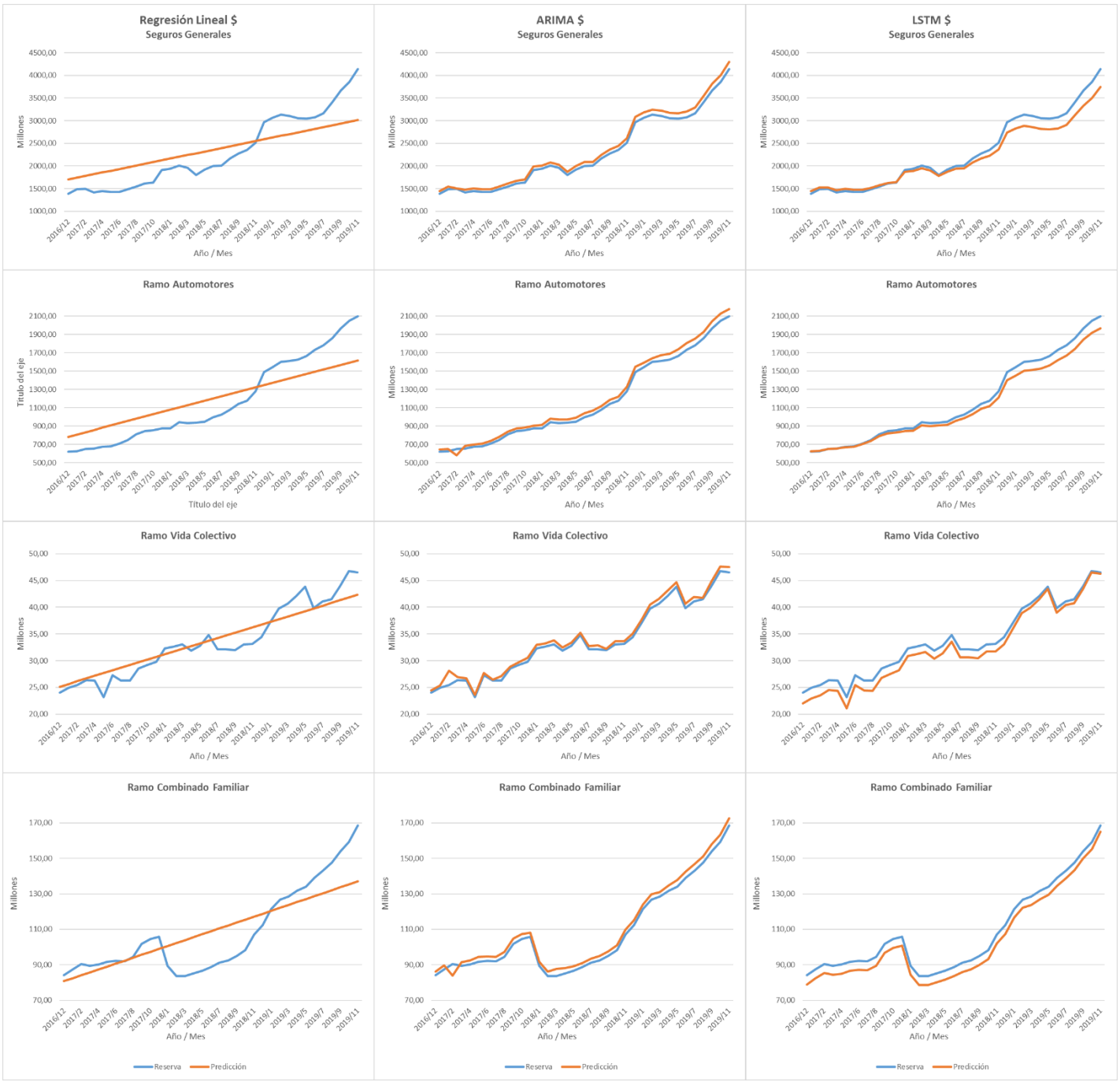


Gráfico 4.24: Curvas de Reserva Riesgo en Curso y las predicciones de los modelos entrenados en pesos argentinos. Se grafican tanto a nivel general como por ramo (automotores, vida colectivo y combinado familiar).

En el ***Gráfico 4.24*** se proyecta una matriz con las curvas de los datos y de las predicciones de los tres modelos expresados en pesos argentinos. Las columnas de la matriz representan los tres tipos de algoritmos utilizados (Regresión Lineal, ARIMA y Redes LSTM), y las filas corresponden a los diferentes sets de datos en moneda nacional, sobre los cuales se realizó el entrenamiento (Reserva Total, Ramo Automotor, Vida Colectivo y Combinado Familiar).

Visualmente se nota que los modelos de regresión lineal si bien siguen efectivamente la tendencia de los datos, no pueden ajustarse a las diferentes fluctuaciones generadas principalmente a partir de la estacionalidad natural de una serie temporal. En contrapartida, los modelos especializados en series temporales tienen un ajuste mucho más cercano, ya que al manejar esas particularidades pueden hacer un acompañamiento de la curva y manejar los vaivenes de cada ciclo.

A continuación en el ***Gráfico 4.25*** se muestran las curvas de los modelos entrenados en dólares, con el adicional del modelo combinado entrenado en pesos argentinos pero convertidos a dólares a la hora de expresar las predicciones, de acuerdo a lo definido en el **Capítulo 4.4.2**.

Los modelos expresados en dólares estadounidenses se diferencian de los anteriores en que en su mayoría las curvas tienen una tendencia negativa (la única excepción es en Automotores) y mucho menos pronunciada, ya que se abstrae de los procesos inflacionarios y devaluatorios de la moneda nacional, como se analizó en el ***Gráfico 4.4*** del **Capítulo 4.2**

Resulta interesante el resultado obtenido con el modelo combinado entrenado en pesos pero con resultados expresados en dólares (segunda columna de la matriz), porque permite transformar un entrenamiento lineal en una moneda, en un modelo no lineal en otra, resultando una curva muy diferente si se compara con el modelo lineal con los datos convertidos a dólares antes del entrenamiento (comparación entre la primera y segunda columna de la matriz).

Por otro lado, al igual que en las predicciones en pesos argentinos, los modelos especializados en series temporales tuvieron un mejor ajuste a los datos en dólares que se manifiesta a simple vista en comparación con los modelos lineales.

Por último, tanto el proceso de entrenamiento como las gráficas y resultados fue muy similar si comparamos las corridas sobre los datos totalizados (primera fila de las matrices) con las ejecuciones sobre los datos filtrados sobre ramos particulares (segunda, tercera y cuarta fila de las matrices). Si bien existen diferencias significativas en las magnitudes y los montos, cuestión que requirió en algunos casos ir variando la parametrización de los algoritmos al realizar el entrenamiento, la forma de las curvas visualmente se muestra muy similares para todos los casos, respetando los mismos ciclos y tendencias.

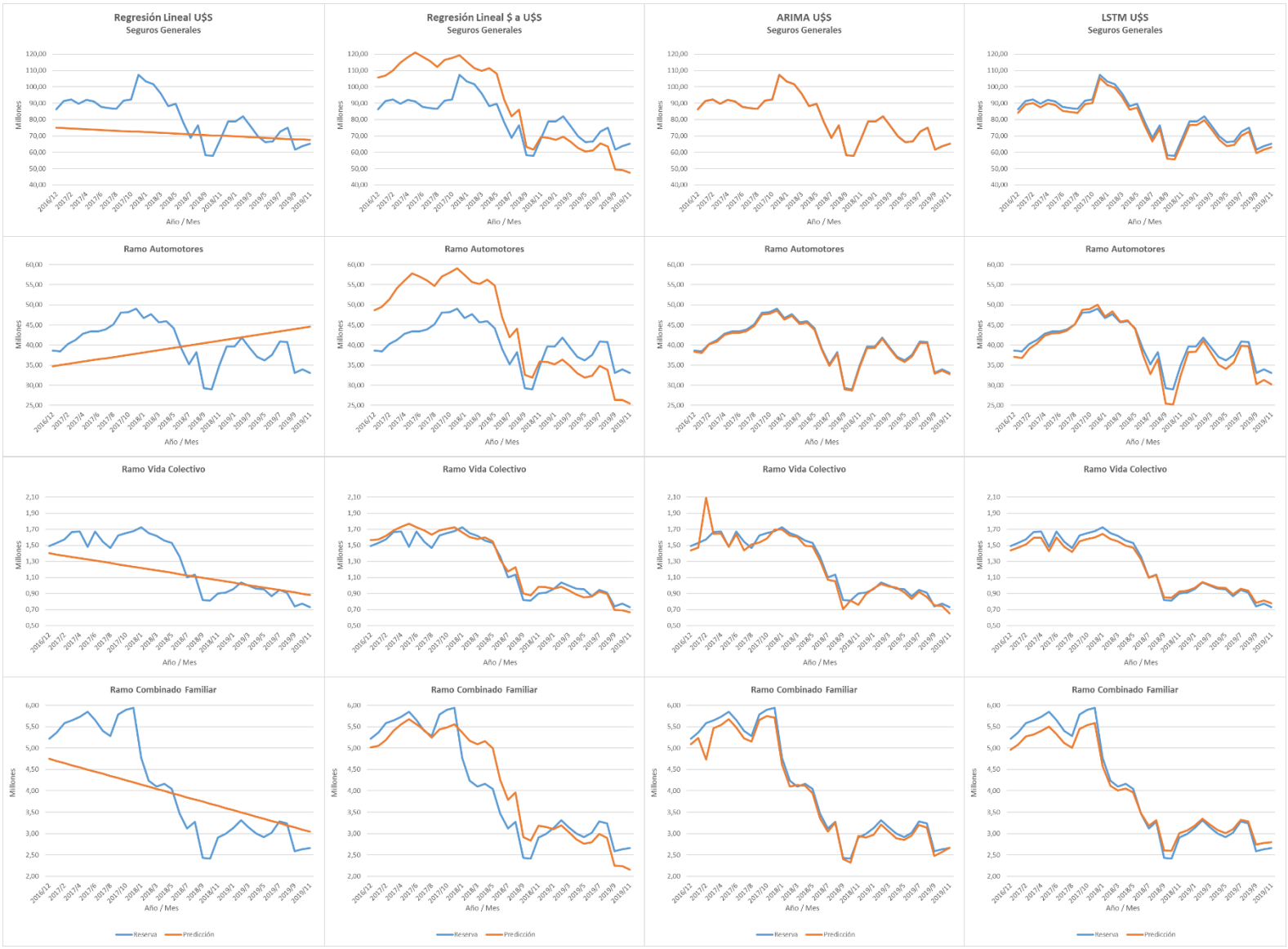


Gráfico 4.25: Curvas de Reserva Riesgo en Curso y las predicciones de los modelos expresados en dólares estadounidenses. Se grafican tanto a nivel general como por ramo (automotores, vida colectivo y combinado familiar).

3.5 ANÁLISIS DE ERRORES:

Como se mencionó en varias oportunidades, el proceso de Machine Learning es iterativo y las fases que mayor cantidad de ciclos sufren son las de **Construcción del Modelo** y **Análisis de Errores**, porque como menciona el **Capítulo 3.2.1** del presente trabajo, luego de entrenar un modelo el siguiente paso es validar los errores, compararlo con lo esperado, y volver para intentar para buscar y entrenar un modelo mejor.

Si bien el cálculo y análisis de los errores forma parte del mismo proceso de entrenamiento, éste último trabaja fundamentalmente con los errores generaros a partir del set de datos de entrenamiento y validación. En cambio ésta etapa específica se centra en el análisis de los errores calculados a partir del set de datos de testing, los cuales fueron apartados en un inicio y no fueron utilizados durante la construcción del modelo. El objetivo es tener un conjunto de datos que el modelo no conozca en absoluto, que puedan utilizarse para medir la efectividad de las predicciones que el modelo sea capaz de realizar, previniendo sobre-ajustes. En éste caso ésta separación solamente fue posible para el caso de los modelos de reserva de pólizas individuales, porque en los modelos de reservas mensuales, como se mencionó en el apartado anterior, la totalidad se utilizó para entrenamiento dada la poca cantidad de datos de la serie temporal (un dato por mes).

Por otro lado, en ésta etapa no siempre se utilizan los mismos indicadores de error que el proceso de entrenamiento. En éste caso, en el proceso de entrenamiento de todos los modelos se utilizó el indicador RMSE (Raíz de Error Cuadrático Medio) por su capacidad para potenciar los errores ya que trabaja con diferencias cuadráticas. Pero a la hora del testing y sobre todo en la comparación entre diferentes modelos que utilizan diferentes unidades de medida (monedas), se necesita un indicador que mida el error en términos relativos, por lo que se utilizó el indicador MAPE (Porcentaje de Error Absoluto Medio). En el **Capítulo 4.2** se definió como criterio de aceptación de los modelos entrenados, un MAPE inferior a 20%.

3.5.1 Análisis de Error en la Predicción de Reserva para una Póliza individual

Los primeros modelos a analizar son los entrenados a partir de datos de reserva de pólizas individuales, que tienen el objetivo de predecir el nivel de reserva que tendrá un contrato tomando como variables independientes atributos de las pólizas como la prima y el capital asegurado. En éste caso como los tres modelos trabajaron sobre datos expresados en la misma unidad de medida, en ésta etapa mantenemos el análisis sobre el error RMSE sin calcular otros indicadores. A continuación en el ***Gráfico 4.26*** se muestra la comparación del error cuadrático RMSE sobre los datos de testing de los tres modelos entrenados en la fase anterior:

1. Modelo Lineal entrenado a partir de datos de Estadística Siniestral, con una única variable independiente que es la prima.
2. Modelo Lineal entrenado a partir de datos de Reserva Riesgo en Curso, con una única variable independiente que es la prima.
3. Modelo Lineal entrenado a partir de datos de Reserva Riesgo en Curso, con dos variables independientes que son la prima y el capital asegurado.

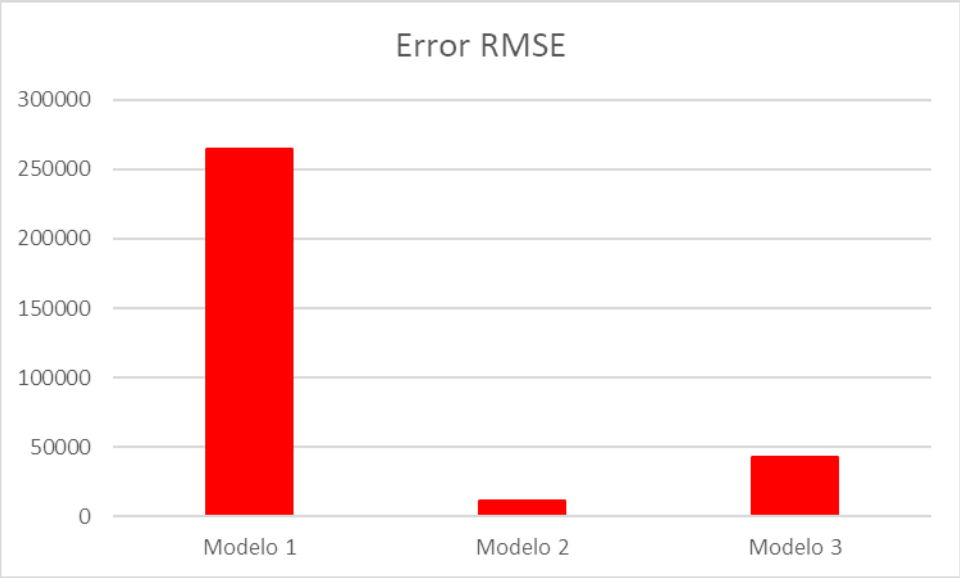


Gráfico 4.26: Error RMSE de los tres modelos de regresión lineal entrenado para predecir la reserva de pólizas individuales de automotores.

Como ya se deslumbró y profundizó en la fase de entrenamiento en el **Capítulo 4.4.1**, el MODELO 1 generado a partir de los datos de Estadística Siniestral es rápidamente descartado a partir de éstos indicadores, quedando el foco de análisis sobre los MODELOS 2 y 3 entrenados a partir de los datos de Reserva Riesgo en Curso.

Sin embargo, tratándose de pólizas individuales el error aún es alto en ambos modelos, por lo que el camino es seguir iterando y volver a las fases anteriores las veces que sean necesarias para obtener modelos con mejor precisión.

En ese camino se pueden probar modelos que involucren más variables como por ejemplo la zona de riesgo que originó el contrato, la edad y/o la actividad del cliente, o algún otro dato que pueda aportar a la correlación lineal con la variable dependiente que es la reserva. También es válido utilizar alguna combinación entre variables que aporte no linealidad al modelo.

Si no se consiguen resultados satisfactorios, se puede optar por escoger alguna otra técnica o algoritmo de Machine Learning que permita mejorar las predicciones. Como se detalló en el **Capítulo 3.2.2**, el abanico de posibilidades de técnicas a aplicar es grande, y si bien existen recomendaciones que orientan la selección de una técnica particular para un problema específico, la mayor certidumbre se obtiene sencillamente probando las diferentes alternativas con los datos disponibles, y quedándose con la que mejor se ajuste a los datos.

Si luego de esa iteración aún se sigue sin obtener resultados satisfactorios, en última instancia se puede optar por retroceder a las primeras etapas del proceso de Machine Learning y revisar la validez de los datos recolectados, e incluso llegar a revisar la problemática que se intenta resolver desde un punto de vista de negocio.

3.5.2 Análisis de Error en la Predicción de Reserva acumulada mensual

Como se mencionó en el **Capítulo 4.4.2**, los modelos entrenados para predecir reservas acumuladas tienen la particularidad de no disponer de un conjunto de datos apartados en un principio para testing, ya que por la poca cantidad de datos disponibles la totalidad se usó para entrenamiento. A continuación en el ***Gráfico 4.27*** se muestra el error RMSE, MAE y MAPE de los siete modelos construidos en la fase de entrenamiento.

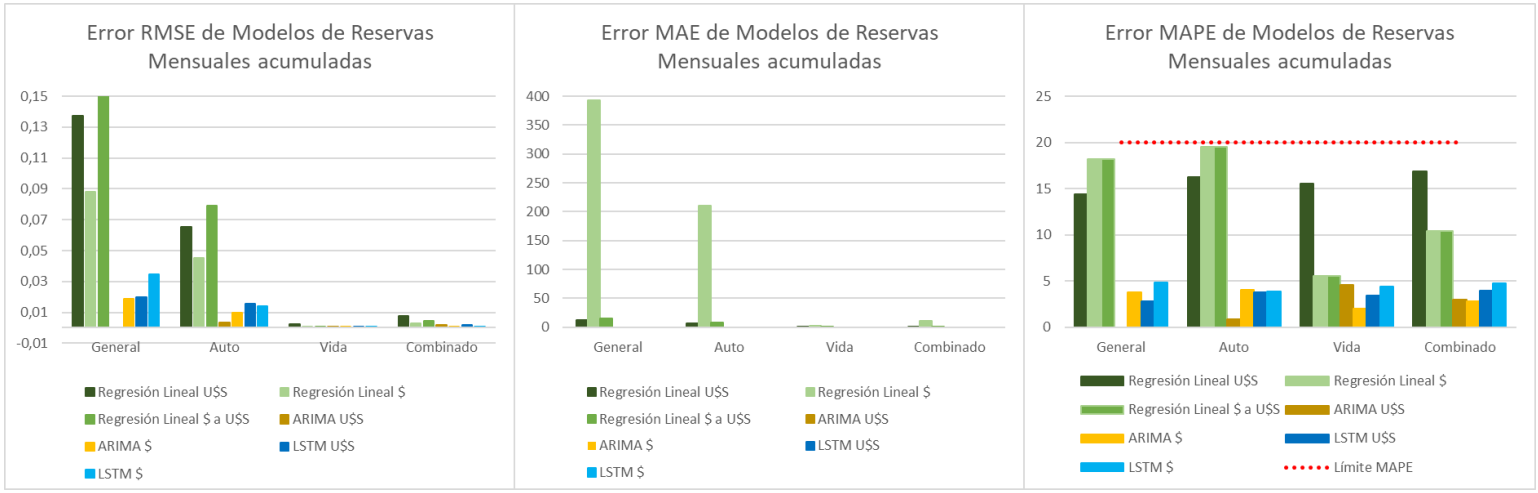


Gráfico 4.27: Error RMSE, MAE y MAPE de los siete modelos entrenados para predecir la reserva agrupada mensual en formal general y por ramo.

Como se mencionó al inicio de éste apartado, al comparar modelos con salidas expresadas en bases monetarias de diferentes órdenes de magnitud, incluso en diferentes monedas, se utilizará el indicador MAPE que tiene una naturaleza relativa y se estableció un umbral de 20% como criterio de aceptación.

Como conclusión principal, todos los modelos cumplen el criterio de aceptación de error definido por lo que se puede dar por finalizado el ciclo de entrenamiento y análisis de error y pasar a la siguiente fase que es la implementación. De todas maneras, se verifica una fuerte diferencia entre los niveles de error de los modelos de regresión lineal y los especializados en Series Temporales, representando la verificación cuantitativa de lo que ya se pudo deslumbrar visualmente en los ***Gráfico 4.24 y 4.25*** en la fase anterior.

Otras conclusiones se pueden obtener a partir del ***Gráfico 4.28*** que se presenta a continuación, en donde se muestra nuevamente la gráfica de error MAPE de todos los modelos, pero proyectada desde diferentes puntos de vista de manera de permitir hacer comparaciones más específicas y concluir sobre los algoritmos o las monedas utilizadas en los modelos.

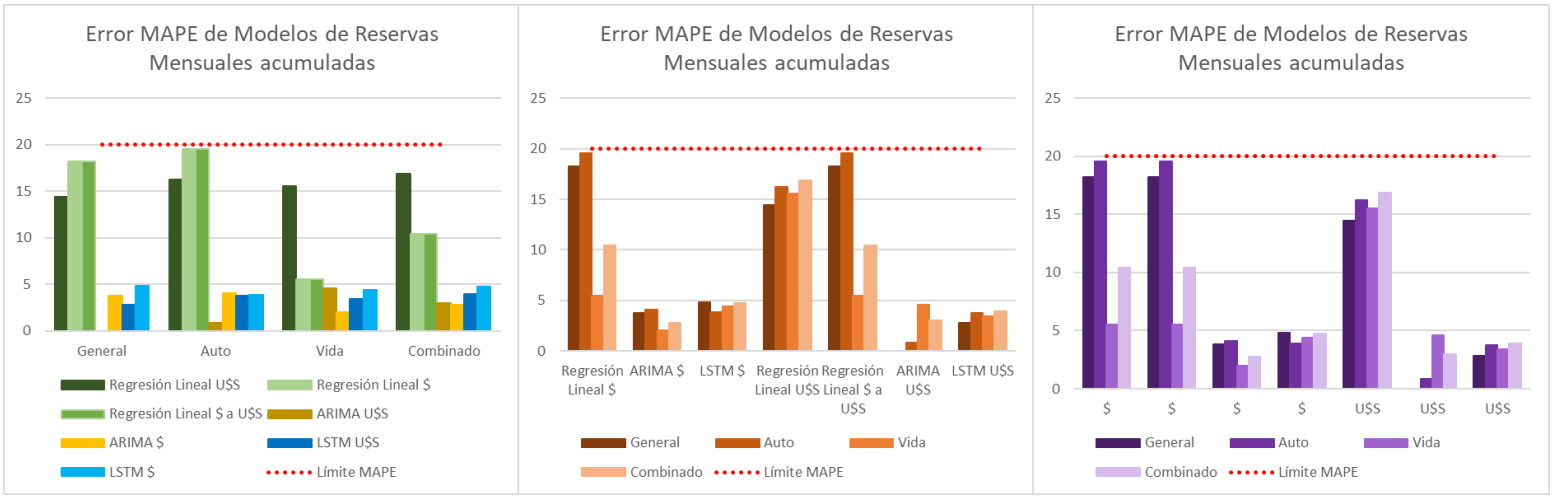


Gráfico 4.28: Error MAPE de los siete modelos entrenados para predecir la reserva agrupada mensual en formal general y por ramo, proyectada desde diferentes puntos de vista en relación a los modelos, los algoritmos y las monedas utilizadas.

Si se profundiza sobre la gráfica donde se compara el error MAPE de cada modelo (primer cuadro del ***Gráfico 4.28***), el modelo ARIMA entrenado en U$S tiene una clara diferencia en cuando a su desempeño tanto en la reserva General como en el ramo Automotor, mientras que el modelo ARIMA entrenado en $ es el que mejor funcionó para el ramo Vida Colectivo y Combinado Familiar. En todos los casos es seguido muy de cerca por los modelos LSTM, quedando relegado por amplio margen los modelos de regresión lineal, que solamente pudieron acercarse al desempeño de los anteriores sólo en los modelos entrenados en $ para el ramo Vida Colectivo (en el resto de los casos, las diferencias fueron significativas).

Si se analiza la gráfica donde se hace el paralelismo resaltando los errores generales entre los diferentes algoritmos (segundo cuadro del ***Gráfico 4.28***), se puede apreciar que los modelos ARIMA tuvieron el mejor desempeño en términos generales pero muy cerca estuvieron los LSTM, lo que refuerza la conclusión anterior. Por otro lado, si se hace foco en la gráfica donde se resalta la moneda utilizada en el proceso de entrenamiento (tercer cuadro del ***Gráfico 4.28***) se manifiesta una leve reducción del error en términos generales en los modelos entrenados sobre datos en dólares en comparación con los que trabajaron sobre los datos en moneda nacional.

Por último, a continuación en el ***Gráfico 4.29*** se muestra el componente de error MAPE desagregado en forma mensual a lo largo de toda la serie de tiempo. En el gráfico se incluyeron todos los modelos entrenados dejando afuera el modelo combinado con entrenamiento en pesos y predicción en dólares, porque tiene exactamente el mismo MAPE que su correspondiente en pesos, siendo natural porque ambos modelos tuvieron el mismo proceso de entrenamiento y por lo tanto tienen el mismo error relativo independientemente de la medida en que se expresen los resultados.

En la gráfica las curvas de los modelos especializados en series temporales se manifiestan muy por debajo del umbral durante toda la serie, mientras que los modelos de regresión lineal

manifiestan picos o momentos en donde se posicionan por encima del límite. Un criterio de aceptación más riguroso hubiese generado que éstos modelos sigan iterando tratando de conseguir predicciones de mejor calidad o bien se hubieran descartado completamente.

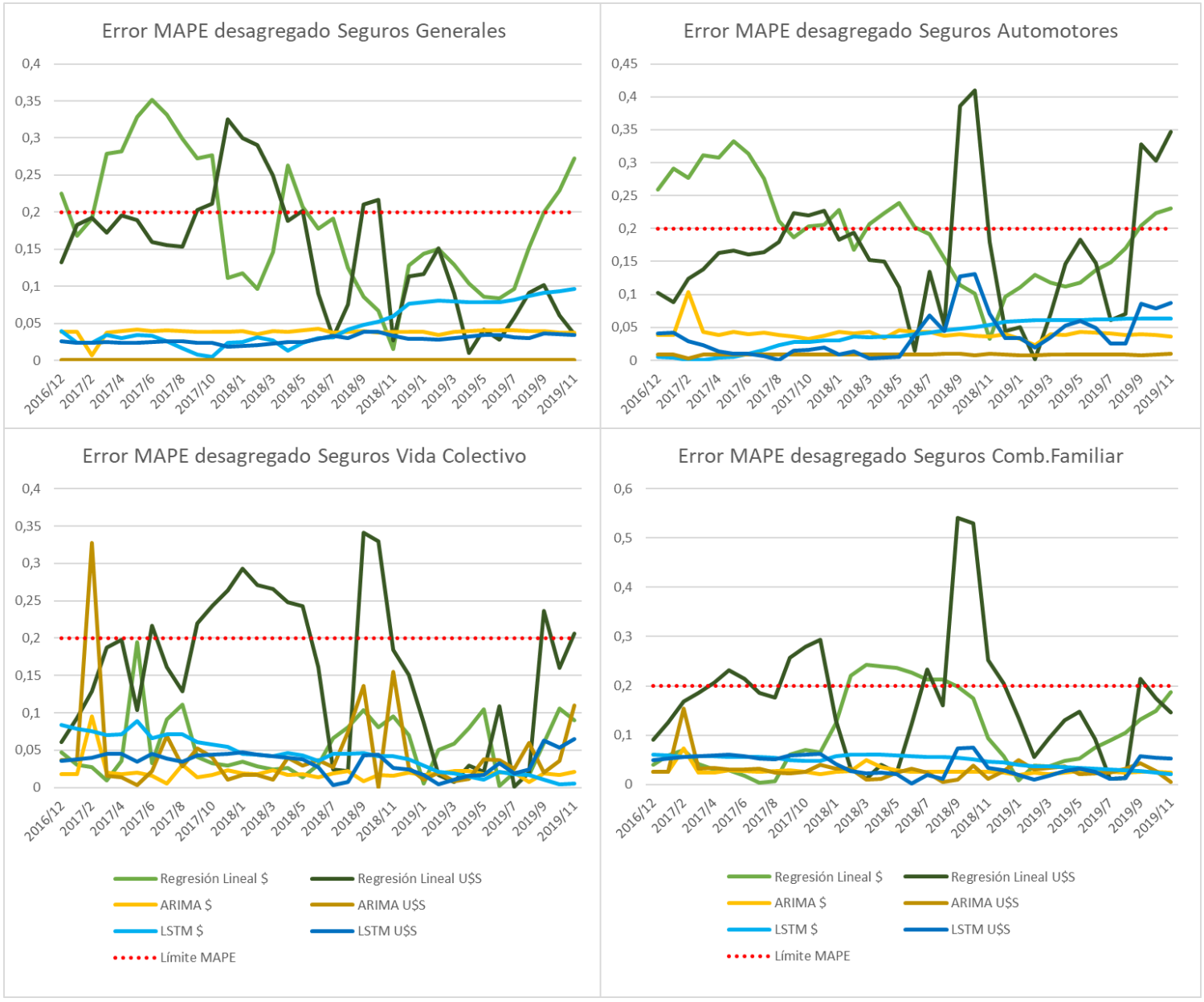


Gráfico 4.29: Error MAPE desagregado para cada mes de la serie de tiempo en cada modelo.

3.6 INTEGRACIÓN CON LOS SISTEMAS DE PRODUCCIÓN

El último paso del proceso de Machine Learning consiste en integrar los modelos que cumplieron los criterios de aceptación a los sistemas productivos de la empresa para poder utilizar efectivamente sus predicciones en los procesos de negocio. En éste caso, los modelos que quedaron por debajo del umbral de error definido en la fase donde se definió el criterio de aceptación, fueron los modelos predictivos de reservas acumuladas, mientras que los de reserva individual deben seguir iterando hasta conseguir modelos de mejor calidad de predicciones.

De ésta manera, los siete modelos de predicción de reservas acumuladas pueden ser implementados como un estimador más en los análisis financieros y de producto que se realizan normalmente en la empresa aseguradora, y contrastados con los procesos batchs tradicionales que realizan los cálculos como lo indica la ley al final de cada período.

3.6.1 Desempeño de los modelos en Producción

En los ***Gráficos 4.30*** y ***4.31*** se muestra el funcionamiento de los modelos entrenados en pesos y dólares respectivamente, durante los seis meses posteriores a la finalización de sus procesos de entrenamiento, contrastados con los datos reales que se fueron obteniendo mes a mes de los procesos tradicionales.

Éstas gráficas representan la continuación en el tiempo de los ***Gráficos 4.24 y 4.25*** del apartado anterior, manifestando la potencialidad de cada modelo para predecir la realidad. Las líneas punteadas representan las predicciones a futuro de los modelos, y la curva oscura la realidad que fue aconteciendo. Ésta contrastación es un verdadero proceso de testing para los modelos de Machine Learning que, como se mencionó en su momento, no se han apartado datos para testing originalmente dada la poca cantidad de datos disponibles para entrenamiento.

En una exploración visual, prácticamente todos los modelos manifiestan una importante separación entre las predicciones y la realidad a medida que transcurren los meses y se alejan de los entrenamientos. Con diferentes grados de intensidad, en todos los casos la curva de datos reales tiene un marcado salto que prácticamente ningún modelo pudo predecir. Ese salto es muy marcado principalmente en las gráficas de las reservas del ramo Vida colectivo, ya que a partir de Enero de 2020 existieron modificaciones normativas relacionadas con el salario mínimo vital y móvil y el valor asignado al seguro por la Superintendencia de Seguros de la Nación. Por otro lado, los datos fundamentalmente a partir de abril se generaron en el contexto de la pandemia de COVID-19, que generó un impacto general en todas las actividades comerciales de todo el mundo.

Éste tipo de cambios normativos y episodios nacionales o mundiales, tienen la característica de ser impredecibles y de romper con todas las estimaciones y objetivos que tenían las empresas para el período, obligándolas a replanificar y ajustarse al nuevo contexto. En el caso de los modelos de Machine Learning, ese ajuste implica necesariamente la revisión de los datos y de los algoritmos, y volver nuevamente a fases de definición, entrenamiento y análisis de error.

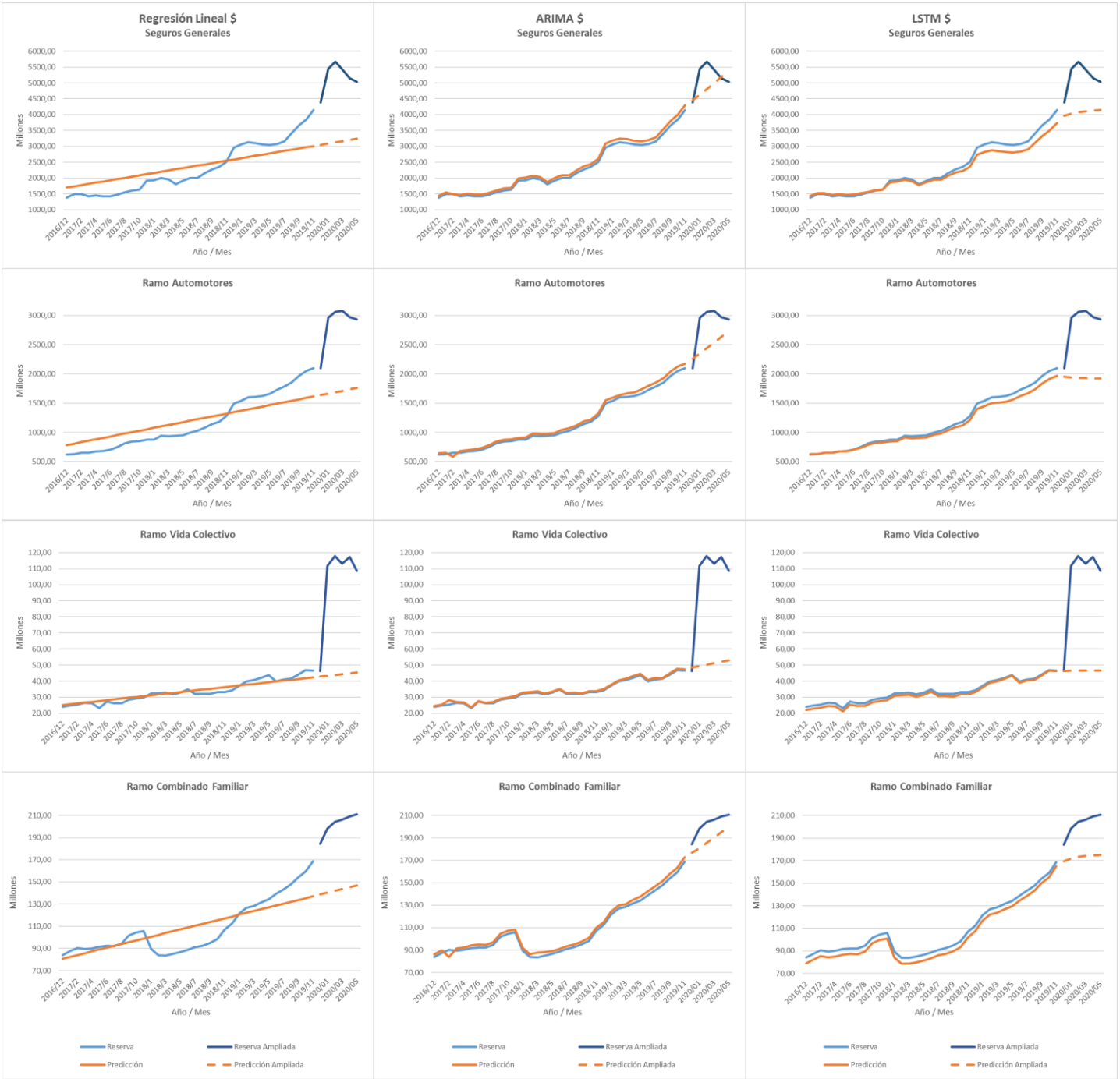


Gráfico 4.30: Desempeño de los modelos de Machine Learning entrenados con datos de Reserva Riesgo en Curso en pesos argentinos entre Dic-2016 y Nov-2019, para predecir los siguientes seis meses, entre Dic-2019 y May-2020. Las predicciones se compararon con los datos reales que fueron acontenciendo en ese período, tanto para Reservas generales como por ramo de producto.

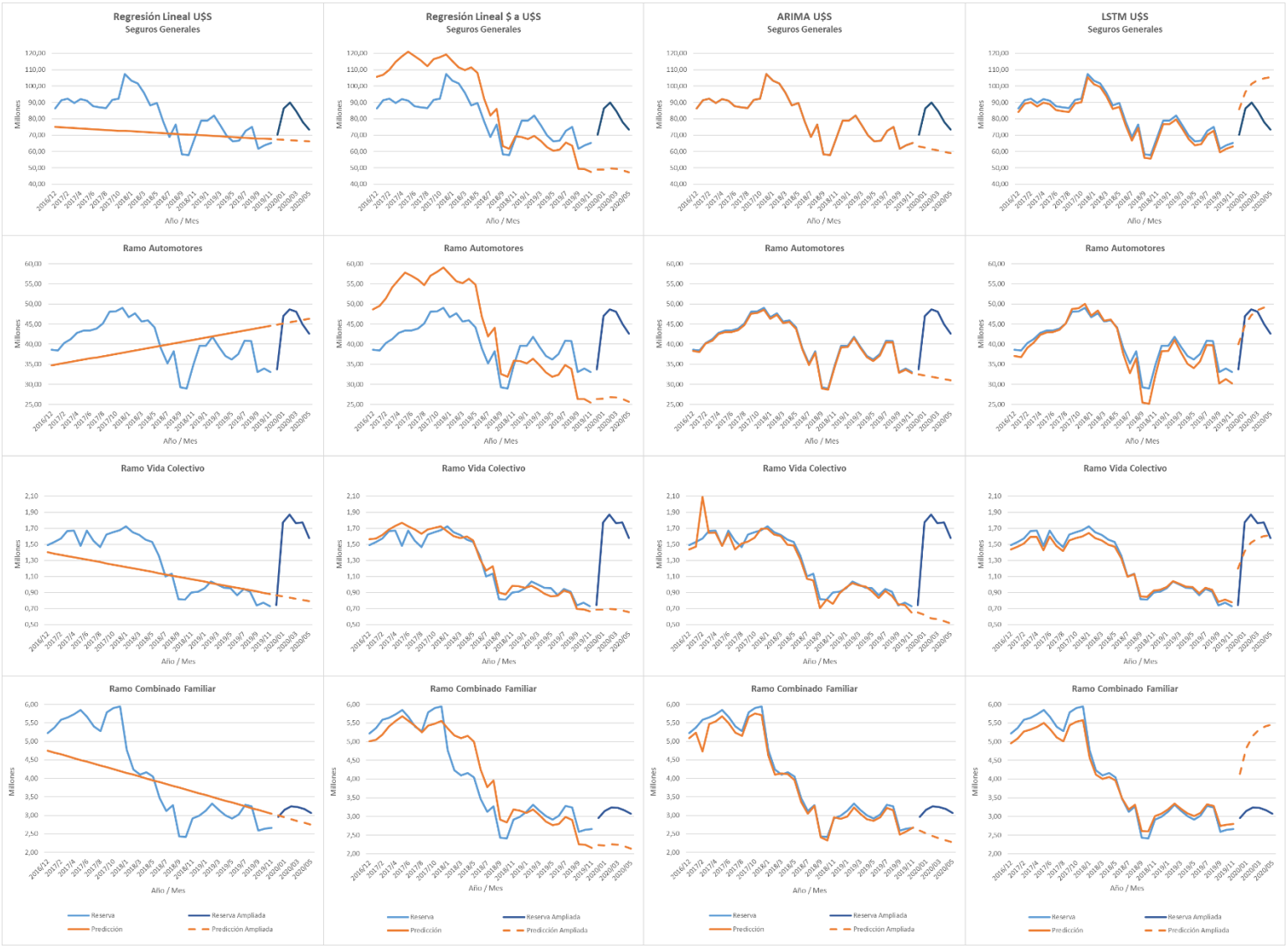


Gráfico 4.31: Desempeño de los modelos de Machine Learning entrenados con datos de Reserva Riesgo en Curso en dólares estadounidences entre Dic-2016 y Nov-2019, para predecir los siguientes seis meses, entre Dic-2019 y May-2020. Las predicciones se compararon con los datos reales que fueron acontenciendo en ese período, tanto para Reservas generales como por ramo de producto.

Es interesante destacar las diferentes estrategias que tienen los algoritmos a la hora de predecir la realidad, situación que se hace bien visible fundamentalmente en los modelos en dólares que gozan de datos más limpios en relación a situaciones inflacionarias y devaluatorias. Si bien la regresión lineal naturalmente sigue una línea recta y no tienen la capacidad de adaptase a un nuevo ciclo estacional en los datos, los algoritmos especializados en series temporales que sí tienen la posibilidad de hacerlo, tuvieron diferentes efectos. Mientras que la técnica ARIMA siguió la tendencia de los últimos períodos con ciclos pequeños y con tendencia descendente, las redes

neuronales LSTM predijeron ciclos más grandes retomando la tendencia alcista de los primeros períodos, situación que hizo que sea la técnica que mejor pudo predecir el salto en los datos.

Haciendo un análisis cuantitativo de los errores, a continuación en el ***Gráfico 4.32*** se muestra cómo los niveles de error de aumentaron significativamente cuando predijeron datos para los cuales no fueron entrenados, muchos de los cuales superaron el umbral de aceptación definido en fases anteriores.

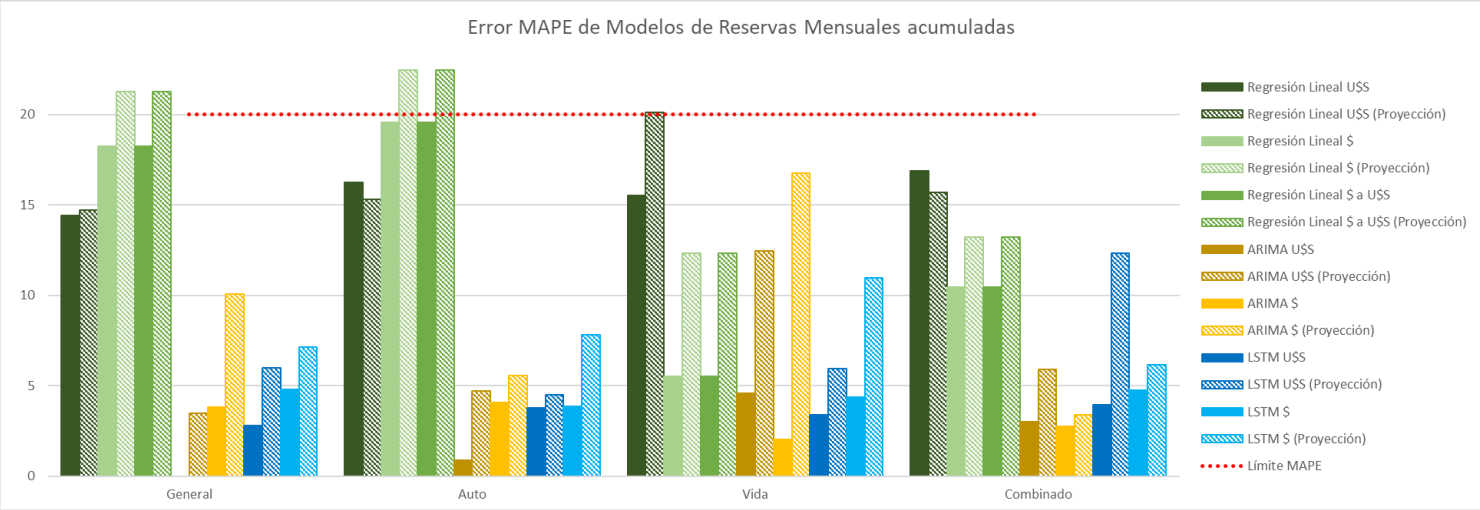


Gráfico 4.32: Error MAPE de los entrenados para predecir la reserva agrupada mensual en formal general y por ramo, diferenciando el error conseguido durante el entrenamiento y el de las predicciones (Proyección).

Una visión más concreta del error de cada técnica se muestra a continuación en el ***Gráfico 4.33***, donde se proyecta la magnitud en que aumentó el error al comparar los períodos incluidos en el entrenamiento y los proyectados. Llamativamente el modelo que mejor se adaptó a los nuevos datos fue la regresión lineal en dólares, situación que se justifica volviendo a analizar visualmente el ***Gráfico 4.31***, y verificando que el nuevo ciclo de los datos cruza justo la recta lineal proyectada en la mayoría de los ramos, resultando en niveles de error similares o inferiores al de la etapa de entrenamiento.

Luego de la regresión lineal en dólares, el desempeño más estable en su capacidad de predicción lo tienen los modelos LSTM, aunque no con diferencias tan significativas en comparación al resto como se podía intuir luego de la exploración visual de las gráficas de tendencias.

Por último, salvo el caso de la regresión lineal en dólares comentado anteriormente, si bien los modelos en dólares siguen gozando de un menor error relativo que sus correspondientes en pesos, no se manifestaron diferencias significativas en relación a la magnitud en que aumentó el error entre los períodos entrenado y no entrenados.

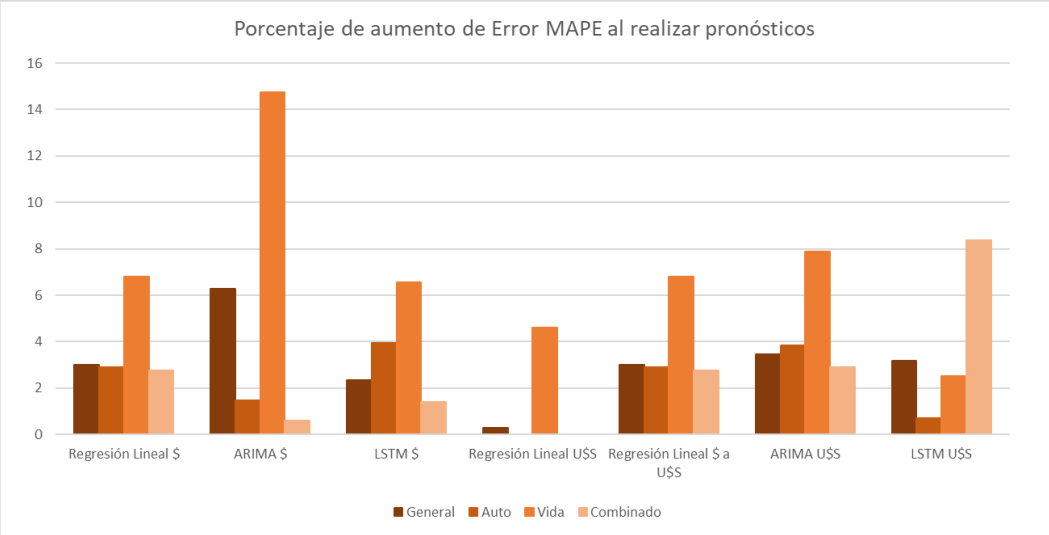


Gráfico 4.33: Aumento porcentual del Error MAPE de los diferentes modelos, haciendo la comparación entre los períodos cuyos datos se usaron para entrenamiento y los que fueron solamente predicciones.

Para completar el análisis, en el ***Gráfico 4.34*** se muestra el error desagregado en forma mensual. Se manifiesta claramente el fuerte aumento que sufre la curva del error cuando tuvo que estimar datos futuros, superando en muchos casos el umbral definido.

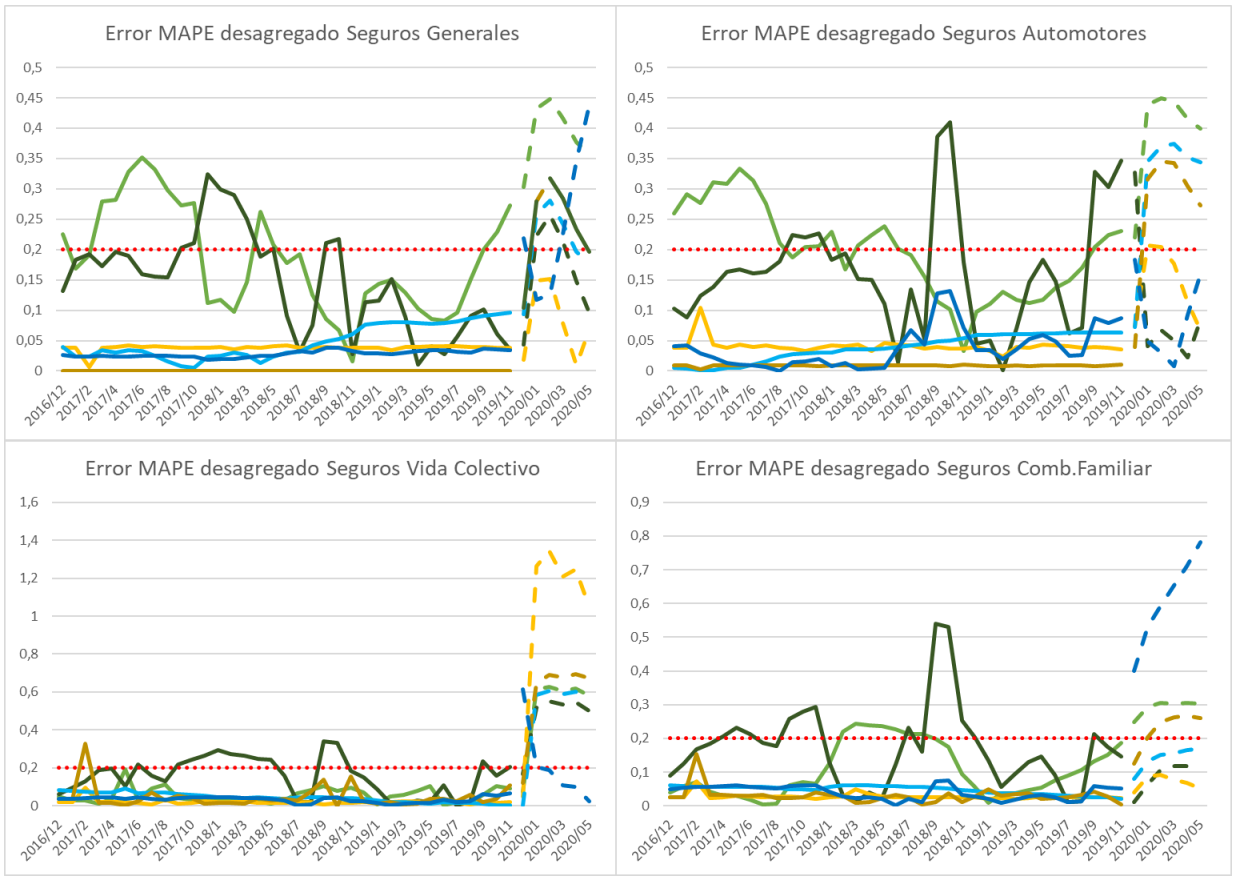


Gráfico 4.34: Error MAPE desagregado para cada mes de la serie de tiempo en cada modelo, incorporando la etapa de las predicciones.

Por lo mencionado anteriormente y como se expresa en el **Capítulo 3.2.1**, los modelos en producción requieren realizar éste tipo de análisis para detectar necesidades de reentrenamiento y evitar que las predicciones se alejen tanto de la realidad que lleven a la obsolescencia del modelo. Los reentrenamientos pueden desencadenarse cuando el error en las predicciones supera un determinado umbral de aceptación, o bien realizarse sistemáticamente todos los meses incorporando los nuevos datos generados al proceso.

Por último, si los procesos de reentrenamiento no logran alcanzar niveles de error aceptables, ocasionado por cambios abruptos en los datos, se debe volver a fases anteriores que impliquen un nuevo proceso de entendimiento y búsqueda de otros algoritmos que se adapten a los cambios ocurridos en los datos.

3.6.2 Reentrenamientos mensuales

Sometiendo los modelos ya productivos a un proceso de reentrenamiento mensual, en donde cada mes se suma al conjunto de datos los nuevos valores calculados con los procesos Batch de Reserva riesgo en curso, se espera que los niveles de error no se disparen sino que se consiga un nivel de estabilidad que permita sostener la confiabilidad de éste tipo de técnicas en el tiempo. A continuación en el ***Gráfico 4.35*** se muestra cómo el proceso de reentrenamiento de uno de los modelos va cambiando las curvas de predicción a medida que se va incorporando los nuevos datos de reserva obtenidos por los procesos batch tradicionales. Tomando el ejemplo de uno de los modelos de regresión lineal, los nuevos datos generarán cambios en la pendiente de las rectas tendiendo a ajustarse de la mejor forma a los nuevos valores reales que van apareciendo, y teniendo mayor precisión en las estimaciones futuras.

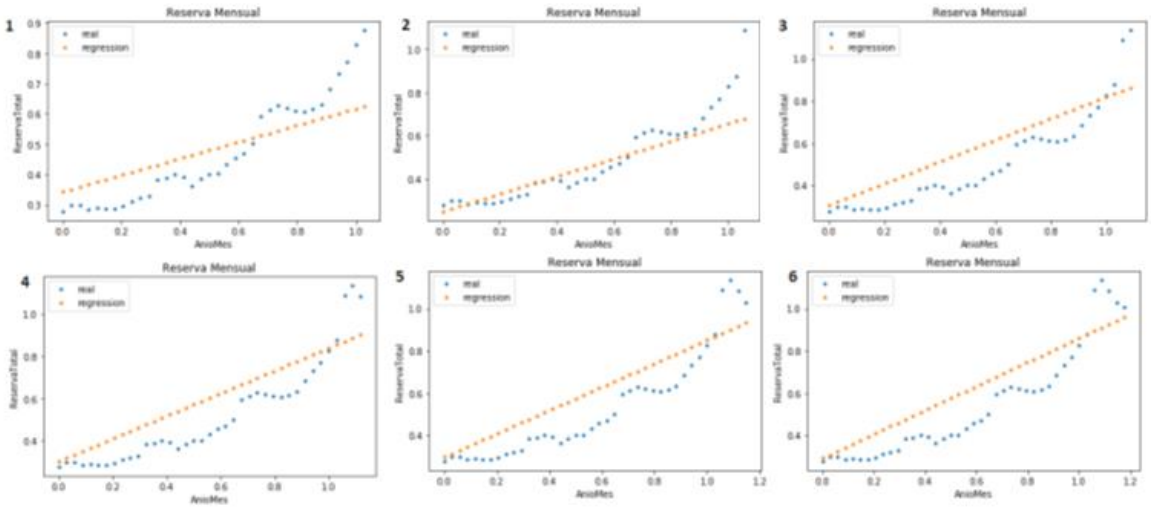


Gráfico 4.35: Cambios en la curva de regresión lineal del Modelo de Reserva en Pesos para Seguros Generales, a medida que se van incorporando nuevos datos reales y realizando un reentrenamiento.

A continuación en los ***Gráficos 4.36 y 4.37*** se agregan éstas nuevas curva de proyección a los ***Gráficos 4.30 y 4.31*** presentado en el apartado anterior, permitiendo comparar las predicciones originales de los modelos sin realizar reentrenamientos, con los modelos reentrenados mensualmente.

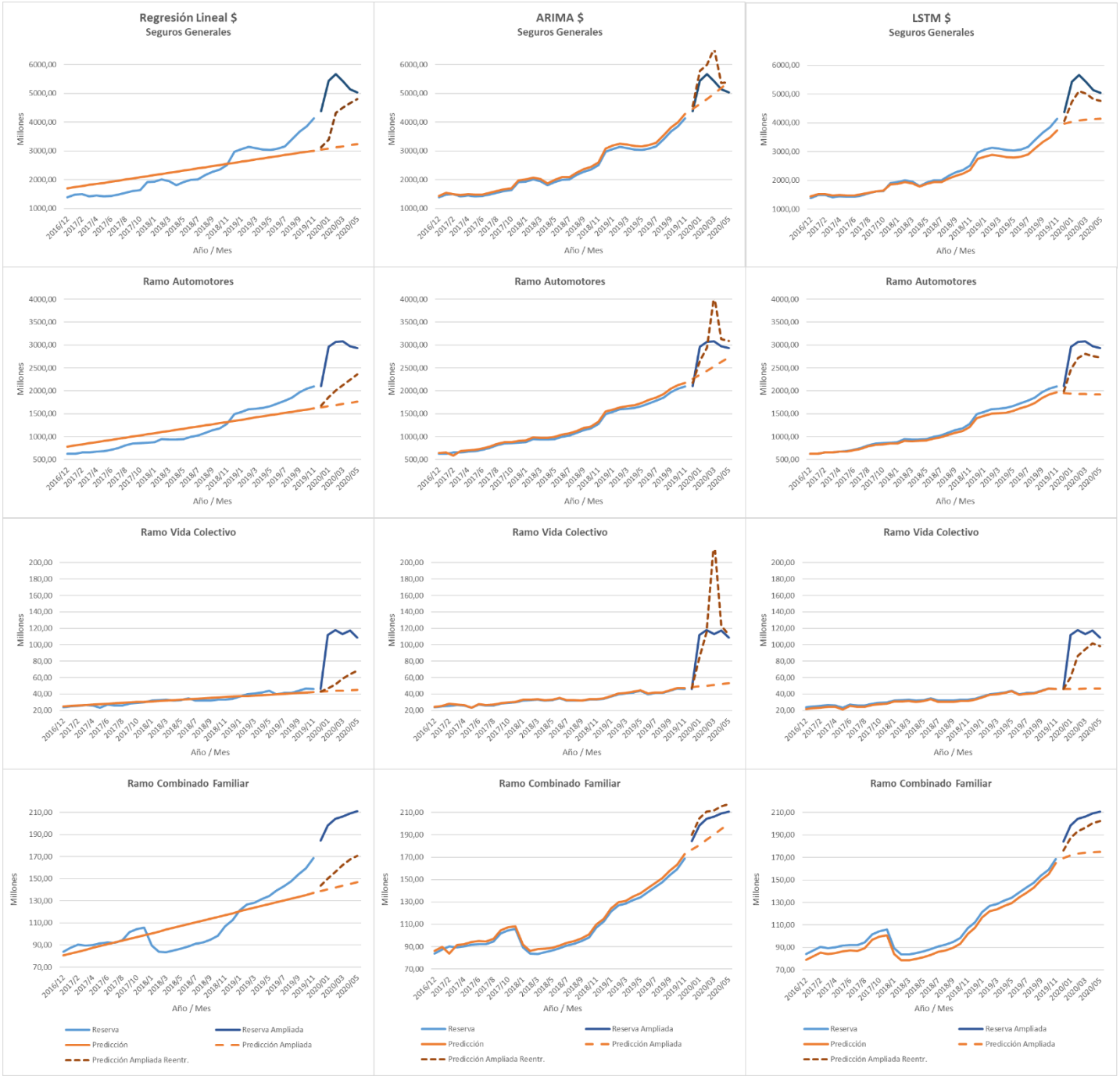


Gráfico 4.36: Desempeño de los modelos de Machine Learning entrenados con datos de Reserva Riesgo en Curso en pesos argentinos entre Dic-2016 y Nov-2019, para predecir los siguientes seis meses, entre Dic-2019 y May-2020, comparando las predicciones de los modelos originales y los reentrenados mensualmente.

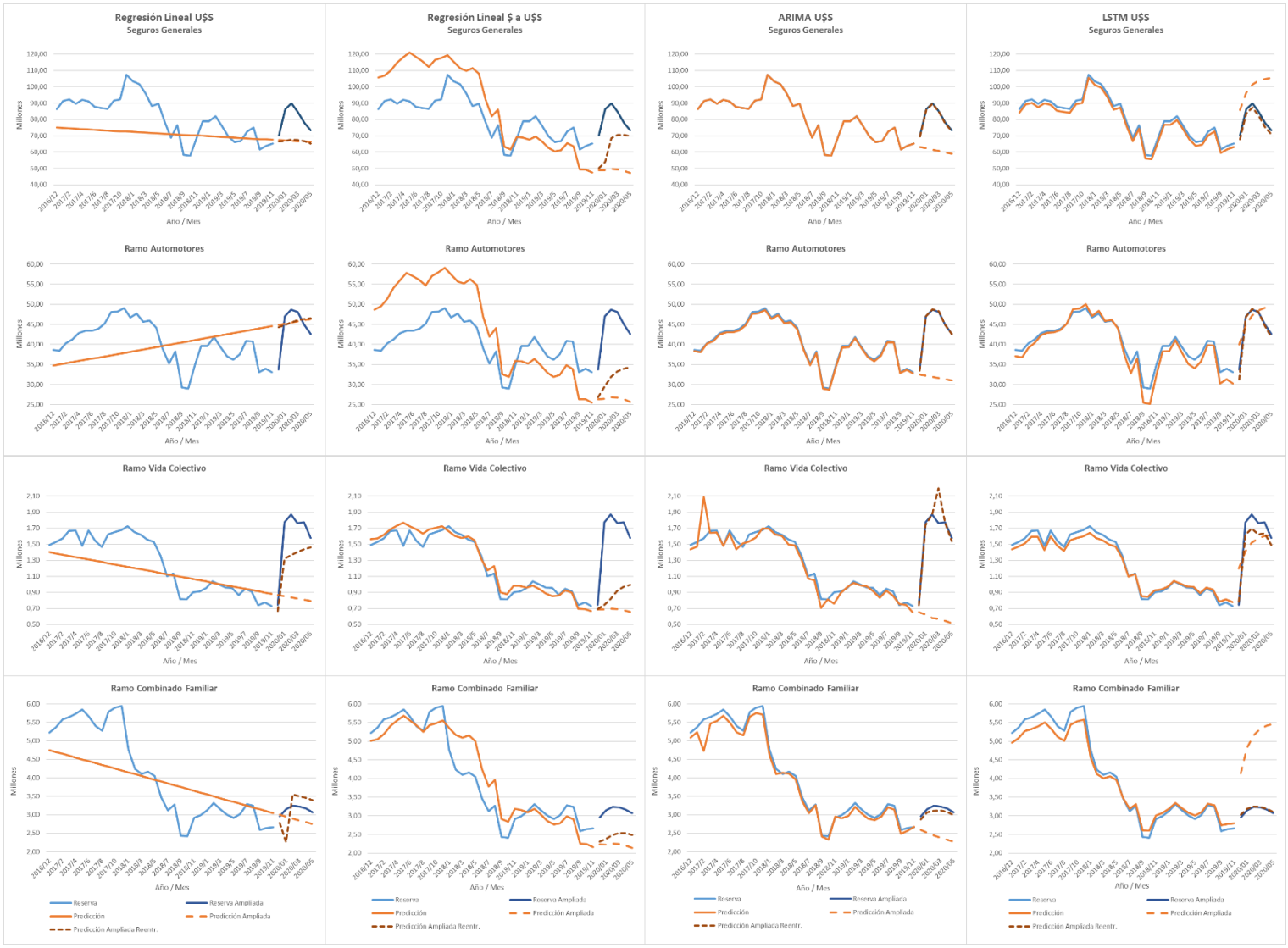


Gráfico 4.37: Desempeño de los modelos de Machine Learning entrenados con datos de Reserva Riesgo en Curso en dólares estadounidences entre Dic-2016 y Nov-2019, para predecir los siguientes seis meses, entre Dic-2019 y May-2020, comparando las predicciones de los modelos originales y los reentrenados mensualmente.

En todos los casos se ve claramente que la curva de las predicciones de los modelos reentrenados mensualmente se adaptan rápidamente a tendencia de los nuevos datos. En el caso de los modelos de regresión lineal, si bien en la gráfica se proyecta la recta original para los períodos de entrenamiento, a medida que se incorporan nuevos datos el modelo reentrenado va generando nuevas rectas con nuevas pendientes que intentan seguir su trayectoria (como se mostró en el ejemplo expuesto al inicio de éste apartado). En el caso de los modelos especializados en series temporales, el proceso de reentrenamiento puede detectar los momentos y magnitudes en que se inicia un nuevo ciclo estacional permitiendo ajustar las predicciones en ese sentido.

Por otro lado, en éstas gráficas se vuelve a manifestar la diferencia en el funcionamiento de los modelos ARIMA y LSTM analizados en la sección anterior, cuando se visualizó que los primeros ponderaban la tendencia de los últimos meses para la realización de sus predicciones futuras, mientras que los segundos tenían una visión más larga hacia atrás en el tiempo. Éste efecto se nota fundamentalmente en los casos en donde hubo cambios bruscos en la tendencia de los datos, como como ejemplo los seguros del ramo Vida Colectivo. La técnica ARIMA estimó para el mes siguiente al cambio brusco un nuevo aumento exponencial mientras que las redes LSTM tomaron en cuenta el incremento de una forma más suavizada.

Por último, en el ***Gráfico 4.38*** se muestra cómo el reentrenamiento mensual mejora significativamente el error general de las predicciones, lo que permite para la mayoría de los casos mantenerse por debajo del criterio de aceptación definido inicialmente.

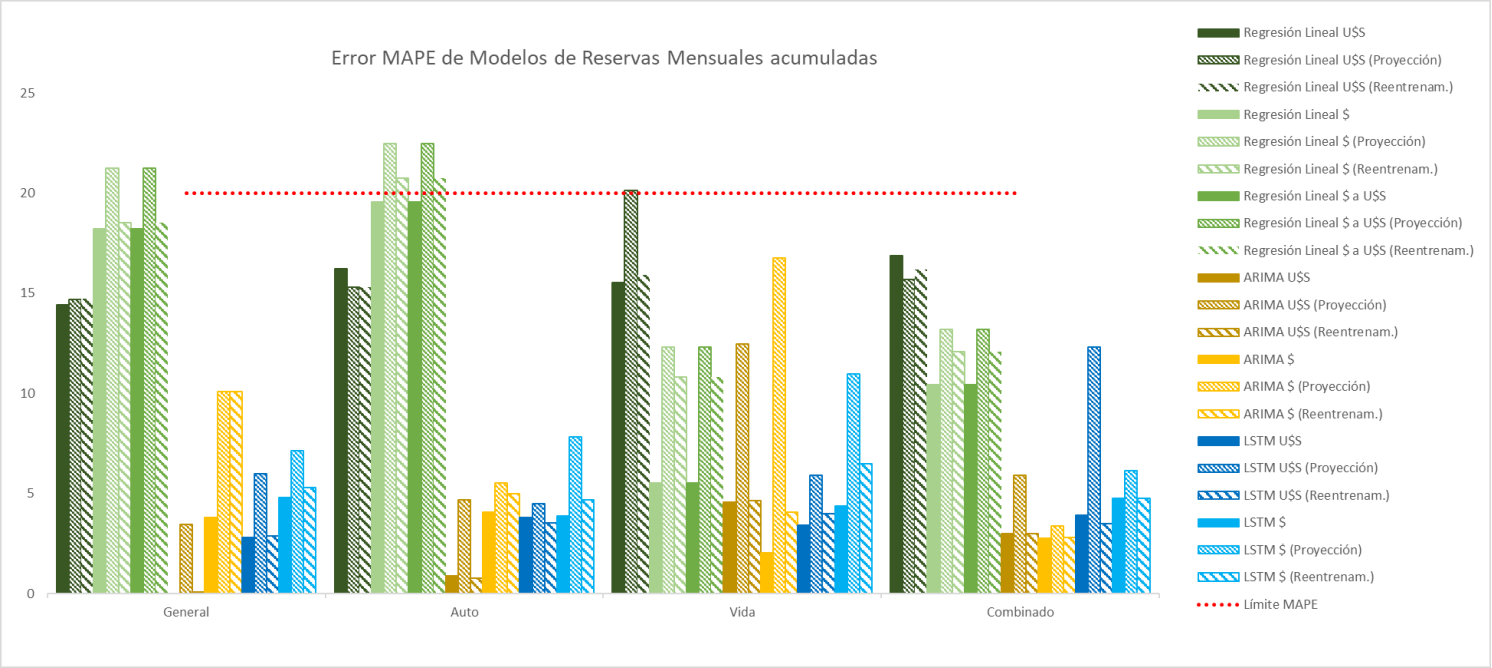


Gráfico 4.38: Error MAPE de los modelos entrenados para predecir la reserva agrupada mensual en formal general y por ramo, diferenciando el error conseguido durante el entrenamiento, el de las predicciones sin reentrenar (Proyección), y el de las predicciones con datos reentrenados mensualmente (Reentrenam.)

En el análisis del error desagregado en forma mensual del ***Gráfico 4.39*** se manifiesta la misma situación en donde el reentrenamiento permite mejorar la tendencia del error mes a mes en comparación con los modelos sin reentrenamiento mensual del ***Gráfico 4.34*** presentado en la sección anterior. Con el reentrenamiento mensual las curvas del error no se disparan ni se posicionan por encima del umbral como ocurrió en los modelos sin reentrenamiento, sino que manifiestan un pico en los meses en donde se produjeron cambios bruscos, pero rápidamente vuelven a posicionarse por debajo del umbral al incorporar ese nuevo dato al proceso de reentrenamiento.

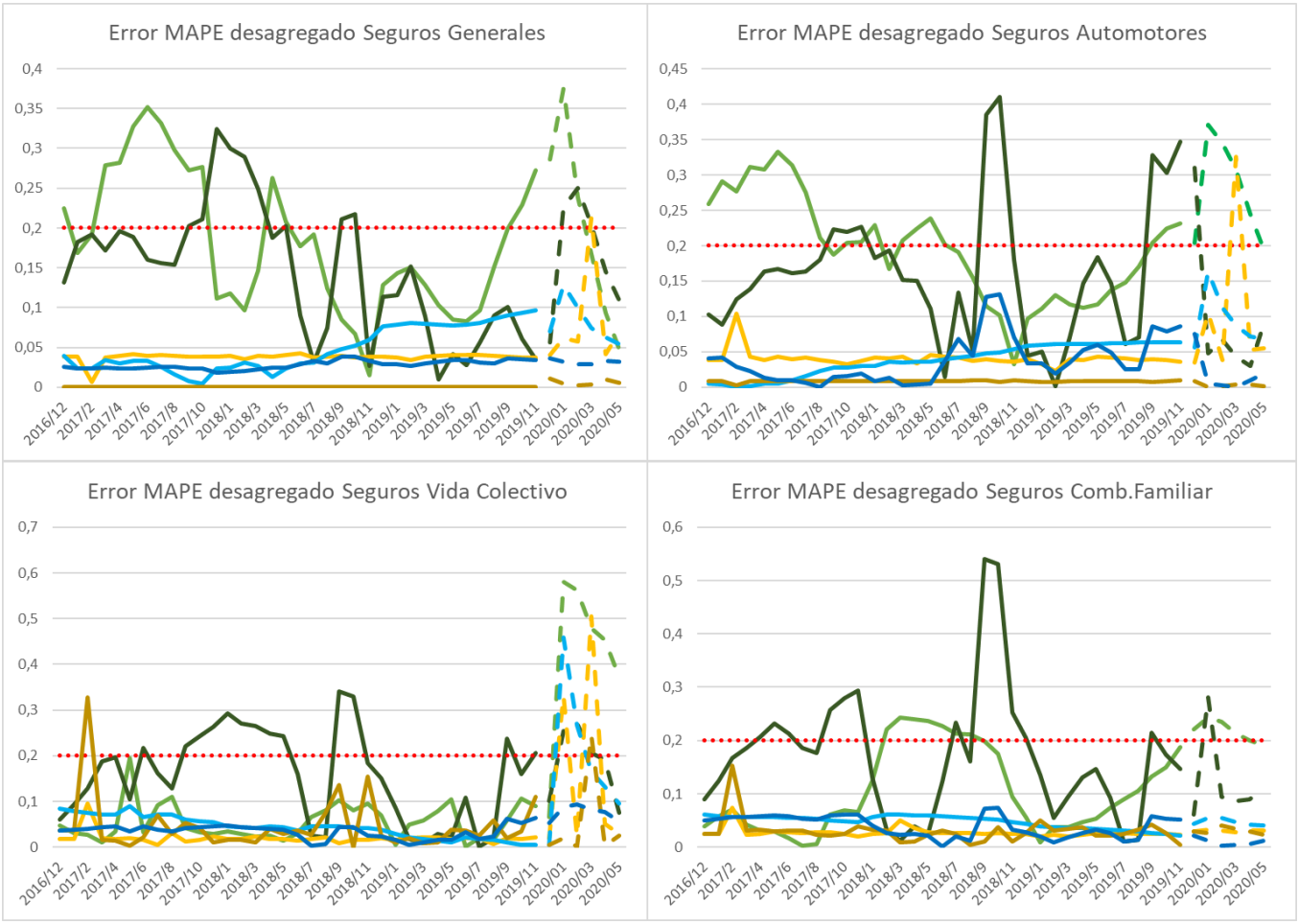


Gráfico 4.39: Error MAPE desagregado para cada mes de la serie de tiempo en cada modelo, incorporando la etapa de las predicciones con reentrenamiento mensual.

El reentrenamiento mensual en éste tipo de series temporales, permite un proceso más estable en el tiempo teniendo en cuenta el criterio de aceptación definido, sosteniendo la calidad de las predicciones.

De todas maneras, esto no significa que un reentrenamiento periódico va mantener la calidad de las predicciones indefinidamente en el tiempo, sino que es importante un monitoreo de esos niveles de error, que permita detectar cuándo se va a requerir realizar un replanteo, y volver a las etapas iniciales del proceso de Machine Learning para buscar nuevas alternativas y algoritmos que permitan restablecer la precisión y la confiabilidad en las precisiones.

4 CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

A lo largo del presente trabajo de investigación, se trabajó el objetivo de aplicar técnicas de Machine Learning en la industria del seguro, que tiene por un lado una necesidad urgente de innovación luego del arribo del concepto de "Insurtech", que propone un mercado mucho más competitivo que se basa justamente en la innovación y en la aplicación de tecnología al negocio. Pero por otro lado se trata de una actividad con un alto grado de regulación por parte de los organismos gubernamentales, que establecen un marco común de aplicación de procedimientos y técnicas de las cuales las aseguradoras no se pueden salir.

El tema de las Reservas es justamente uno de los conceptos que se desarrolla en esa puja entre innovación y regulación, porque si bien se trata de un tema definido por ley, tanto en lo conceptual como en lo técnico, representa uno de los indicadores más importantes a tener en cuenta en la toma de decisiones porque establece el límite de la libre disponibilidad de fondos que la compañía puede invertir para obtener resultados financieros.

No preocupa tanto si en un período un rubro o producto ofrecido por una aseguradora es deficitario si se tiene en cuenta sus resultados técnicos, porque la ganancia se puede generar a partir de los resultados financieros producto de invertir el dinero en el tiempo transcurrido entre que se cobra una prima y se paga un siniestro. En esa actividad cuanta más precisión se tenga en el control de los indicadores de siniestralidad, productividad y reservas, mayor será el rédito que se puede obtener de las disponibilidades monetarias en un momento específico.

En el presente trabajo describimos primero en forma general los diferentes tipos de reservas definidas por ley, y luego brindamos detalles de su complejo proceso de cálculo tanto en forma individual como agregada mensualmente como lo indica la ley, para luego enfocarnos en el objetivo general de aplicar técnicas de Machine Learning que puedan estimarlas de una forma simple y rápida.

4.1 CONCLUSIONES DEL PROCESO DE IMPLEMENTACIÓN DE MACHINE LEARNING

Se llevó a cabo un proceso de implementación de Machine Learning iterativo basado en el estándar CRISP-DM, en el cual se pudieron obtener las siguientes conclusiones:

- Fue significativo el tiempo y la complejidad de los procedimientos administrativos para conseguir las autorizaciones necesarias para poder acceder a los datos de la compañía de seguros sobre la cual se trabajó. Se trata de una empresa grande con presencia nacional e internacional, con fuertes estándar de seguridad y auditorías que regulan en acceso a la información de negocio.
- Fue significativo el tiempo incurrido en el entendimiento de los modelos de datos de los cuales obtener la información, dada la heterogeneidad de sistemas "legados" y nuevos, con documentación incompleta en algos casos, que fue completada a través de entrevistas con personal idóneo en IT.

- Los dos puntos anteriores sumado al volumen de los datos, hicieron que el 60% del tiempo total del proceso de Machine Learning se incurra en el acceso, entendimiento y preparación de los sets de datos con los cuales trabajar.

- Se realizó el entrenamiento, validación y testing de modelos lineales que predigan la reserva de una póliza individual de seguros automotor, utilizando la prima y capital asegurado como las características de entrada del modelo, pero no se consiguieron niveles de error satisfactorios que cumplan los criterios de evaluación definidos.

- Se realizó el entrenamiento de modelos lineales y técnicas especializadas en series temporales (ARIMA y Redes LSTM) que predigan la reserva acumulada en forma mensual, tanto a nivel general como desagregada en los ramos de Automotores, Vida colectivo y Combinado Familiar. Si bien la cantidad de datos de los sets de entrenamiento era bajo (un valor por mes durante 3 años), todos los modelos cumplieron el criterio de aceptación definido en un 20% de error MAPE.

- Si bien los modelos lineales se ajustaron efectivamente a la tendencia de los datos de la serie de tiempo, no pudieron manejar los ciclos estacionales por lo que los niveles de error fueron significativamente más altos en comparación con los algoritmos especializados en series temporales que sí manejan tanto tendencia como estacionalidad.

- Los modelos entrenados con datos expresados en dólares estadounidenses (convertidos utilizando el tipo de cambio oficial de cada mes), presentaron menores niveles de error en comparación al entrenamiento en pesos argentinos, ya que tienen menos influencia del contexto devaluatorio e inflacionario que caracterizó a nuestro país los últimos años.

- Si bien todos los modelos cumplieron los criterios de aceptación en sus predicciones para los meses en los que fueron entrenados, el error aumentó significativamente cuando realizaron predicciones a futuro. Ésta situación se vio agravada por un contexto especial, donde ocurrieron cambios normativos a nivel nacional para algunos ramos, y situaciones excepcionales como la pandemia mundial de COVID-19.

- Los niveles de error de los modelos mejoraron y volvieron a cumplir los criterios de aceptación al someterlos a un proceso de reentrenamiento mensual, sumándole mes a mes los nuevos valores de reservas calculados por los procesos tradicionales al set de datos de entrenamiento de los modelos.

- Los mejores indicadores de error en las predicciones de la serie temporal de reservas acumuladas lo obtuvo la técnica ARIMA entrenada sobre datos en dólares para los ramos en donde no hubo cambios bruscos en los datos de un mes a otro, mientras que las redes neuronales LSTM, entrenadas también con datos expresados en dólares, tuvieron mejor desempeño al momento de realizar predicciones en los ramos que sufrieron cambios bruscos de tendencia.

4.2 LÍNEAS FUTURAS DE INVESTIGACIÓN

La industria del seguro genera un sinnúmero de oportunidades para el desarrollo de técnicas de Machine Learning. En ésta investigación se trabajó en torno al concepto de reservas, pero existen otros campos de aplicación que se pueden explorar como siniestralidad, producción, fidelización de clientes, detección de morosidad y clientes deficitarios, reconocimiento de patrones en imágenes de siniestros agrícolas y patrimoniales, zonas de riesgos, etc..

En el presente trabajo se generaron modelos predictivos para la estimación de reservas tanto individual como agregadas mensualmente, del cual se desprenden las siguientes líneas concretas de investigación:

- Trabajar con otros algoritmos y/o características en los modelos predictivos de reservas en pólizas individuales desarrollados en el presente trabajo, que mejoren los indicadores de error y puedan cumplir con el criterio de aceptación definido.
- Analizar la aplicación de los algoritmos, modelos, y metodología de reentrenamiento mensual sobre reservas acumuladas del presente trabajo, para predecir sobre la misma línea temporal otros indicadores de importancia en el mundo del seguro como siniestralidad y producción.
- Ampliar la gama de algoritmos candidatos a ser utilizados en las predicciones para analizar su nivel de ajuste y márgenes de error, como puede ser por ejemplo regresiones logísticas, polinómicas, basadas en árboles de decisión, otros tipos de redes neuronales, etc..
- Profundizar en la parametrización de los modelo ARIMA, analizando la posibilidad de prevenir los puntos extremos en las predicciones en el mes siguiente a un cambio brusco en la tendencia de los datos.
- Analizar la posibilidad de reducir la complejidad de los modelos de redes LSMT sin afectar los niveles de error conseguidos.

5 ANEXO I

REGRESION INDIVIDUAL CON DATOS DE ESTADÍSTICA SINIESTRAL

In [1]:
```
from __future__ import print_function

from datetime import datetime
from time import time, sleep
import math
import datetime

def convert(n):
    return str(datetime.timedelta(seconds = n))

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
import tensorflow as tf
from tensorflow.python.data import Dataset

tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

claimHistory_dataframe = pd.read_csv("/home/alejandro/Downloads/Exprt02_200_99.csv", sep=",")

claimHistory_dataframe = claimHistory_dataframe.reindex(
    np.random.permutation(claimHistory_dataframe.index))
```
In [2]:
```
def preprocess_features(claimHistory_dataframe):

  selected_features = claimHistory_dataframe[
    ["nPremium"]]
  processed_features = selected_features.copy()

  return processed_features

def preprocess_traditional_prediction(claimHistory_dataframe):

  selected_traditional_prediction = claimHistory_dataframe[
    ["nReserve"]]
  processed_traditional_prediction = selected_traditional_prediction.copy()

  return selected_traditional_prediction

def preprocess_targets(claimHistory_dataframe):

  selected_targets = claimHistory_dataframe[
    ["nPayAmount"]]
  processed_targets = selected_targets.copy()

  return processed_targets
```
In [4]:
```
training_validation_data = claimHistory_dataframe.head(59691)
testing_data = claimHistory_dataframe.tail(10533)

training_examples = preprocess_features(training_validation_data.head(49158))
training_traditional_prediction = preprocess_traditional_prediction(training_validation_data.head(49158))
training_targets = preprocess_targets(training_validation_data.head(49158))

validation_examples = preprocess_features(training_validation_data.tail(10533))
validation_traditional_prediction =
preprocess_traditional_prediction(training_validation_data.tail(10533))
validation_targets = preprocess_targets(training_validation_data.tail(10533))

test_examples = preprocess_features(testing_data)
test_traditional_prediction = preprocess_traditional_prediction(testing_data)
test_targets = preprocess_targets(testing_data)
```
In [10]:
```
def my_input_fn(features, targets, batch_size=1, shuffle=True, num_epochs=None):
```

```
    """Trains a linear regression model of multiple features.

    Args:
      features: pandas DataFrame of features
      targets: pandas DataFrame of targets
      batch_size: Size of batches to be passed to the model
      shuffle: True or False. Whether to shuffle the data.
      num_epochs: Number of epochs for which data should be repeated. None = repeat indefinitely
    Returns:
      Tuple of (features, labels) for next data batch
    """

    # Convert pandas data into a dict of np arrays.
    features = {key:np.array(value) for key,value in dict(features).items()}

    # Construct a dataset, and configure batching/repeating.
    ds = Dataset.from_tensor_slices((features,targets)) # warning: 2GB limit
    ds = ds.batch(batch_size).repeat(num_epochs)

    # Shuffle the data, if specified.
    if shuffle:
      ds = ds.shuffle(10000)

    # Return the next batch of data.
    features, labels = ds.make_one_shot_iterator().get_next()
    return features, labels
```

In [13]:

```
def construct_feature_columns(input_features):
  """Construct the TensorFlow Feature Columns.

  Args:
    input_features: The names of the numerical input features to use.
  Returns:
    A set of feature columns
  """
  return set([tf.feature_column.numeric_column(my_feature)
              for my_feature in input_features])
```

In [14]:

```
def train_model(
    learning_rate,
    steps,
    batch_size,
    training_examples,
    training_targets,
    validation_examples,
    validation_targets):
  """Trains a linear regression model of multiple features.

  In addition to training, this function also prints training progress information,
  as well as a plot of the training and validation loss over time.

  Args:
    learning_rate: A `float`, the learning rate.
    steps: A non-zero `int`, the total number of training steps. A training step
      consists of a forward and backward pass using a single batch.
    batch_size: A non-zero `int`, the batch size.
    training_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for training.
    training_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for training.
    validation_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for validation.
    validation_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for validation.

  Returns:
    A `LinearRegressor` object trained on the training data.
  """

  periods = 10
  steps_per_period = steps / periods

  # Create a linear regressor object.
  my_optimizer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate)
  my_optimizer = tf.contrib.estimator.clip_gradients_by_norm(my_optimizer, 5.0)
  linear_regressor = tf.estimator.LinearRegressor(
      feature_columns=construct_feature_columns(training_examples),
      optimizer=my_optimizer
  )

  # Create input functions.
```

```
  training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["nPayAmount"],
      batch_size=batch_size)
  predict_training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["nPayAmount"],
      num_epochs=1,
      shuffle=False)
  predict_validation_input_fn = lambda: my_input_fn(
      validation_examples, validation_targets["nPayAmount"],
      num_epochs=1,
      shuffle=False)

  # Train the model, but do so inside a loop so that we can periodically assess
  # loss metrics.
  print("Training model... learning_rate=%0.2f,steps=%02d,batch_size=%02d" % (learning_rate, steps,
batch_size))
  print("Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)")
  training_rmse = []
  validation_rmse = []
  total_elapsed_time = 0
  for period in range (0, periods):
    # Start counting
    period_start_time = time()

    # Train the model, starting from the prior state.
    linear_regressor.train(
        input_fn=training_input_fn,
        steps=steps_per_period,
    )
    # Take a break and compute predictions.
    training_predictions = linear_regressor.predict(input_fn=predict_training_input_fn)
    training_predictions = np.array([item['predictions'][0] for item in training_predictions])

    validation_predictions = linear_regressor.predict(input_fn=predict_validation_input_fn)
    validation_predictions = np.array([item['predictions'][0] for item in validation_predictions])


    # Compute training and validation loss.
    training_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(training_predictions, training_targets))
    validation_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(validation_predictions, validation_targets))

    # Calculate de elapsed time
    period_elapsed_time = time() - period_start_time

    # Occasionally print the current loss.
    print(" %02d - %0.2f - %0.2f - %0.2f" % (period, period_elapsed_time / 60,
training_root_mean_squared_error, validation_root_mean_squared_error))

    # Add the loss metrics from this period to our list.
    training_rmse.append(training_root_mean_squared_error)
    validation_rmse.append(validation_root_mean_squared_error)
    total_elapsed_time = total_elapsed_time + period_elapsed_time

  print("Model training finished.")
  print("Total Minutes (ElapsedTime): %0.10f" % (total_elapsed_time / 60))

  # Output a graph of loss metrics over periods.
  plt.ylabel("RMSE")
  plt.xlabel("Periods")
  plt.title("Root Mean Squared Error vs. Periods")
  plt.tight_layout()
  plt.plot(training_rmse, label="training")
  plt.plot(validation_rmse, label="validation")
  plt.legend()

  return linear_regressor
```

In [20]:

```
linear_regressor = train_model(
    learning_rate=0.001,
    steps=100,
    batch_size=5,
    training_examples=training_examples,
    training_targets=training_targets,
    validation_examples=validation_examples,
    validation_targets=validation_targets)
Training model...
Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)
```

```
 00 - 0.56 - 527109.87 - 301559.91
 01 - 0.56 - 484163.08 - 293835.69
 02 - 0.53 - 454590.71 - 288450.89
 03 - 0.54 - 427803.86 - 283438.42
 04 - 0.53 - 397387.40 - 277368.96
 05 - 0.54 - 374366.02 - 272043.75
 06 - 0.53 - 362817.87 - 268565.13
 07 - 0.53 - 355952.25 - 264647.13
 08 - 0.54 - 359304.57 - 261585.63
 09 - 0.54 - 372599.15 - 259410.95
Model training finished.
Total Minutes (ElapsedTime): 5.3944045901
```

In [21]:

```
predict_test_input_fn = lambda: my_input_fn(
      test_examples,
      test_targets["nPayAmount"],
      num_epochs=1,
      shuffle=False)

test_predictions = linear_regressor.predict(input_fn=predict_test_input_fn)
test_predictions = np.array([item['predictions'][0] for item in test_predictions])

root_mean_squared_error = math.sqrt(
    metrics.mean_squared_error(test_predictions, test_targets))

print("Final RMSE (on test data): %0.2f" % root_mean_squared_error)
Final RMSE (on test data): 264601.94
```

In [22]:

```
root_mean_squared_error = math.sqrt(
    metrics.mean_squared_error(test_traditional_prediction, test_targets))

print("Final RMSE (on traditional prediction data): %0.2f" % root_mean_squared_error)
Final RMSE (on traditional prediction data): 317258.68
```

REGRESION INDIVIDUAL CON DATOS DE RESERVA RIESGO EN CURSO ## (una variable)

In [3]:

```
from __future__ import print_function

from datetime import datetime
from time import time, sleep
import math
import datetime

def convert(n):
    return str(datetime.timedelta(seconds = n))

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
import tensorflow as tf
from tensorflow.python.data import Dataset

tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

claimHistory_dataframe = pd.read_csv("/home/alejandro/Downloads/Exprt03_200_99.csv", sep=",")

claimHistory_dataframe = claimHistory_dataframe.reindex(
    np.random.permutation(claimHistory_dataframe.index))
```

In [4]:

```
def preprocess_features(claimHistory_dataframe):

  selected_features = claimHistory_dataframe[
    ["nPremium"]]
  processed_features = selected_features.copy()

  return processed_features

def preprocess_targets(claimHistory_dataframe):

  selected_targets = claimHistory_dataframe[
    ["nReserve"]]
```

```
  processed_targets = selected_targets.copy()

  return processed_targets
```
In [6]:
```
training_validation_data = claimHistory_dataframe.head(513004)
testing_data = claimHistory_dataframe.tail(90530)

training_examples = preprocess_features(training_validation_data.head(422474))
training_targets = preprocess_targets(training_validation_data.head(422474))

validation_examples = preprocess_features(training_validation_data.tail(90530))
validation_targets = preprocess_targets(training_validation_data.tail(90530))

test_examples = preprocess_features(testing_data)
test_targets = preprocess_targets(testing_data)
```
In [12]:
```
def my_input_fn(features, targets, batch_size=1, shuffle=True, num_epochs=None):
    """Trains a linear regression model of multiple features.

    Args:
      features: pandas DataFrame of features
      targets: pandas DataFrame of targets
      batch_size: Size of batches to be passed to the model
      shuffle: True or False. Whether to shuffle the data.
      num_epochs: Number of epochs for which data should be repeated. None = repeat indefinitely
    Returns:
      Tuple of (features, labels) for next data batch
    """

    # Convert pandas data into a dict of np arrays.
    features = {key:np.array(value) for key,value in dict(features).items()}

    # Construct a dataset, and configure batching/repeating.
    ds = Dataset.from_tensor_slices((features,targets)) # warning: 2GB limit
    ds = ds.batch(batch_size).repeat(num_epochs)

    # Shuffle the data, if specified.
    if shuffle:
      ds = ds.shuffle(10000)

    # Return the next batch of data.
    features, labels = ds.make_one_shot_iterator().get_next()
    return features, labels
```
In [13]:
```
def construct_feature_columns(input_features):
  """Construct the TensorFlow Feature Columns.

  Args:
    input_features: The names of the numerical input features to use.
  Returns:
    A set of feature columns
  """
  return set([tf.feature_column.numeric_column(my_feature)
              for my_feature in input_features])
```
In [15]:
```
def train_model(
    learning_rate,
    steps,
    batch_size,
    training_examples,
    training_targets,
    validation_examples,
    validation_targets):
  """Trains a linear regression model of multiple features.

  In addition to training, this function also prints training progress information,
  as well as a plot of the training and validation loss over time.

  Args:
    learning_rate: A `float`, the learning rate.
    steps: A non-zero `int`, the total number of training steps. A training step
      consists of a forward and backward pass using a single batch.
    batch_size: A non-zero `int`, the batch size.
    training_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for training.
    training_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for training.
    validation_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for validation.
    validation_targets: A `DataFrame` containing exactly one column from
```

```
      `california_housing_dataframe` to use as target for validation.

  Returns:
    A `LinearRegressor` object trained on the training data.
  """

  periods = 10
  steps_per_period = steps / periods

  # Create a linear regressor object.
  my_optimizer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate)
  my_optimizer = tf.contrib.estimator.clip_gradients_by_norm(my_optimizer, 5.0)
  linear_regressor = tf.estimator.LinearRegressor(
      feature_columns=construct_feature_columns(training_examples),
      optimizer=my_optimizer
  )

  # Create input functions.
  training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["nReserve"],
      batch_size=batch_size)
  predict_training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["nReserve"],
      num_epochs=1,
      shuffle=False)
  predict_validation_input_fn = lambda: my_input_fn(
      validation_examples, validation_targets["nReserve"],
      num_epochs=1,
      shuffle=False)

  # Train the model, but do so inside a loop so that we can periodically assess
  # loss metrics.
  print("Training model... learning_rate=%0.2f,steps=%02d,batch_size=%02d" % (learning_rate, steps,
batch_size))
  print("Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)")
  training_rmse = []
  validation_rmse = []
  total_elapsed_time = 0
  for period in range (0, periods):
    # Start counting
    period_start_time = time()

    # Train the model, starting from the prior state.
    linear_regressor.train(
        input_fn=training_input_fn,
        steps=steps_per_period,
    )
    # Take a break and compute predictions.
    training_predictions = linear_regressor.predict(input_fn=predict_training_input_fn)
    training_predictions = np.array([item['predictions'][0] for item in training_predictions])

    validation_predictions = linear_regressor.predict(input_fn=predict_validation_input_fn)
    validation_predictions = np.array([item['predictions'][0] for item in validation_predictions])


    # Compute training and validation loss.
    training_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(training_predictions, training_targets))
    validation_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(validation_predictions, validation_targets))

    # Calculate de elapsed time
    period_elapsed_time = time() - period_start_time

    # Occasionally print the current loss.
    print(" %02d - %0.2f - %0.2f - %0.2f" % (period, period_elapsed_time / 60,
training_root_mean_squared_error, validation_root_mean_squared_error))

    # Add the loss metrics from this period to our list.
    training_rmse.append(training_root_mean_squared_error)
    validation_rmse.append(validation_root_mean_squared_error)
    total_elapsed_time = total_elapsed_time + period_elapsed_time

  print("Model training finished.")
  print("Total Minutes (ElapsedTime): %0.10f" % (total_elapsed_time / 60))

  # Output a graph of loss metrics over periods.
  plt.ylabel("RMSE")
  plt.xlabel("Periods")
  plt.title("Root Mean Squared Error vs. Periods")
```

```
  plt.tight_layout()
  plt.plot(training_rmse, label="training")
  plt.plot(validation_rmse, label="validation")
  plt.legend()

  return linear_regressor
```

In [16]:

```
linear_regressor = train_model(
    learning_rate=0.005,
    steps=100,
    batch_size=100,
    training_examples=training_examples,
    training_targets=training_targets,
    validation_examples=validation_examples,
    validation_targets=validation_targets)
```

```
Training model... learning_rate=0.01,steps=100,batch_size=100
Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)
 00 - 4.30 - 36938.27 - 45088.55
 01 - 4.27 - 25579.09 - 27822.24
 02 - 4.26 - 19734.86 - 20677.87
 03 - 4.29 - 25580.25 - 27823.90
 04 - 4.44 - 19734.86 - 20677.87
 05 - 4.35 - 17877.22 - 20052.40
 06 - 4.31 - 19734.86 - 20677.87
 07 - 4.27 - 17877.21 - 20052.40
 08 - 4.27 - 17877.21 - 20052.40
 09 - 4.49 - 17877.21 - 20052.40
Model training finished.
Total Minutes (ElapsedTime): 43.2510897597
```

In [17]:

```
predict_test_input_fn = lambda: my_input_fn(
      test_examples,
      test_targets["nReserve"],
      num_epochs=1,
      shuffle=False)

test_predictions = linear_regressor.predict(input_fn=predict_test_input_fn)
test_predictions = np.array([item['predictions'][0] for item in test_predictions])

root_mean_squared_error = math.sqrt(
    metrics.mean_squared_error(test_predictions, test_targets))

print("Final RMSE (on test data): %0.2f" % root_mean_squared_error)
```

```
Final RMSE (on test data): 11370.82
```

REGRESION INDIVIDUAL CON DATOS DE RESERVA RIESGO EN CURSO ## (dos variables)

In [1]:

```
from __future__ import print_function

from datetime import datetime
from time import time, sleep
import math
import datetime

def convert(n):
    return str(datetime.timedelta(seconds = n))

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
import tensorflow as tf
from tensorflow.python.data import Dataset

tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

claimHistory_dataframe = pd.read_csv("/home/alejandro/Downloads/Exprt03_200_99.csv", sep=",")

claimHistory_dataframe = claimHistory_dataframe.reindex(
    np.random.permutation(claimHistory_dataframe.index))
```

In [2]:

```
def preprocess_features(claimHistory_dataframe):
```

```
  selected_features = claimHistory_dataframe[
    ["nPremium", "nCapital"]]
  processed_features = selected_features.copy()

  return processed_features

def preprocess_targets(claimHistory_dataframe):

  selected_targets = claimHistory_dataframe[
    ["nReserve"]]
  processed_targets = selected_targets.copy()

  return processed_targets
```

In [4]:

```
training_validation_data = claimHistory_dataframe.head(513004)
testing_data = claimHistory_dataframe.tail(90530)

training_examples = preprocess_features(training_validation_data.head(422474))
training_targets = preprocess_targets(training_validation_data.head(422474))

validation_examples = preprocess_features(training_validation_data.tail(90530))
validation_targets = preprocess_targets(training_validation_data.tail(90530))

test_examples = preprocess_features(testing_data)
test_targets = preprocess_targets(testing_data)
```

In [10]:

```
def my_input_fn(features, targets, batch_size=1, shuffle=True, num_epochs=None):
    """Trains a linear regression model of multiple features.

    Args:
      features: pandas DataFrame of features
      targets: pandas DataFrame of targets
      batch_size: Size of batches to be passed to the model
      shuffle: True or False. Whether to shuffle the data.
      num_epochs: Number of epochs for which data should be repeated. None = repeat indefinitely
    Returns:
      Tuple of (features, labels) for next data batch
    """

    # Convert pandas data into a dict of np arrays.
    features = {key:np.array(value) for key,value in dict(features).items()}

    # Construct a dataset, and configure batching/repeating.
    ds = Dataset.from_tensor_slices((features,targets)) # warning: 2GB limit
    ds = ds.batch(batch_size).repeat(num_epochs)

    # Shuffle the data, if specified.
    if shuffle:
      ds = ds.shuffle(10000)

    # Return the next batch of data.
    features, labels = ds.make_one_shot_iterator().get_next()
    return features, labels
```

In [11]:

```
def construct_feature_columns(input_features):
  """Construct the TensorFlow Feature Columns.

  Args:
    input_features: The names of the numerical input features to use.
  Returns:
    A set of feature columns
  """
  return set([tf.feature_column.numeric_column(my_feature)
              for my_feature in input_features])
```

In [12]:

```
def train_model(
    learning_rate,
    steps,
    batch_size,
    training_examples,
    training_targets,
    validation_examples,
    validation_targets):
  """Trains a linear regression model of multiple features.

  In addition to training, this function also prints training progress information,
  as well as a plot of the training and validation loss over time.

  Args:
```

```
    learning_rate: A `float`, the learning rate.
    steps: A non-zero `int`, the total number of training steps. A training step
      consists of a forward and backward pass using a single batch.
    batch_size: A non-zero `int`, the batch size.
    training_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for training.
    training_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for training.
    validation_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for validation.
    validation_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for validation.

  Returns:
    A `LinearRegressor` object trained on the training data.
  """

  periods = 10
  steps_per_period = steps / periods

  # Create a linear regressor object.
  my_optimizer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate)
  my_optimizer = tf.contrib.estimator.clip_gradients_by_norm(my_optimizer, 5.0)
  linear_regressor = tf.estimator.LinearRegressor(
      feature_columns=construct_feature_columns(training_examples),
      optimizer=my_optimizer
  )

  # Create input functions.
  training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["nReserve"],
      batch_size=batch_size)
  predict_training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["nReserve"],
      num_epochs=1,
      shuffle=False)
  predict_validation_input_fn = lambda: my_input_fn(
      validation_examples, validation_targets["nReserve"],
      num_epochs=1,
      shuffle=False)

  # Train the model, but do so inside a loop so that we can periodically assess
  # loss metrics.
  print("Training model... learning_rate=%0.2f,steps=%02d,batch_size=%02d" % (learning_rate, steps,
batch_size))
  print("Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)")
  training_rmse = []
  validation_rmse = []
  total_elapsed_time = 0
  for period in range (0, periods):
    # Start counting
    period_start_time = time()

    # Train the model, starting from the prior state.
    linear_regressor.train(
        input_fn=training_input_fn,
        steps=steps_per_period,
    )
    # Take a break and compute predictions.
    training_predictions = linear_regressor.predict(input_fn=predict_training_input_fn)
    training_predictions = np.array([item['predictions'][0] for item in training_predictions])

    validation_predictions = linear_regressor.predict(input_fn=predict_validation_input_fn)
    validation_predictions = np.array([item['predictions'][0] for item in validation_predictions])


    # Compute training and validation loss.
    training_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(training_predictions, training_targets))
    validation_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(validation_predictions, validation_targets))

    # Calculate de elapsed time
    period_elapsed_time = time() - period_start_time

    # Occasionally print the current loss.
    print(" %02d - %0.2f - %0.2f - %0.2f" % (period, period_elapsed_time / 60,
training_root_mean_squared_error, validation_root_mean_squared_error))

    # Add the loss metrics from this period to our list.
```

```
    training_rmse.append(training_root_mean_squared_error)
    validation_rmse.append(validation_root_mean_squared_error)
    total_elapsed_time = total_elapsed_time + period_elapsed_time

  print("Model training finished.")
  print("Total Minutes (ElapsedTime): %0.10f" % (total_elapsed_time / 60))

  # Output a graph of loss metrics over periods.
  plt.ylabel("RMSE")
  plt.xlabel("Periods")
  plt.title("Root Mean Squared Error vs. Periods")
  plt.tight_layout()
  plt.plot(training_rmse, label="training")
  plt.plot(validation_rmse, label="validation")
  plt.legend()

  return linear_regressor
```

In [27]:

```
linear_regressor = train_model(
    learning_rate=0.0001,
    steps=100,
    batch_size=50,
    training_examples=training_examples,
    training_targets=training_targets,
    validation_examples=validation_examples,
    validation_targets=validation_targets)
Training model...
Period - Minutes (ElapsedTime) - RMSE (Training) - RMSE (Validation)
 00 - 3.52 - 72694.75 - 75541.94
 01 - 3.52 - 72506.75 - 75347.93
 02 - 3.57 - 72137.40 - 74965.86
 03 - 3.50 - 48108.96 - 48678.81
 04 - 3.51 - 72056.96 - 74882.86
 05 - 3.74 - 71853.00 - 74671.86
 06 - 3.60 - 71789.84 - 74606.75
 07 - 3.52 - 108405.51 - 111954.92
 08 - 3.52 - 71551.62 - 74360.42
 09 - 3.62 - 47783.49 - 48263.33
Model training finished.
Total Minutes (ElapsedTime): 35.6133875012
```

In [28]:

```
predict_test_input_fn = lambda: my_input_fn(
      test_examples,
      test_targets["nReserve"],
      num_epochs=1,
      shuffle=False)

test_predictions = linear_regressor.predict(input_fn=predict_test_input_fn)
test_predictions = np.array([item['predictions'][0] for item in test_predictions])

root_mean_squared_error = math.sqrt(
    metrics.mean_squared_error(test_predictions, test_targets))

print("Final RMSE (on test data): %0.2f" % root_mean_squared_error)
Final RMSE (on test data): 42720.38
```

RESERVA RIESGO EN CURSO ACUMULADA (Regresión Gral $)

In [2]:

```
from __future__ import print_function

from datetime import datetime
from time import time, sleep
import math
import datetime

def convert(n):
    return str(datetime.timedelta(seconds = n))

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
import tensorflow as tf
from tensorflow.python.data import Dataset
```

```
tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

claimHistory_dataframe = pd.read_csv("/home/alejandro/Downloads/Exprt04__.csv", sep=",")

claimHistory_dataframe = claimHistory_dataframe.reindex(
    np.random.permutation(claimHistory_dataframe.index))
In [3]:
def preprocess_features(claimHistory_dataframe):

  selected_features = claimHistory_dataframe[
    ["AnioMes"]]
  processed_features = selected_features.copy()

  return processed_features

def preprocess_targets(claimHistory_dataframe):

  selected_targets = claimHistory_dataframe[
    ["ReservaTotal"]]
  processed_targets = selected_targets.copy()

  return processed_targets

In [8]:
def my_input_fn(features, targets, batch_size=1, shuffle=True, num_epochs=None):
    """Trains a linear regression model of multiple features.

    Args:
      features: pandas DataFrame of features
      targets: pandas DataFrame of targets
      batch_size: Size of batches to be passed to the model
      shuffle: True or False. Whether to shuffle the data.
      num_epochs: Number of epochs for which data should be repeated. None = repeat indefinitely
    Returns:
      Tuple of (features, labels) for next data batch
    """

    # Convert pandas data into a dict of np arrays.
    features = {key:np.array(value) for key,value in dict(features).items()}

    # Construct a dataset, and configure batching/repeating.
    ds = Dataset.from_tensor_slices((features,targets)) # warning: 2GB limit
    ds = ds.batch(batch_size).repeat(num_epochs)

    # Shuffle the data, if specified.
    if shuffle:
      ds = ds.shuffle(10000)

    # Return the next batch of data.
    features, labels = ds.make_one_shot_iterator().get_next()
    return features, labels
In [9]:
def construct_feature_columns(input_features):
  """Construct the TensorFlow Feature Columns.

  Args:
    input_features: The names of the numerical input features to use.
  Returns:
    A set of feature columns
  """
  return set([tf.feature_column.numeric_column(my_feature)
              for my_feature in input_features])
In [10]:
def train_model(
    learning_rate,
    steps,
    batch_size,
    training_examples,
    training_targets):
  """Trains a linear regression model of multiple features.

  In addition to training, this function also prints training progress information,
  as well as a plot of the training and validation loss over time.

  Args:
    learning_rate: A `float`, the learning rate.
    steps: A non-zero `int`, the total number of training steps. A training step
      consists of a forward and backward pass using a single batch.
    batch_size: A non-zero `int`, the batch size.
    training_examples: A `DataFrame` containing one or more columns from
```

```
      `california_housing_dataframe` to use as input features for training.
    training_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for training.
    validation_examples: A `DataFrame` containing one or more columns from
      `california_housing_dataframe` to use as input features for validation.
    validation_targets: A `DataFrame` containing exactly one column from
      `california_housing_dataframe` to use as target for validation.

  Returns:
    A `LinearRegressor` object trained on the training data.
  """

  periods = 10
  steps_per_period = steps / periods

  # Create a linear regressor object.
  my_optimizer = tf.train.GradientDescentOptimizer(learning_rate=learning_rate)
  my_optimizer = tf.contrib.estimator.clip_gradients_by_norm(my_optimizer, 5.0)
  linear_regressor = tf.estimator.LinearRegressor(
      feature_columns=construct_feature_columns(training_examples),
      optimizer=my_optimizer
  )

  # Create input functions.
  training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["ReservaTotal"],
      batch_size=batch_size)
  predict_training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["ReservaTotal"],
      num_epochs=1,
      shuffle=False)

  # Train the model, but do so inside a loop so that we can periodically assess
  # loss metrics.
  print("Training model... learning_rate=%0.2f,steps=%02d,batch_size=%02d" % (learning_rate, steps,
batch_size))
  print("Period - Minutes (ElapsedTime) - RMSE (Training)")
  training_rmse = []
  total_elapsed_time = 0
  for period in range (0, periods):
    # Start counting
    period_start_time = time()

    # Train the model, starting from the prior state.
    linear_regressor.train(
        input_fn=training_input_fn,
        steps=steps_per_period,
    )
    # Take a break and compute predictions.
    training_predictions = linear_regressor.predict(input_fn=predict_training_input_fn)
    training_predictions = np.array([item['predictions'][0] for item in training_predictions])

    # Compute training and validation loss.
    training_root_mean_squared_error = math.sqrt(
        metrics.mean_squared_error(training_predictions, training_targets))

    # Calculate de elapsed time
    period_elapsed_time = time() - period_start_time

    # Occasionally print the current loss.
    print(" %02d - %0.2f - %0.2f" % (period, period_elapsed_time / 60, training_root_mean_squared_error))

    # Add the loss metrics from this period to our list.
    training_rmse.append(training_root_mean_squared_error)
    total_elapsed_time = total_elapsed_time + period_elapsed_time

  print("Model training finished.")
  print("Total Minutes (ElapsedTime): %0.10f" % (total_elapsed_time / 60))

  # Output a graph of loss metrics over periods.
  plt.ylabel("RMSE")
  plt.xlabel("Periods")
  plt.title("Root Mean Squared Error vs. Periods")
  plt.tight_layout()
  plt.plot(training_rmse, label="training")
  plt.legend()

  return linear_regressor
In [11]:
linear_regressor = train_model(
```

```
    learning_rate=0.01,
    steps=10,
    batch_size=37,
    training_examples=training_examples,
    training_targets=training_targets)
Training model... learning_rate=0.01,steps=10,batch_size=37
Period - Minutes (ElapsedTime) - RMSE (Training)
 00 - 0.03 - 0.43
 01 - 0.02 - 0.38
 02 - 0.02 - 0.33
 03 - 0.02 - 0.27
 04 - 0.02 - 0.22
 05 - 0.02 - 0.17
 06 - 0.02 - 0.13
 07 - 0.02 - 0.10
 08 - 0.02 - 0.09
 09 - 0.02 - 0.09
Model training finished.
Total Minutes (ElapsedTime): 0.2149017890

In [12]:
predict_training_input_fn = lambda: my_input_fn(
      training_examples,
      training_targets["ReservaTotal"],
      num_epochs=1,
      shuffle=False)
training_predictions = linear_regressor.predict(input_fn=predict_training_input_fn)
training_predictions = np.array([item['predictions'][0] for item in training_predictions])
```

RESERVA RIESGO EN CURSO ACUMULADA (ARIMA Gral $)

```
In [1]:
from __future__ import print_function

from datetime import datetime
from time import time, sleep
import math
import datetime

def convert(n):
    return str(datetime.timedelta(seconds = n))

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn import metrics
import tensorflow as tf
from tensorflow.python.data import Dataset

tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

claimHistory_dataframe = pd.read_csv("/home/alejandro/Downloads/Exprt04_S.csv", sep=",")
pd.set_option('display.max_rows', 45)

In [5]:
model = sm.tsa.SARIMAX(claimHistory_dataframe['ReservaTotal'], order=(1, 0, 0), seasonal_order=(1,0,0,3),
enforce_stationarity=False, enforce_invertibility=False)
result = model.fit()
result.summary()
Out[5]:
SARIMAX Results Dep. Variable: ReservaTotal No. Observations:  35
Model: SARIMAX(1, 0, 0)x(1, 0, 0, 3) Log Likelihood  -619.262
Date: Tue, 29 Sep 2020 AIC  1244.524
Time: 06:47:58 BIC  1248.826
Sample: 0 HQIC  1245.926
 - 35
Covariance Type: opg
 coef std err z P>|z| [0.025 0.975]
ar.L1 1.0387 0.008 136.950 0.000 1.024 1.054
ar.S.L3 -0.0346 0.182 -0.190 0.849 -0.392 0.323
sigma2 1.263e+16 3.65e-18 3.46e+33 0.000 1.26e+16 1.26e+16
Ljung-Box (L1) (Q): 2.54 Jarque-Bera (JB):  8.09
Prob(Q): 0.11 Prob(JB):  0.02
Heteroskedasticity (H): 2.14 Skew:  0.91
```

```
Prob(H) (two-sided): 0.24 Kurtosis:  4.71


Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 1.53e+49. Standard errors may
be unstable.
In [6]:
prediction = result.predict(start=1, end=36)
prediction.reset_index(drop=True, inplace=True)
claimHistory_dataframe['predictions'] =  prediction
plot = claimHistory_dataframe[['AnioMes', 'ReservaTotal', 'predictions']].plot(x='AnioMes', figsize=(8,
5))
```

RESERVA RIESGO EN CURSO ACUMULADA (LSTM Gral $)

```
In [1]:
from __future__ import print_function

from datetime import datetime
from time import time, sleep
import math
import datetime

def convert(n):
    return str(datetime.timedelta(seconds = n))

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn import metrics
from sklearn.preprocessing import MinMaxScaler
import tensorflow as tf
from tensorflow.python.data import Dataset
from tensorflow import keras
from keras.models import Sequential
from keras.layers import *
from keras.optimizers import *

tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

In [5]:
df = pd.DataFrame()
df['Date'] = pd.to_datetime(claimHistory_dataframe['AnioMes'])
df['Reserva'] = claimHistory_dataframe['ReservaTotal']
indexed_df = df.set_index(["Date"], drop=True)
shifted_df = indexed_df.shift()
concat_df = [indexed_df, shifted_df]
data = pd.concat(concat_df, axis=1)
data.fillna(0, inplace=True)
data = np.array(data)
scaler = MinMaxScaler()
train_scaled = scaler.fit_transform(data)
y_train = train_scaled[:,-1]
X_train = train_scaled[:,0:-1]
X_train = X_train.reshape(len(X_train), 1, 1)
In [6]:
model = Sequential()

model.add(LSTM(1000,
    input_shape=(1,1)))
model.add(Dropout(0.00001))
model.add(Dense(1))
optimizer = Adam(lr=1e-3)


model.compile(loss='mean_squared_error', optimizer=optimizer, metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=100, batch_size=35, shuffle=False)

plt.plot(history.history['loss'])
plt.title('Root Mean Squared Error vs. Epochs')
plt.ylabel('RMSE')
plt.xlabel('Epoch')
plt.legend(['training'], loc='upper right')
plt.show
```

```
Epoch 1/100
35/35 [==============================] - 1s 32ms/step - loss: 0.3706 - acc: 0.0286
Epoch 2/100
35/35 [==============================] - 0s 3ms/step - loss: 0.3516 - acc: 0.0286
Epoch 3/100
35/35 [==============================] - 0s 3ms/step - loss: 0.3331 - acc: 0.0286
Epoch 4/100
35/35 [==============================] - 0s 3ms/step - loss: 0.3150 - acc: 0.0286
Epoch 5/100
35/35 [==============================] - 0s 3ms/step - loss: 0.2973 - acc: 0.0286
Epoch 6/100
35/35 [==============================] - 0s 3ms/step - loss: 0.2799 - acc: 0.0286
Epoch 7/100
35/35 [==============================] - 0s 3ms/step - loss: 0.2628 - acc: 0.0286
Epoch 8/100
35/35 [==============================] - 0s 3ms/step - loss: 0.2461 - acc: 0.0286
Epoch 9/100
35/35 [==============================] - 0s 3ms/step - loss: 0.2296 - acc: 0.0286
Epoch 10/100
35/35 [==============================] - 0s 3ms/step - loss: 0.2134 - acc: 0.0286
Epoch 11/100
35/35 [==============================] - 0s 3ms/step - loss: 0.1974 - acc: 0.0286
Epoch 12/100
35/35 [==============================] - 0s 4ms/step - loss: 0.1818 - acc: 0.0286
Epoch 13/100
35/35 [==============================] - 0s 3ms/step - loss: 0.1664 - acc: 0.0286
Epoch 14/100
35/35 [==============================] - 0s 3ms/step - loss: 0.1514 - acc: 0.0286
Epoch 15/100
35/35 [==============================] - 0s 3ms/step - loss: 0.1368 - acc: 0.0286
Epoch 16/100
35/35 [==============================] - 0s 2ms/step - loss: 0.1227 - acc: 0.0286
Epoch 17/100
35/35 [==============================] - 0s 3ms/step - loss: 0.1090 - acc: 0.0286
Epoch 18/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0959 - acc: 0.0286
Epoch 19/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0835 - acc: 0.0571
Epoch 20/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0718 - acc: 0.0571
Epoch 21/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0609 - acc: 0.0571
Epoch 22/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0509 - acc: 0.0571
Epoch 23/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0419 - acc: 0.0571
Epoch 24/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0338 - acc: 0.0571
Epoch 25/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0269 - acc: 0.0571
Epoch 26/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0210 - acc: 0.0571
Epoch 27/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0163 - acc: 0.0571
Epoch 28/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0127 - acc: 0.0571
Epoch 29/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0101 - acc: 0.0571
Epoch 30/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0086 - acc: 0.0571
Epoch 31/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0079 - acc: 0.0571
Epoch 32/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0080 - acc: 0.0571
Epoch 33/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0086 - acc: 0.0571
Epoch 34/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0096 - acc: 0.0571
Epoch 35/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0109 - acc: 0.0571
Epoch 36/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0121 - acc: 0.0571
Epoch 37/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0133 - acc: 0.0571
Epoch 38/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0142 - acc: 0.0571
Epoch 39/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0148 - acc: 0.0571
Epoch 40/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0150 - acc: 0.0571
Epoch 41/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0149 - acc: 0.0571
```

```
Epoch 42/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0145 - acc: 0.0571
Epoch 43/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0138 - acc: 0.0571
Epoch 44/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0129 - acc: 0.0571
Epoch 45/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0119 - acc: 0.0571
Epoch 46/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0109 - acc: 0.0571
Epoch 47/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0099 - acc: 0.0571
Epoch 48/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0090 - acc: 0.0571
Epoch 49/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0082 - acc: 0.0571
Epoch 50/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0075 - acc: 0.0571
Epoch 51/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0070 - acc: 0.0571
Epoch 52/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0066 - acc: 0.0571
Epoch 53/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0063 - acc: 0.0571
Epoch 54/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0062 - acc: 0.0571
Epoch 55/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0061 - acc: 0.0571
Epoch 56/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0061 - acc: 0.0571
Epoch 57/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0062 - acc: 0.0571
Epoch 58/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0062 - acc: 0.0571
Epoch 59/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0063 - acc: 0.0571
Epoch 60/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0064 - acc: 0.0571
Epoch 61/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0064 - acc: 0.0571
Epoch 62/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0064 - acc: 0.0571
Epoch 63/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0064 - acc: 0.0571
Epoch 64/100
35/35 [==============================] - 0s 4ms/step - loss: 0.0063 - acc: 0.0571
Epoch 65/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0063 - acc: 0.0571
Epoch 66/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0062 - acc: 0.0571
Epoch 67/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0060 - acc: 0.0571
Epoch 68/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0059 - acc: 0.0571
Epoch 69/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0058 - acc: 0.0571
Epoch 70/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0056 - acc: 0.0571
Epoch 71/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0055 - acc: 0.0571
Epoch 72/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0054 - acc: 0.0571
Epoch 73/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0053 - acc: 0.0571
Epoch 74/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0052 - acc: 0.0571
Epoch 75/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0051 - acc: 0.0571
Epoch 76/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0051 - acc: 0.0571
Epoch 77/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0050 - acc: 0.0571
Epoch 78/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0050 - acc: 0.0571
Epoch 79/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0050 - acc: 0.0571
Epoch 80/100
35/35 [==============================] - 0s 2ms/step - loss: 0.0050 - acc: 0.0571
Epoch 81/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0049 - acc: 0.0571
Epoch 82/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0049 - acc: 0.0571
```

```
Epoch 83/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0049 - acc: 0.0571
Epoch 84/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0049 - acc: 0.0571
Epoch 85/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0049 - acc: 0.0571
Epoch 86/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0048 - acc: 0.0571
Epoch 87/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0048 - acc: 0.0571
Epoch 88/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0048 - acc: 0.0571
Epoch 89/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0047 - acc: 0.0571
Epoch 90/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0047 - acc: 0.0571
Epoch 91/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0047 - acc: 0.0571
Epoch 92/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0046 - acc: 0.0571
Epoch 93/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0046 - acc: 0.0571
Epoch 94/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0046 - acc: 0.0571
Epoch 95/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0046 - acc: 0.0571
Epoch 96/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Epoch 97/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Epoch 98/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Epoch 99/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0045 - acc: 0.0571
Epoch 100/100
35/35 [==============================] - 0s 3ms/step - loss: 0.0045 - acc: 0.0571

In [7]:
predic_scaled = model.predict(X_train)

pr = pd.DataFrame()
pr['Date'] = pd.to_datetime(claimHistory_dataframe['AnioMes'])
pr['Prediccion'] = predic_scaled
indexed_pr = pr.set_index(["Date"], drop=True)
concat_pr = [indexed_pr, indexed_pr]
data_pr = pd.concat(concat_pr, axis=1)
data_pr = np.array(data_pr)
predic = scaler.inverse_transform(data_pr)
y_predic = predic[:,-1]

claimHistory_dataframe['prediction'] = y_predic
plot = claimHistory_dataframe[['AnioMes', 'ReservaTotal', 'prediction']].plot(x='AnioMes', figsize=(8,
5))

In [33]:
claimHistory_dataframe_predict = claimHistory_dataframe.append(pd.Series(data={'AnioMes':'2019-12'},
name=35))
claimHistory_dataframe_predict = claimHistory_dataframe_predict.append(pd.Series(data={'AnioMes':'2020-
01'}, name=36))
claimHistory_dataframe_predict = claimHistory_dataframe_predict.append(pd.Series(data={'AnioMes':'2020-
02'}, name=37))
claimHistory_dataframe_predict = claimHistory_dataframe_predict.append(pd.Series(data={'AnioMes':'2020-
03'}, name=38))
claimHistory_dataframe_predict = claimHistory_dataframe_predict.append(pd.Series(data={'AnioMes':'2020-
04'}, name=39))
claimHistory_dataframe_predict = claimHistory_dataframe_predict.append(pd.Series(data={'AnioMes':'2020-
05'}, name=40))
claimHistory_dataframe_predict['predictionsAm'] =  pd.Series()

last_X = claimHistory_dataframe['prediction'][34]
for i in range(35, 41):
    last_scaled = scaler.transform(np.array([[last_X, last_X]]))
    last_X_scaled = last_scaled[:,0:-1]
    last_X_scaled = last_X_scaled.reshape(len(last_X_scaled), 1, 1)
    next_scaled = model.predict(last_X_scaled)
    next_scaled = scaler.inverse_transform(np.array([[next_scaled[0][0], next_scaled[0][0]]]))
    next_Y = next_scaled[0][0]
    claimHistory_dataframe_predict['predictionsAm'][i] = next_Y
    last_X = next_Y

plot = claimHistory_dataframe_predict[['AnioMes', 'ReservaTotal', 'prediction',
'predictionsAm']].plot(x='AnioMes', figsize=(8, 5))
```

6 REFERENCIAS

Nieto de Alba U (1964) "*Bases técnicas y reservas de riesgos en curso*" Anales Instituto Actuarios Españoles.

Stiglitz R, Stiglitz G (1988): "*Contrato de Seguro*", Buenos Aires, Ediciones La Rocca

Meilij G (1990) "*Manual de Seguros*" 2° Ed. Act., Buenos Aires, Depalma

Mapfre (1990) "*Manual de Introducción al Seguro*", Madrid, Editorial Mapfre S.A.

Femández Dirube A (1993) "*Manual de Reaseguros*", Ed. General Re, 20 Edición Buenos Aires

Burges C (1998) "*A Tutorial on Support Vector Machines for Pattern Recognition*" Kluwer Academic Publishers

Mulhern, F. J. (1999). "*Customer profitability analysis: Measurement, concentration, and research directions*". Journal of Interactive Marketing, 13(1), 25–40.

Kramer T (2002) "*Assessment of the Commercial Applicability of Artificial Intelligence in Electronic Businesses*" Diplomarbeiten Agentur

López Saavedra D (2003) "*El pago tardío de la prima por el asegurado*" - Mercado Asegurador- N~ 287 Octubre/2003 pg. 24. 4

Di Giorgio A (2006) *"Aspectos tributarios de la actividad aseguradora"*. Buenos Aires. Universidad de Buenos Aires. Facultad de Ciencias Económicas. Escue la de Estudios de Posgrado.

Weinberger, K. Q., Blitzer, J. y L. K. Saul (2006). "*Distance metric learning for large margin nearest neighbor classification"*. Advances in neural information processing systems, 1473-1480.

Peng, Y., Kou, G., Sabatka, A., Matza, J., Chen, Z., Khazanchi, D., & Shi, Y. (2007) "*Application of classification methods to individual disability income insurance fraud detection"*. In Computational science–ICCS 2007 (pp. 852–858). Springer.

Kotsiantis, S. B. (2007). "*Supervised machine learning: A review of classification techniques*". Informatica 31, 3-24.

Xindong Wu et al (2008) "*Top 10 algorithms in data mining*" Knowledge and Information Systems

Turing A.M. (2009) "*Computing Machinery and Intelligence*" Springer, 23-65.

Fernandez Dirube A (2012) "*Principios Técnicos del Seguro*" http://www.elseguroenaccion.com.ar/?p=2198

Fernandez Dirube A (2012) "*Función y Base Económica del Seguro*" http://www.elseguroenaccion.com.ar/?p=2300

Fernandez Dirube A (2012) "*Los Riesgos, El Seguro y los Aseguradores*" http://www.elseguroenaccion.com.ar/?p=2135

Michalski, R. S., Carbonnel, J. G. y T. M. Mitchell (2013) "*Machine learning: an artificial intelligence approach*". Springer Science & Business Media.

Superintendencia de Seguros de la Nación (2014) "*Reglamento General de la Actividad Aseguradora*" Ministerio de Hacienda.

Burr S (2014) "*Active Learning Literature Survey*" Computer Sciences Technical Report 1648. University of Wisconsin–Madison

Traverso A (2014) "*¿Dónde está mi Prima? ¿Quién se quedó con mi Siniestro*" http://www.elseguroenaccion.com.ar/?p=7914

Moreno C (2015) "*Solvencia: La reserva para riesgos en curso*" http://asegurandome.com.ve/solvencia-la-reserva-para-riesgos-en-curso/

Kosea I, Mehmet Gokturkb,Kemal Kilicc (2015) *An interactive machine-learning-based electronic fraud and abusedetection system in healthcare insurance*" Computers & Industrial Engineering. Appl. Soft Comput. J. http://dx.doi.org/10.1016/j.asoc.2015.07.018

Aggarwal C (2015) "*Data Classification Algorithms and Applications*" CRC Press

Pereira J, Basto M, Ferreira da Silva A (2015) "*The logistic lasso and ridge regression in predicting corporate failure*" Procedia Economics and Finance 39 (2016) 634 – 641.

Marsland S (2015) "*Machine Learning An Algorithmic Perspective*" Second Edition, CRC Press

López Prior A (2016) "*Analítica Avanzada en el Sector Asegurador: Machine Learning*" https://cleverdata.io/sector-asegurador-machine-learning/

Witten, I. H., Frank, E., Hall, M. A. y C. J. Pal (2016). "*Data Mining: Practical machine learning tools and techniques"* Morgan Kaufmann.

Fratta V (2016) "*Señores: ¿Qué es el Seguro*" http://www.elseguroenaccion.com.ar/?p=14362

Russo C et al (2016) "*Tratamiento Masivo de Datos Utilizando Técnicas de Machine Learning*" Instituto de Investigación y Transferencia en Tecnología (ITT)

Vázquez Burguillo (2016) "*Ciencia Actuarial*" https://economipedia.com/definiciones/ciencia-actuarial.html

Kuangnan Fang a, Yefei Jiang b, Malin Song (2016) "*Customer profitability forecasting using Big Data analytics: A case studyof the insurance industry*" Computers & Industrial Engineering.

Lopez Briega (2016) "*Series de tiempo con Python*" https://relopezbriega.github.io/blog/2016/09/26/series-de-tiempo-con-python/

Brownlee, Jason (2017) "*Multivariate Time Series Forecasting with LSTMs in Keras*", https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/

Acevedo E, Serna A, y Serna E (2017) "*Principios y características de las redes neuronales artificiales*" Universidad Cooperativa de Colombia. Corporación Universitaria Remington. Medellín, Antioquia

Bengio, Goodfellow, Courville (2017) "*Deep Learning*" MIT Press

Roldán P (2017) "*Estadística*" https://economipedia.com/definiciones/estadistica.html

Weiwei Lin, Ziming Wu, Longxin Lin, Angzhan Wen, Jin Li (2017) "*An Ensemble Random Forest Algorithm for Insurance Big Data Analysis*" IEEE ACCESS

Zaforas Manuel (2017) *"De la ficción a la realidad: la maduración de la Inteligencia Artificial"* https://www.paradigmadigital.com/techbiz/la-ficcion-la-realidad-la-maduracion-la-inteligencia-artificial/

Guillen M, Pesantez Narvaez J (2018) "*Machine Learning y Modelización Predictiva para la Tarifación en el Seguro de Automóviles*" Anales del Instituto de Actuarios Españoles, 4ª época, 24, 2018/123-147

Eason (2018) "*Tasting Python Machine Learning : Insurance Claim Prediction*" https://medium.com/@easonlai888/tasting-python-machine-learning-insurance-claim-prediction-8775cd0aa926

Channabasava Gola (2018) *"Metrics: Types, Differences, Takeaway"* https://channabasavagola.github.io/2018-01-09-metrics/ https://channabasavagola.github.io/2018-01-11-metrics2/

Alarcón Madrid L (2018) "*Oportunidades Estratégicas de Insurtech en Seguros Personales*" Universidad Pontificia de Salamanca

Swalin Alvira (2018) "How to Make Your Machine Learning Models Robust to Outliers" https://heartbeat.fritz.ai/how-to-make-your-machine-learning-models-robust-to-outliers-44d404067d07

Tychobra (2018) "*Machine Learning for Insurance Claims*" https://www.r-bloggers.com/machine-learning-for-insurance-claims/

Roman Víctor (2019) "*Machine Learning: Cómo desarrollar un modelo desde Cero*" https://medium.com/datos-y-ciencia/machine-learning-c%C3%B3mo-desarrollar-un-modelo-desde-cero-cc17654f0d48

Kopczyk D (2019) "*Machine Learning in Insurance: Claim prediction*" http://dkopczyk.quantee.co.uk/claim-prediction/

Mayorga Muñoz L (2019) "*Data Mining vs Machine Learning: ¿Cuál es la diferencia?*" https://www.elperiodicodearagon.com/noticias/mas-voces/data-mining-vs-machine-learning-cual-es-diferencia_1364595.html

López J (2019) "*Teoría de la Probabilidad*". https://economipedia.com/definiciones/teoria-de-la-probabilidad.html

Lazcano R (2019) "*Big data, machine learning y deep learning: conceptos y diferencias*" https://blog.enzymeadvisinggroup.com/big-data-machine-learning

Pacheco Víctor (2019) *"Una Breve Historia del Machine Learning"* https://empresas.blogthinkbig.com/una-breve-historia-del-machine-learning/

Martínez Heras José (2019) *"Fases del Proceso de Machine Learning"* https://iartificial.net/fases-del-proceso-de-machine-learning/

Superintendencia de Seguros de la Nación (2020) "*Glosario de Seguros*" https://www.argentina.gob.ar/superintendencia-de-seguros/conciencia-aseguradora/glosario-ssn

Superintendencia de Riesgos del Trabajo (2020) "*Misión, Funciones y Objetivos*" https://www.argentina.gob.ar/srt/institucional/mision-funciones-y-objetivos