Clasificación según características de gestión de proyectos en ambientes académicos

Classifier according to characteristics of project management in academic environments

Rambo Alice R.[1,2], Boari Mariana[2], Sueldo Roberto L. [1], Rodriguez Miriam I. [2]

[1.] Depto. de Informática, Facultad de Ciencias Exactas Químicas y Naturales (FCEQyN), Universidad Nacional de Misiones (UNaM) Posadas, Misiones 3300/Argentina.
[2.] Depto. de Formación Docente e Investigación Científica. Facultad de Ciencias Exactas Químicas y Naturales (FCEQyN), Universidad Nacional de Misiones (UNaM) Posadas, Misiones 3300/Argentina.

alirambo@fceqyn.unam.edu.ar, marianneboar@gmail.com, roberto.sueldo@gmail.com, mirodriiguez8@gmail.com

Resumen En las carreras "Analista en Sistemas de Computación" y "Licenciatura en Sistemas de Información" de la Facultad de Ciencias Exactas Químicas y Naturales de la Universidad Nacional de Misiones, hay un espacio curricular perteneciente al tercer año de ambas carreras. Esta materia es obligatoria y cuatrimestral. La misma consiste en la gestión integral de proyectos unipersonales de desarrollo software. Se realizaron encuestas a los estudiantes agregando una variable final de bandera según estos llegaban a concluir exitosamente sus proyectos. A partir del análisis de los datos relevados, se logra determinar características presentes y proponer un clasificador que ayude a definir la posibilidad de éxito de los estudiantes según vayan presentando estas características mencionadas.

Abstract In the "Computer Systems Analyst" and "Bachelor's Degree in Information Systems" careers of the Faculty of Exact Chemical and Natural Sciences of the National University of Misiones, there is a curricular space belonging to the third year of both careers. This subject is compulsory and four-monthly. It deals with the management of individual software development projects. Students were surveyed adding a final flag variable as they successfully completed their projects. From the analysis of the data collected, it is possible to determine present characteristics and propose a classifier that helps define the possibility of success of the students as they present these characteristics mentioned.

Palabras claves: gestión de proyectos, desarrollo de software, aprendizaje automático, graduados informática
Keywords: project management, software development, machine learning, computer science graduates

1 Introducción

El presente trabajo se realiza en el contexto del proyecto de "Metodología para la definición y ponderación de factores de éxito para procesos de gestión de proyectos académicos unipersonales de práctica profesional supervisada en carreras de informática" Código 16Q646-PI. En las carreras "Analista en Sistemas de Computación" y "Licenciatura en Sistemas de Información" de la Facultad de Ciencias Exactas Químicas y Naturales de la Universidad Nacional de Misiones (FCEQyN-UNaM), hay un espacio curricular perteneciente al tercer año de ambas carreras, de cursado obligatorio cuatrimestral, en el segundo cuatrimestre, el cual consiste en la realización y ejecución de proyectos de desarrollo software. Partimos de un escenario donde se tiene un conjunto de datos de encuestas a estudiantes de la cátedra Trabajo Final y Proyecto Software de las carreras de Informática de la FCEQyN-UNaM. Estos datos se han cruzado con la información que determina si han finalizado la carrera pasando a ser graduados en las mismas. El proyecto de investigación dio inicio como tal desde el año 2018, contando con encuestas realizadas desde el año 2015 y con análisis iniciales sobre el conjunto de datos de tipo estadístico principalmente [1],[2],[3] entre otros.

2 Desarrollo

Sobre la Metodología se selecciona el estándar CRISP-DM [4] el cual cuenta con distintas fases del proceso de Machine Learning (ML) o Aprendizaje automático y con interaccionan entre ellas los cuales para el presente trabajo se van a resumir a cumplimentar la finalidad académica que consiste en proponer un modelo de aprendizaje automático, y generar el informe final. Dejando para el ámbito del proyecto de investigación, donde se recolectaron los datos, generar a posterior del presente trabajo una extensión y análisis más profundo y llevar el modelo a una solución integrada a una plataforma de consulta como se menciona más adelante en las líneas futuras.
En cuanto a herramientas para el presente estudio se utilizó Colaboratory, también llamado "Colab", permite ejecutar y programar en Python en el navegador con las siguientes ventajas: No requiere configuración. Da acceso gratuito a GPUs. Permite compartir contenido fácilmente. Los notebooks de Colab permiten combinar código ejecutable y texto enriquecido en un único documento. Los notebooks de Colab son notebooks de Jupyter que aloja Colab. El lenguaje utilizado es Python, es un lenguaje de programación promovido por *Python Software Foundation* utilizado ampliamente para soluciones de aprendizaje automático por la variedad de librerías implementadas en general es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.
Se procede en primer lugar a un análisis de los datos donde según relevamiento se agrega el dato de la condición de cada estudiante a marzo 2021 para determinar si han finalizado la carrera. Por otra parte, aunque la carrera a la que informan cada estudiante que pertenece en primera instancia se cataloga como ASC (Analista en Sistemas de Computación), LSI (Licenciatura en Sistemas de Información) y como tercera opción Ambas donde identifica estar inscripto a ambas carreras, se agrega la columna carrera2

donde se cuantifica el dato para convertirlo en una opción numérica más sencilla de procesar por algunos algoritmos. Teniendo un total de 49 en la opción Ambas, 30 en ASC y 6 en LSI. En este punto se plantea la posibilidad de tener como criterio que al consignar ambas carreras la rigurosidad y exigencia a la que se encuentran es el de la carrera de mayor grado, en este sentido, se agrega una columna nivelando hacia arriba donde la opción Ambas se unifica con solamente LSI, agregando la columna carrera3 donde se observan solo dos categorías en las carreras, quedando 55 en la opción 1 donde se agrupan LSI y Ambas y 30 en la opción donde se encuentran los que cursan solo ASC. Como se puede observar en la Tabla 1. Así también se identifica que se cuenta con 49 no recibidos y 36 sí recibidos entre los encuestados indicando que existe entre los encuestados un 42,35% de recibidos, como se observa en la Tabla 2.

Table 1. Detalle de las carreras existentes en las encuestas

Carrera definida	Cantidad
Ambas	49
ASC	30
LSI	6

Table 2. Cantidad de recibidos

Recibidos	Cantidad
SI	49
NO	36

Al visualizar una sola dimensión vemos con respecto al año de ingreso a la carrera consignado en la columna ingreso, que se puede observar en la Figura 1, de los encuestados la mayor distribución se encuentra en los años 2009 y 2015.

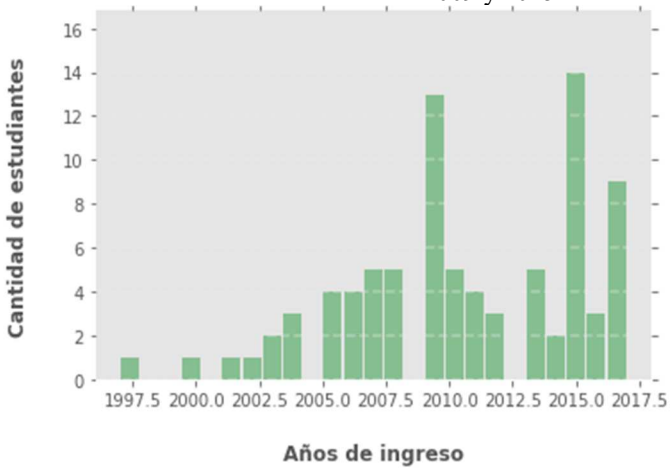

Fig. 1. Ingresos

2.1 Análisis de las relaciones entre los atributos

Una de las mejores formas de comprobar las posibles relaciones o correlaciones entre los diferentes atributos de los datos es aprovechar una matriz de correlación por pares y representarla como un mapa de calor. Los mapas de calor muestran datos tabulares numéricos donde los datos se pueden observar plasmados en celdas las cuales

están coloreadas según el valor contenido, graduando el color en representación al valor que contienen. Los mapas de calor son excelentes para hacer que las tendencias en este tipo de datos sean más evidentes, particularmente cuando los datos están ordenados y hay agrupaciones[1].

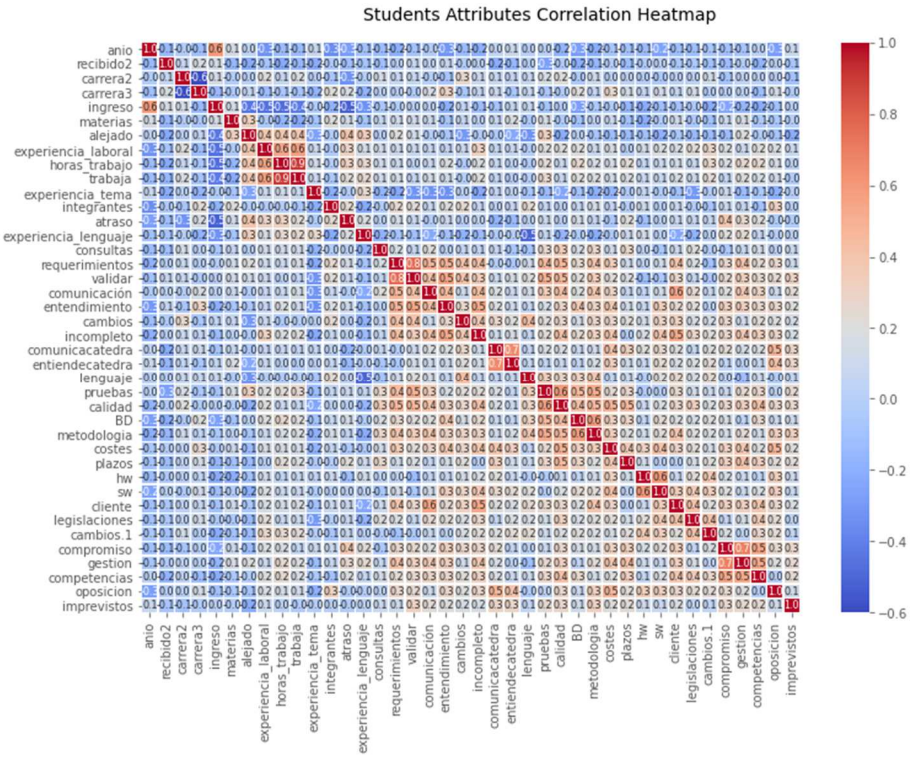

Fig. 2. Mapa de calor general

En la práctica, los mapas de calor a menudo se usan para mostrar la matriz de coeficientes de correlación de un grupo de variables, y también tienen un mayor uso en la visualización de la distribución de datos de las tablas de contingencia. Para lo cual realizamos una gráfica cruzando todas las variables obtenemos la gráfica de la Figura 2.

Podemos ver en la figura 2 focos de calor, observando valores de 0.7 o superiores se puede ver que el valor más alto es de 0.9 en trabaja y horas de trabajo, siguiendo con 0.8 entre requerimientos y validar, siguiendo 0.7 en gestión y compromiso, lo mismo para *entiendecatedra* y *comunicacátedra*. Con valores por debajo de 0.7 se puede observar trabaja /horas trabaja con experiencia laboral con 0.6, así también pruebas con

1. Waskom,M. 2021. seaborn: statistical data visualization. seaborn.heatmap. Recuperado de: https://seaborn.pydata.org/generated/seaborn.heatmap.html

calidad, base de datos (BD) con metodología, cliente con comunicación y Hardware (HW) con software (SW).

Le siguen en año (de la consulta, es decir, de cursado de la cátedra trabajo final) e ingreso (año de ingreso a la carrera) tiene una relación de 0.5, entendimiento con validar, metodología con calidad y pruebas, competencias con gestión y compromiso, cliente con incompleto, requerimientos con comunicación, entendimiento y calidad.

Si dejamos los valores menores a 0.5 para otro análisis y focalizándose para el presente análisis en las relaciones con valores iguales o mayores a 0.5 terminamos definiendo un total de 15 relaciones entre variables a ser analizadas. En la Tabla 3 se pueden observar las variables con valores en el mapa de calor iguales o mayores a 0.5 No se ven relaciones de valores mayores a 0.5 con respecto al atributo de recibidos.

Table 3. Variables con valores en el mapa de calor igual o mayores a 0.5.

variable	*BD (base de datos), calidad, competencias, compromiso, comunica cátedra, comunicación, cliente, entendimiento, entiende cátedra, experiencia laboral, gestión, horas_trabajo, HW (hardware), metodología, incompleto, ingreso, pruebas, requerimientos, SW (software, trabaja), validar*

Además, hay otras variables que inicialmente se pensaron que podrían tener relación, pero en el mapa de calor no se ven influencias considerables estas variables se ven en la Tabla 4.

Table 4. Variables que inicialmente se creían de influencia y en el mapa de calor tienen valores menores a 0.5

Variables con valores menores a 0.5	*Alejado, cambios, carrera, costes, experiencia lenguaje, experiencia tema, legislaciones, materias*

Los gradientes en el mapa de calor varían según la fuerza de la correlación y puede ver claramente que es muy fácil detectar atributos potenciales que tienen fuertes correlaciones entre ellos. Otra forma de visualizar lo mismo es usar diagramas de dispersión por pares entre los atributos de interés. Al observar los focos de calor es que pasamos a analizar la relación entre las variables de interés.

Al visualizar las variables de requerimientos y validar y cruzándolos con recibido, se obtiene la figura 3. donde se observan los gráficos de densidad y de puntos de las variables seleccionadas relacionadas con la variable denominada recibido.

3 Propuesta de un modelo de clasificación

Sobre el modelo propuesto mencionamos el modelo de regresión logística es un método estadístico para predecir clases binarias como la que tenemos en este planteo que es recibidos o no recibidos. El resultado o la variable objetivo es de naturaleza dicotómica.

Dicotómico significa que solo hay dos clases posibles. Es un caso especial de regresión lineal donde la variable objetivo es de naturaleza categórica. Utiliza un registro de probabilidades como variable dependiente. La regresión logística predice la probabilidad de ocurrencia de un evento binario utilizando una función *logit*.

Aunque habitualmente se trata sobre procesos dicotómicos, también puede ser usada para estimar procesos politómicos, con sucesos con más de dos categorías.

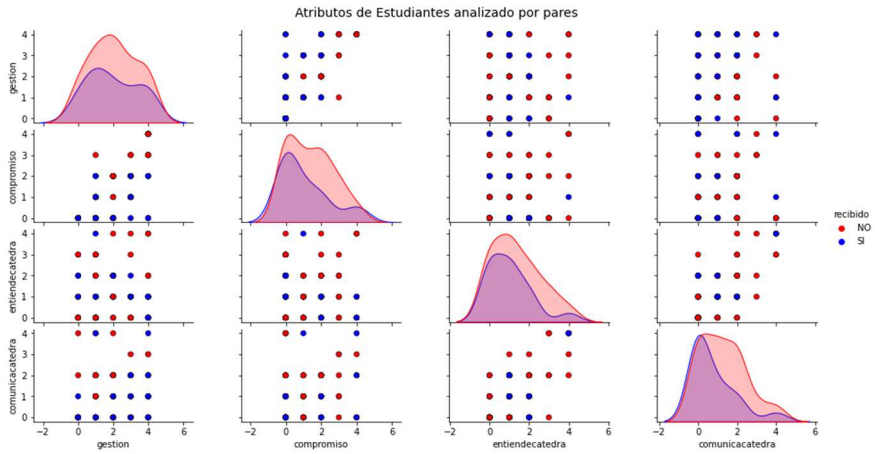

Fig. 3. Diagrama de dispersión variable de gestión, compromiso, entiendecatedra y comunicacatedra

El algoritmo toma como entrada una serie de datos X y genera a la salida una variable Y= h θ (x) con uno de dos posibles valores: 0 o 1. A estos valores de salida se les denomina clases o categorías. El objetivo de la Regresión Logística es clasificar de forma automática estos datos en las dos categorías existentes. Esto equivale a encontrar una frontera que permita separar los datos en dos agrupaciones diferentes.

Ecuación de regresión lineal:

$$h_\theta(x)=\theta_0+\theta_1 x_1+\theta_2 x_2 \quad (1)$$

Como elección inicial, se decide aproximar *y* como una función lineal de *x* como se observa en la función 1 esta función representa el modelo de regresión lineal [5] y se aplica dentro de la librería *numpy* de *phyton* por medio de la función *LogisticRegression*. Aquí, las θ i son los parámetros (también llamados pesos) que parametrizan el espacio de mapeo de funciones lineales de X a Y. Cuando no hay riesgo de confusión, eliminaremos el subíndice θ en h θ (x), y lo escribiremos más simplemente como h (x) que será la variable dependiente y x1, x2 y Xn son variables explicativas.

Cuando tenemos una variable de tipo binaria y la codificamos en = y 1, el ajuste del modelo de regresión lineal se puede realizar por mínimos cuadrados pero en estos casos al existir múltiples variables algunos valores podrían brindar resultados distintos a 0 y

1, por otra parte algunas probabilidades podrían estar fuera del intervalo entre [0,1] por este motivo la misma librería permite por medio de la aplicación de la función logística *sigmoide* transformar el valor devuelto por la regresión lineal en un valor comprendido entre [0,1]. Para ello se cambia la forma de la hipótesis h θ (x). Quedando como se detalla en la función 2 la hipótesis h θ (x).

$$h\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{2}$$

Dónde

$$g(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

En la función 3 vemos función logística o función sigmoidea [5]. La función sigmoidea es una función matemática que tiene una curva característica en forma de "S", que 5ransforma los valores entre el rango 0 y 1. La función sigmoidea también se llama curva sigmoidea o función logística. Es una de las funciones de activación no lineal más utilizadas ya que muchos procesos de aprendizaje automático, e incluso sucesos naturales muestran una aceleración considerable intermedia al cambio de estado con periodos de adaptación entre estados, esta evolución es descrita por la función sigmoide. Los datos transformados inicialmente al tratar las entradas o atributos de cada tupla de datos dan como resultado un rango continuo de valores. Sin embargo, para el proceso de clasificación requiere un rango discreto (0 ó 1, es decir sólo uno de dos posibles valores), el cual se puede obtener con la función de activación. Esta función tiene un comportamiento no lineal, y en el caso de la Regresión Logística se usa la función sigmoidal mostrada en la Función 3.

3.1 Diseño de Pruebas modelo de clasificación

Con respecto a la librería sklearn[2,3,4] se analiza la propuesta implementada del clasificador de regresión logística (también conocido como logit, MaxEnt).

Esta clase implementa la regresión logística regularizada usando la biblioteca "liblinear", los resolvedores"newton-cg", "sag", "saga" y "lbfgs".

2. Rodrigo, J. M., 2020. Machine learning con Python y Scikit-learn. Recuperado de: https://www.cienciadedatos.net/documentos/py06_machine_learning_ python_scikitlearn.html
3. scikit-learn developers. 2020. Modelos lineales, Linear Models. Recuperado de: https://scikit-learn.org/stable/modules/linear_model.html
4. scikit-learn developers. 2020. Manual de la librería sklearn.linear_model.LinearRegression. Recuperado de: https://scikit- learn .org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Los resolvedores "newton-cg", "sag" y "lbfgs" solo admiten la regularización L2 que es la seleccionada para las pruebas. El resolvedor "liblinear" admite la regularización L1 y L2, con una formulación dual solo para la penalización L2 por lo cual se opta por utilizar L2. La regularización de Elastic-Net solo es compatible con el solucionador de "saga". [6]

Teniendo en cuenta aquellas variables con mayor dependencia que finalmente son compromiso, gestión, requerimientos, validar, entiende cátedra y comunica cátedra. Cruzando con la variable que identifica si se recibieron o no. Se busca crear un modelo que permita identificar a cuál grupo pertenecen. Debemos considerar como en todo problema de clasificación en donde tenemos que etiquetar por ejemplo entre "positivos" o "negativos" o entre múltiples categorías (recibidos, no recibidos).

Solemos encontrar que en nuestro conjunto de datos de entrenamiento contamos con que alguna de las clases de muestra es una clase "minoritaria" es decir, de la cual tenemos muy poca cantidad de muestras. Esto provoca un desbalance en los datos que utilizaremos para el entrenamiento de nuestra máquina, se podría decir que el grupo de recibidos es menor que los no recibidos, pero el resultado de las encuestas por lo menos el 42,35%, al aislar solo estos datos para armar nuestro modelo vemos que hay datos faltantes por lo cual se procede a suprimir estos datos faltantes y se conservan 32 casos recibidos y 46 no recibidos siendo los recibidos un 41% de los datos los representan como se muestra en la Figura 4.

Para verificar el nivel de asertividad de nuestra propuesta podemos utilizar una matriz de confusión, trabajando con métricas: precisión y recall donde la precisión de una clase define cuán confiable es un modelo en responder si un punto pertenece a esa clase, y el recall de una clase expresa cuán bien puede el modelo detectar a esa clase.

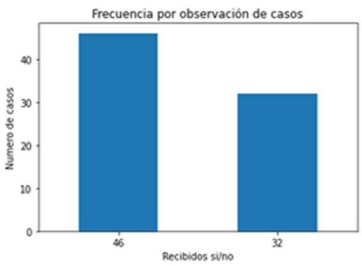

Fig. 4. Datos reducidos con atributos seleccionados según el grado de influencia

Se procede a crear los modelos definiendo las etiquetas y las características en las variables X e y y luego dividimos lo que sería el conjunto de entrenamiento y prueba (cabe destacar que se realizaron un total de 98 pruebas variando en el modelo características como el tamaño del conjunto de pruebas y de entrenamiento, el factor de penalti que determina la regularización del modelo y que si no se quiere aplicar se deja este parámetro en "*none*", los revolvedores y determinando el nivel de precisión brindado por el modelo donde en la tabla 5 se observa solamente un subconjunto de estas pruebas donde se filtraron los resultados donde se obtuvieron mejores valores de precisión.

3.2 Análisis del Modelo Clasificador propuesto

Con la finalidad de explicar el análisis realizado sobre cada uno de los 98 modelos es que a continuación en la en la Figura 5 vemos la matriz de confusión para nuestro modelo de clasificación propuesto como ejemplo, con un determinado conjunto de parámetros descriptos en la tabla 5 como prueba 8 (una de las 98 pruebas realizadas) y con class_weightdict=”none”. Se explican a continuación los datos extraídos de este modelo a fin de explicar el análisis realizado.

Se observa en la clase 1 correspondiente a los 46 no recibidos que se han reconocido bien 7 reales y 4 falsos (es lo que nos interesa detectar) vemos en la clase 2 de los 32 casos de recibidos 11 que fueron falsamente reconocidos como no recibidos y solo 2 correctamente reconocidos como recibidos.

La exactitud (Accuracy) del modelo es básicamente el número total de predicciones correctas dividido por el número total de predicciones. En este caso da (7+2)/(11+7+4+2)= 37.5%.

La Precisión de una clase define cuán confiable es un modelo en responder si un punto pertenece a esa clase. Para la clase no recibidos es 7/(7+11)=38.8% y recibidos es 2/(2+4)=33,3%

El Recall de una clase expresa cuán bien puede el modelo detectar a esa clase. Para no recibidos es de 7/(7+4)=63.6% y para recibidos es 2/(2+11)=15.3%.

El/La score/puntuación F1 de una clase se define como la media armónica de precisión y recall (2 x precision x recall / (precision+recall) siendo para no recibidos (2*0.388*0.636)/(0.388*0.636)=0.481 y para recibidos (2*0.333*0.153)/(0.333*0.153)=0.209 digamos que combina precisión y recall en una sola métrica. Por lo general en la matriz de confusión se pueden dar diferentes combinaciones

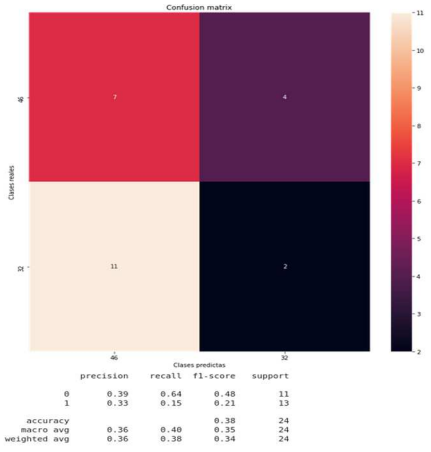

Fig. 5. Matriz de confusión para la propuesta del clasificador. Prueba 8 conjunto de entrenamiento 0.7. Tabla 5 y con class_weightdict=”none”, Modelo base sacando tuplas con datos faltantes

1. Caso 1: Alta precisión y alto recall: el modelo maneja perfectamente esa clase
2. Caso 2: Alta precisión y bajo recall: el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
3. Caso 3: Baja precisión y alto recall: El modelo detecta bien la clase, pero también incluye muestras de otras clases.
4. Caso 4: Baja precisión y bajo recall: El modelo no logra clasificar la clase correctamente.

En este caso podemos observar que contamos para los no recibidos nos encontramos en el caso 3 y para los recibidos, caso 4. Tomando criterios de bajo y alto con un corte en torno al 50% se podría decir que no son muy acertados los datos en esta clasificación, considerando que se propuso a modelo que tome un 70% de los datos para entrenamiento y el resto para prueba. También se puede observar que tenemos valor superior de precisión en la clase Mayoritaria que son los no recibidos y un bajo recall en la clase Minoritaria que son los recibidos.

Luego se realizan varias corridas del modelo variando únicamente el parámetro que indica el tamaño de datos para entrenamiento, teniendo un total de nueve pruebas registradas en la tabla 5, donde se observa que los mejores valores de rendimiento corresponden a las pruebas donde en cada uno de los datos observados se consiguió una precisión igual o mayor a 0.5. Como se observa que estos datos se obtuvieron con valores del conjunto de datos de 0.9 y 0.7 respectivamente es que se plantea realizar más pruebas variando otros parámetros que justifiquen mejor los resultados obtenidos.

Se plantea la posibilidad de mejorar el conjunto de entrenamiento ante un posible desbalance de las muestras. Se utilizará una estrategia de ajuste de parámetros del propio algoritmo para intentar equilibrar a la clase minoritaria penalizando a la clase mayoritaria durante el entrenamiento [1]. En logisticregression tenemos el parámetro class_weight= “balanced” un parámetro adicional en el modelo de Regresión logística en donde indicamos weight = “balanced” y con esto el algoritmo se encargará de equilibrar a la clase minoritaria durante el entrenamiento. Los resultados alterando este parámetro se observan en la tabla 5.

No se conocen márgenes de precisión numéricamente considerados como aceptables para cada modelo, ya que esto podría depender de la naturaleza de los datos, del problema analizado y de su comparativa con otros métodos de clasificación si existieran. En este trabajo se pretende conseguir una precisión mayor al 70% como aceptable y mayor al 90% como deseable. Esto es debido a que por medio de la encuesta se pretenden asignar recursos dentro de las cátedras para elevar los índices de los estudiantes pertenecientes al grupo de recibidos.

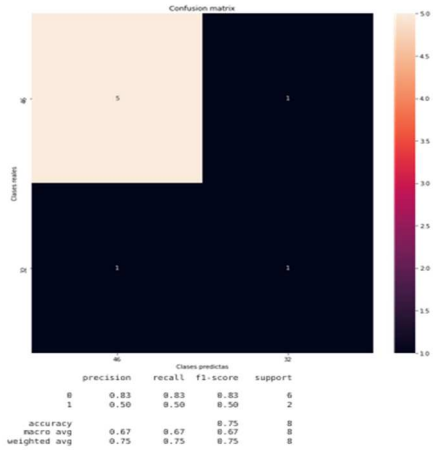

Fig. 6. Matriz de confusión para la propuesta del clasificador. Prueba 1 conjunto de entrenamiento 0.9 Tabla 5 class_weightdict =”none”. Modelo base sacando tuplas con datos faltantes.

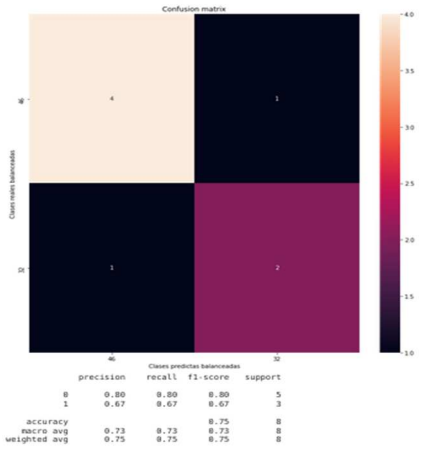

Fig. 7. Matriz de confusión para la propuesta del clasificador, penalizando a la clase mayoritaria. Prueba 1 Tabla 5 class_weightdict=”balanced”.

Table 5. Subconjunto de pruebas realizadas variando el conjunto de entrenamiento y con class_weightdict=”none”, variando el conjunto de entrenamiento y con class_weightdict=”balanced”, variando el conjunto de entrenamiento con solver lbfgs y con class_weightdict=”balanced” y variando el conjunto de entrenamiento con solver liblinear y con class_weightdict =”balanced”

Prueba	trainign set	penalty	C	random_state	solver	class_weightdict	precision clase 1	precision clase 2	recall clase 1	recall clase 2	f1 score clse 1	f1 score clase 2	support1	support2	accuracy
1	0,9	l2	1	1	newton-cg	none	0,83	0,5	0,83	0,5	0,8	0.50	6	2	0,75
9	0,7	l2	1	1	newton-cg	none	0,79	0,4	0,83	0,33	0,81	0,36	18	6	0,71
4	0,7	l2	1	1	newton-cg	none	0,6	0,75	0,92	0,27	0,73	0,4	13	11	0,63
8	0,7	l2	1	1	newton-cg	none	0,39	0,33	0,64	0,15	0,48	0,21	11	13	0,38

5	0,9	l2	1	1	newton-cg	balanced	0,67	1	1	0,5	0,8	0,67	4	4	0,75
1	0,9	l2	1	1	newton-cg	balanced	0,8	0,67	0,8	0,67	0,8	0,67	5	3	0,75
7	0,8	l2	1	1	newton-cg	balanced	0,71	0,67	0,62	0,75	0,67	0,71	8	8	0,69
4	0,8	l2	1	1	newton-cg	balanced	0,6	0,73	0,5	0,8	0,55	0,76	6	10	0,69
5	0,8	l2	1	1	lbfgs	balanced	1	0,71	0,82	1	0,9	0,83	11	5	0,88
8	0,7	l2	1	1	lbfgs	balanced	0,8	0,43	0,5	0,75	0,62	0,55	16	8	0,58
9	0,7	l2	1	1	lbfgs	balanced	0,53	0,44	0,62	0,36	0,57	0,4	13	11	0,50
7	0,7	l2	1	1	lbfgs	balanced	0,56	0,25	0,6	0,22	0,58	0,24	15	9	0,46
1	0,9	l2	1	1	liblinear	balanced	1	0,75	0,8	1	0,89	0,86	5	3	0,88
7	0,7	l2	1	1	liblinear	balanced	0,73	0,78	0,85	0,64	0,79	0,7	13	11	0,75
4	0,8	l2	1	1	liblinear	balanced	0,86	0,44	0,55	0,8	0,67	0,57	11	5	0,63
3	0,9	l2	1	1	liblinear	balanced	0,6	0,67	0,75	0,5	0,67	0,57	4	4	0,63

Con la propuesta del modelo balanceado se consiguieron los siguientes datos. Luego se realizaron más pruebas variando básicamente los parámetros *trainign_set*, *solver* y *class_weight*. Considerando como comparativa la precisión (accuracy). Se observa que los mejores valores se logran con el resolvedor lbfgs, class_weight balanced y training_set=0,8 como se puede ver en la tabla 5 y con resolvedor liblinear, class_weight balanced y training_set=0.9 como se observa en la tabla 5.

En ambos modelos se logra una precisión de 88% pero en el segundo caso es donde esta precisión mantiene valores elevados, por encima del 50% en 7 de los 9 casos de prueba contra 3 de 9 casos de prueba. Por este motivo se podría decir que el resolvedor liblinear es el que más se ajusta para este conjunto de datos, lo cual según la bibliografía de la librería es el resolvedor recomendado cuando se cuenta con un conjunto de datos pequeños.

4 Conclusiones

En el presente trabajo se toma un conjunto de datos relevados de datos de encuestas realizadas a estudiantes de carreras de grado que se encuentran en la etapa de realización de su proyecto integrador de desarrollo de software. Luego se cruzan estos datos con el estado o situación final de cada uno de los estudiantes al momento de realización del presente trabajo determinando si se han recibido o no.

Se logra establecer por medio del análisis de los datos que existen relaciones entre los datos relevados según los diagramas analizados en las figuras 2 y 3 estableciendo cruce entre las variables que demostraron tener relaciones más marcadas en el mapa de calor. Estas relaciones en primera instancia no parecen determinar características estrictamente diferenciales entre ambos grupos, pero sí permiten identificar algunas tendencias como ser los problemas al momento de la gestión del proyecto, el compromiso con el mismo, la capacidad de relevar y validar requerimientos y la comunicación con el cliente como los más relevantes.

Al momento de clasificar los grupos se presenta la propuesta de un modelo de regresión logística debido a la naturaleza dicotómica de los grupos. Para ellos se prueban diferentes parámetros de modelo y se logra presentar una propuesta con un margen de precisión de 88% en la predicción de la pertenencia a los grupos como se observan en la Tabla 5. Lo cual se considera razonablemente aceptable debido a que se obtienen márgenes de precisión mayores a un 70% que era lo esperado inicialmente.

En cuanto a la importancia de las variables. Cabe destacar como se observa en la figura 2 en el mapa de calor realizado para definir las características que se usarían para el modelo de clasificación de regresión lineal, esta selección de variables se realizó incluyendo los valores más altos donde 0.7 se observa entre en trabaja y horas de trabajo, siendo un factor importante que el estudiante trabaje y dedique horas de trabajo en su rutina diaria, siguiendo con requerimientos y validar, se puede destacar como entender, identificar y validar requerimientos es un factor altamente relacionado a la culminación de los trabajos integradores finales con los cuales se gradúan, siguiendo gestión con compromiso con un valor de 0.6, lo cual indica la importancia en la gestión del tiempo y el compromiso que asumen en la realización del proyecto, lo mismo para *entiendecatedra* y *comunicacátedra* que se relaciona a entender lo que se espera de sus trabajos y la comunicación con la catedra que es la que realiza el trabajo de tutoría o guía académica.

5 Trabajos futuros

A partir del análisis realizado en el presente trabajo se pretende seguir realizando las encuestas para recaudar año a año más datos de manera permanente. También analizar otras variables relevadas en la encuesta, como también las incluidas en este trabajo que al aislarlas del resto podrían determinar dependencias más fuertes. Proponer y comparar los resultados obtenidos con otros clasificadores de Máquinas de Vector Soporte (SVM) que implementan otros kernels (como ser radial o redes neuronales NN). Probar

el modelo con otras técnicas de validación como Validación Cruzada (Cross Validation). Finalmente luego de probar otros clasificadores se pretende llevar el modelo a una solución integrada para su ejecución mediante una plataforma de consulta.

6 Referencias

1. Rambo, A., Kuna, H., Sueldo, R., Urquijo, R., Piotroski F. 2018. "Análisis de factores de éxito para gestión de proyectos académicos unipersonales de práctica profesional supervisada en carreras de informática". Congreso Virtual Educa Innovación, Desarrollo, Inclusión. 2018. Argentina. ISBN 978-959-312-332-7
2. Rambo, A. R., Boari, M. I., Sueldo, R. L., Urquijo. 2019. "Análisis de indicadores de la práctica profesional supervisada en carreras de informática de la FCE-QyN – UNaM". " Décimo Sexto Simposium Iberoamericano en Educación, Cibernética e Informática: SIECI 2019. Décima Octava Conferencia Iberoamericana en Sistemas, Cibernética e Informática: CISCI 2019. 6 al 9 de Julio de 2019. Orlando, Florida, EE.UU. Memorias de la Décima Octava Conferencia Iberoamericana en Sistemas, Cibernética e Informática: CISCI 2019. Vol II. ISBN - Collection: 978-1-950492-14-5. ISBN - Volume II:978-1-950492-22-0. Recuperado de: http://www.iiis.org/CDs2019/CD2019Summer/PapersC2.htm#/
3. Rambo, A. R., Boari, M. I., Sueldo, R. L., Urquijo, R., Chripczuk, H., Ramirez, U. 2020. "Prototipo de dinámica de sistemas aplicado a la gestión de proyectos académicos de práctica profesional supervisada en carreras de informática.". "Engenharia na Prática: Importância Teórica e Tecnológica". Ponta Grossa – PR. Brasil. DOI: 10.22533/at.ed.088202408. ISBN: 978-65-5706-308-8. https://www.atenaeditora.com.br/post-ebook/3440
4. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. 2000 CRISP-DM. Step-by-step data mining guide. CRISP-DM consortium: NCR Systems Engineering Copenhagen (USA and Denmark), DaimlerChrysler AG (Germany), SPSS Inc. (USA), and OHRA Verzekeringen en Bank Groep B.V. (The Netherlands). Recuperado de: https://www.the-modeling-agency.com/crisp-dm.pdf
5. Ng, A. Supervised learning. CS229 - Machine Learning. Stanford University, Stanford, California 94305. Recuperado de: https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf
6. Bagnato, J. I. 2019. blog Aprende Machine Learning. Clasificación con datos desbalanceados. Recuperado de: https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/