

Desarrollo de un modelo predictivo como soporte para la revisión de cuestiones lingüísticas y de estilo de los recursos ordinarios de apelación

Mariano Cervellini¹, Miguel Carbone¹

¹ Fiscalía de Estado de la Provincia de Buenos Aires, Av. 1 esq 60, 1342
La Plata (1900), Argentina
{cervellini, carbone}@fepba.gov.ar

Abstract

El presente trabajo aborda el análisis de técnicas de aprendizaje automático aplicadas a la etapa de revisión de los recursos ordinarios de apelación, confeccionados por agentes de Fiscalía de Estado de la Provincia de Buenos Aires en el marco de las causas judiciales en las que interviene.

Tomando como parámetro determinados acuerdos lingüísticos y de estilo, se realizó una prueba de concepto en la que se los clasificó automáticamente entre aquellos que CUMPLEN y NO CUMPLEN con tales acuerdos, mediante la aplicación de dos métodos: bolsa de palabras y determinación de *features* a partir métricas preestablecidas.

La utilización de este tipo de herramientas posibilita que el/la funcionario/a interviniente en los distintos circuitos de revisión, enfoque sus mayores esfuerzos en el análisis y consideración de las cuestiones formales y sustanciales según la ley vigente del escrito que CUMPLE, mientras que aquel que NO CUMPLE, debe ser reexaminado por el profesional encargado de su elaboración.

De esa manera se logra una mejora en la actuación estatal y el cumplimiento de sus funciones, sin relegar el control que el/la funcionario/a continúa realizando con la ayuda del modelo predictivo.

Keywords: Machine Learning, Clasificación automática de textos, Automatización, Aprendizaje supervisado.

1. Introducción

El Fiscal de Estado de la Provincia de Buenos Aires es quien tiene a su cargo el mandato constitucional de tutelar los intereses patrimoniales de la provincia y ejercer su representación en juicio ante cualquier fuero o jurisdicción (art. 155 de la Constitución provincial y Decreto Ley N° 7543/79).

Con una población de 15.625.000 de habitantes en 2010, la provincia de Buenos Aires es la jurisdicción más poblada de la Argentina, sus 307.571 km² la hacen la más extensa luego de la de Tierra del Fuego, Antártida e Islas del Atlántico Sur, y con 50,8 hab/km² es la tercera jurisdicción más densamente poblada luego de la Ciudad Autónoma de Buenos Aires y la provincia de Tucumán¹.

La contundencia de las cifras resulta por demás suficiente para tener una idea aproximada de la complejidad y el caudal de la labor que desempeña este Organismo, como así también de su envergadura, considerando el personal técnico especializado y los recursos técnicos y materiales necesarios para llevar adelante tan delicado e importante rol institucional.

En razón de tales características, a través de la presente experiencia buscamos visibilizar la potencialidad que el uso de un modelo predictivo puede reportar para agilizar y mejorar su desempeño mediante la clasificación automática de un gran número de escritos judiciales en tan solo unos pocos segundos.

Para ello y a través del desarrollo de una prueba de concepto (POC, por sus siglas en inglés), se ha seleccionado un grupo de los escritos judiciales confeccionados a diario denominados **recursos ordinarios de apelación**, y tomando como parámetro la utilización de determinados acuerdos lingüísticos y de estilo, hemos logrado agruparlos automáticamente en dos categorías: CUMPLE y NO CUMPLE.

Postulamos que a partir de esta clasificación el personal técnico especializado que interviene en los distintos circuitos de revisión, podrá enfocar sus mayores esfuerzos en el análisis y consideración de las cuestiones formales y sustanciales según la ley vigente del escrito que CUMPLE, mientras que aquel que NO CUMPLE deberá ser reexaminado por el profesional encargado de su elaboración antes de continuar el circuito de revisión.

2. El circuito de revisión de los recursos ordinarios de apelación y su relevancia

Por su ley orgánica² el Fiscal de Estado aprueba la Estructura Orgánico Funcional y el Plantel Básico según las necesidades correspondientes a cada ejercicio anual.

En la actualidad, se compone por dos Fiscales Adjuntos y seis Subsecretarías con funciones contenciosas, además de un Delegado Fiscal por cada Departamento Judicial -salvo el de La Plata- y en la Ciudad Autónoma de Buenos Aires, sobre la base de un plantel de profesionales especializados en su mayoría en materia jurídica, complementado por personal de apoyo.

Si bien los agentes que integran el cuerpo profesional son los encargados de confeccionar los escritos judiciales de los casos que les son asignados y realizar las pre-

¹ Disponible: https://es.wikipedia.org/wiki/Provincia_de_Buenos_Aires

² Disponible: <http://www.gob.gba.gov.ar/legislacion/legislacion/l-7543.html>

sentaciones judiciales ante los Tribunales competentes; la ley determina ciertos actos procesales en los que por su importancia el Fiscal de Estado actúa como patrocinante³.

En efecto, al revestir el **recurso ordinario de apelación** un acto determinante para la suerte final del juicio⁴, el patrocinio está a cargo del Fiscal de Estado, aunque la ley prevé que fuera del Departamento Judicial de La Plata (asiento de su sede) puede sustituirse en la interposición de los mismos, lo que en la práctica sucede en favor de los Delegados Fiscales, o en su caso, de los apoderados fiscales externos que actúan en los juicios de apremio.

En el primer caso el Fiscal tiene a su cargo la revisión final del escrito judicial previa intervención de el/la funcionario/a de la Subsecretaría con competencia en la materia⁵, mientras que cuando actúen los sustitutos o apoderados serán éstos los responsables de tal extremo, bajo la supervisión de determinadas Subsecretarías⁶ con funciones a ese efecto⁷.

3. La importancia de un correcto uso del lenguaje y el cumplimiento de determinados acuerdos lingüísticos y de estilo

La magnitud de la labor que se lleva a cabo obliga a contar con herramientas que permitan optimizar tiempo y recursos en pos de abastecer adecuadamente la forma y el contenido de cada uno de los actos procesales que se suceden durante la tramitación de un juicio.

Cuando en este trabajo hablamos de la forma de los escritos, no nos referimos a los requisitos legales establecidos a ese efecto en los Códigos de Procedimiento⁸, sino

³ En los escritos de demanda, contestación y reconvencción, oposición y contestación a excepciones, pedidos de disponibilidad y entrega de fondos a terceros, pedidos de venta en los juicios de herencias vacantes, interposición de recursos contra sentencias definitivas que deban presentarse fundados, memorias expresiones y contestaciones de agravios y deducción de recursos extraordinarios ante la Suprema Corte de Justicia de la Nación.

⁴ El éxito de un proceso depende muchas veces de la correcta utilización de los recursos, siendo que quien domine esa técnica podrá revertir resultados adversos, y quien la desatienda puede llegar a perder el juicio, por dejar firme – involuntariamente- resoluciones contrarias a su derecho -LOS RECURSOS ORDINARIOS EN EL CPCA (SAFI).pdf p 1 Por Leandro K. Safi.Publicado en Revista EL DERECHO (29/11/2013)-.

⁵ Judicial en lo Contencioso Administrativo I y II, Judicial La Plata I y II.

⁶ De Interior y de Ejecución de Créditos Fiscales y Tributarios y Judicial en lo Contencioso Administrativo I y II en lo que hace a su materia específica.

⁷ Ello sin perjuicio de las directivas generales y particulares que el Fiscal establece a diario para lograr el objetivo constitucional que signa su actuación.

⁸ ¿qué se puede apelar?, ¿quién puede apelar?, ¿cuándo se apela?, ¿cómo se apela?, ¿cómo tramita la apelación?...

a la necesidad de prestar la atención que merece al correcto uso del lenguaje mediante la fijación de determinados acuerdos lingüísticos y de estilo.

“La forma bien aplicada es una herramienta que ayuda a cumplir el propósito último de que se haga justicia, porque acota los equívocos, los malos entendidos y los errores impermisibles desde el punto de vista del rigor de la ley adjetiva. (...) También la puntuación y el uso de recursos lingüísticos apropiados que sugiere. Desde su creación en la lejana Biblioteca de Alejandría, la puntuación en sus distintas formas sirvió en su origen para facilitar la respiración del lector en voz alta. Con el tiempo, ordenaron la diástole y la sístole pero también los pensamientos. Los recursos lingüísticos facilitan la expresión clara de los argumentos, y su lectura amable por el lector, sea juez o contraparte. Todo hace a un mejor modo de juzgar, ejercer y defender derechos.”⁹ -los destacados son propios-.

Es en función de lo expuesto que se presenta como una herramienta indispensable en cualquier tipo de organización que realiza su labor a través de la redacción de piezas escriturales, confeccionar un Manual de Estilo¹⁰ que fije los consensos lingüísticos a utilizar por la organización, no sólo para mejorar su calidad y optimizar el mensaje, sino también a efectos de facilitar la correcta aplicación de técnicas de automatización como las que a través del presente trabajo estamos postulando, y que desarrollamos en los acápite siguientes.

4. El aprendizaje automático

La revisión de documentos basada en criterios predefinidos puede ser una tarea tediosa, repetitiva y propensa a errores cuando se trata de grandes cantidades de información.

El desarrollo de software tradicional no resuelve problemáticas como las planteadas en la sección anterior; para esto, es necesario utilizar herramientas de aprendizaje automático que permitan, a partir de un modelo debidamente entrenado, predecir una o más variables desconocidas.

El aprendizaje de máquinas o aprendizaje automático (ML, por sus siglas en inglés) consiste en extraer el conocimiento de los datos. Es un campo de investigación de las ciencias estadísticas, la inteligencia artificial, y la informática, y también es conocido como análisis predictivo o aprendizaje estadístico [1]. Según Tom M. Mitchell, el aprendizaje automático es el estudio de algoritmos informáticos que mejoran automáticamente a través de la experiencia [2]. Se señala asimismo que el término aprendizaje automático se refiere a la detección automatizada de patrones significativos en los datos [3].

Dado que el aprendizaje implica una interacción entre el aprendiz (modelo) y el entorno, se pueden dividir las tareas de aprendizaje de acuerdo a la naturaleza de esa

⁹ Extraído del Prólogo al Nuevo Manual de Estilo de la Procuración del Tesoro de la Nación - Ed. 2018.

¹⁰ Actualmente en elaboración.

interacción. La principal distinción a destacar es la diferencia entre el aprendizaje supervisado y el no supervisado.

- a) **Aprendizaje supervisado.** Permite asignar una clase a un nuevo elemento en base a un conjunto de categorías previamente establecidas. Se basa en un entrenamiento con ejemplos con la solución conocida (supervisado) para crear modelos que permitan clasificar nuevos casos análogos. En esta configuración la experiencia adquirida tiene por objetivo predecir la información faltante. Aquí podemos pensar el entorno como un maestro que "supervisa" al aprendiz proporcionando información adicional (etiquetas) [3, p. 23]. Algunos ejemplos de algoritmos de aprendizaje supervisado son: Linear Regression, Logistic Regression, Decision Tree Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree Classification, etc.
- b) **Aprendizaje no supervisado.** Sin embargo, en el aprendizaje no supervisado, no hay distinción entre los datos de entrenamiento y de prueba. El aprendiz procesa los datos de entrada con el objetivo de generar un resumen o una versión comprimida de esos datos [3, p. 23]. Las clases o categorías no se conocen de antemano, sino que son determinadas en base a las propias similitudes entre los elementos; es decir, se centran en la "medida de similitud" entre elementos. Algunos ejemplos de algoritmos de aprendizaje supervisado son: K-Means Clustering, Hierarchical Clustering, Principal Component Analysis, etc.

5. Representación del documento y validación del modelo predictivo

5.1. Representación

Los algoritmos de aprendizaje automático no pueden procesar directamente los documentos de texto en su forma original, ya que la mayoría de ellos esperan vectores de *features*¹¹ numéricos de tamaño fijo, en lugar de documentos de texto de longitud variable. Por lo tanto, durante el preprocesamiento, los documentos de texto son convertidos a una representación acorde.

¹¹ Es una propiedad individual medible o característica de un fenómeno que se observa. Disponible: [https://en.wikipedia.org/wiki/Feature_\(machine_learning\)](https://en.wikipedia.org/wiki/Feature_(machine_learning))

Uno de los métodos comúnmente utilizadas para la representación de los documentos y ampliamente aplicado en la categorización de textos, es el método de **bolsa de palabras** (BoW, por sus siglas en inglés).

El concepto de "bolsa de palabras" es mencionado por primera vez por el lingüista Zellig Harris en 1954 [4]. El modelo BoW es una representación simplificada utilizada en el procesamiento del lenguaje natural (NLP, por sus siglas en inglés) y la recuperación de información (IR). En la práctica, se utiliza principalmente como herramienta de generación de *features*. Luego de transformar el texto en una "bolsa de palabras", es posible calcular varias medidas para caracterizar el texto. El tipo más común de *features* a partir del modelo de bolsa de palabras es la frecuencia de los términos, es decir, la cantidad de veces que aparece un término en el texto [5]. Asimismo permite un modelado basado en diccionario, donde cada documento parece una bolsa (no se considera el orden), que contiene algunas palabras del diccionario [6, p. 39].

La representación de bolsa de palabras es fácil de generar, pero está lejos de ser perfecta [7]. Las frecuencias de términos no son necesariamente la mejor representación de un documento. Palabras comunes como "el", "a", "para", casi siempre tienen la frecuencia de término más alta en el texto, por lo que tener un recuento alto no significa necesariamente que la palabra correspondiente sea más importante.

Para abordar este problema, una de las formas populares de "normalizar" las frecuencias del término es ponderar un término por el *inverso de la frecuencia del documento* [5] (Tf-Idf, del inglés Term frequency – Inverse document frequency). Se trata de una medida numérica que expresa cuán relevante es una palabra para un documento de una colección y consiste en un simple giro en el enfoque de la bolsa de palabras. En lugar de ver los recuentos en bruto de cada palabra de cada documento en un dataset, tf-idf mira un recuento normalizado donde cada recuento de palabras se divide por el número de documentos en que aparece esta palabra [7].

Como método alternativo para la representación de los documentos se puede trabajar en la **generación de métricas** que permitan describir y obtener los *features* de cada documento.

5.2. Validación

Por otra parte, existen diversas técnicas para validar un modelo, comúnmente se utiliza la técnica de validación cruzada (cross-validation)¹², dividiendo el set de entrenamiento en dos partes (30% testing, 70% training) y entrenando el clasificador con la porción más larga, mientras que la porción más pequeña (validation set) es utilizada para realizar el test de validación. Este método funciona muy bien, pero presenta la desventaja de que el modelo no es entrenado y validado con todos los casos del dataset.

¹² La validación cruzada (cross-validation) es un procedimiento de remuestreo que se utiliza para evaluar modelos de aprendizaje automático en una muestra de datos limitada.

Una variante más sofisticada es el método conocido como *k-fold cross-validation*, a partir de cuya aplicación se divide el dataset en k partes (folds), el proceso se repite k veces (para cada fold), utilizando un subconjunto distinto de datos para validar el modelo entrenado. Luego se promedia el puntaje medido para cada fold, para obtener una estimación más precisa del rendimiento del modelo.

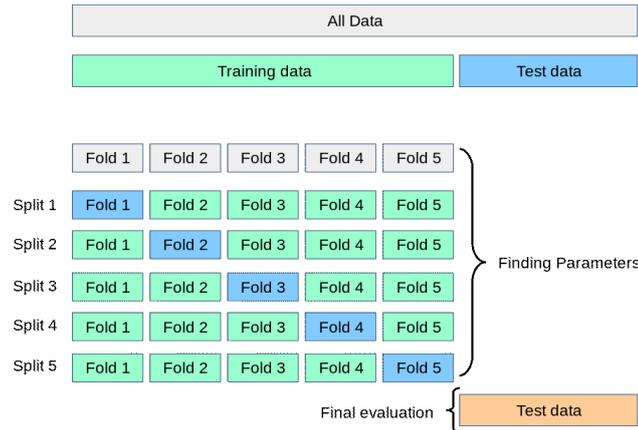


Fig. 1. Cross-validation
 fuente: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

6. Caso de Estudio

6.1. Preliminares

A través del presente trabajo postulamos la aplicación de un modelo predictivo como herramienta de apoyo en los distintos circuitos de revisión que se aplican en el ámbito de la Fiscalía de Estado de la Provincia de Buenos Aires para la elaboración de los recursos ordinarios de apelación.

En base a la determinación de ciertos acuerdos lingüísticos y de estilo, el modelo posibilita que el/la funcionario/a interviniente enfoque sus esfuerzos en el análisis y consideración de las cuestiones formales y sustanciales según la ley vigente del escrito que CUMPLE con aquellos, mientras que aquel que NO CUMPLE debe ser reexaminado por el profesional encargado de su elaboración antes de continuar el circuito de revisión.

6.2. Representación de documentos y métodos utilizados

Durante la etapa experimental se trabajó en la implementación de varios modelos predictivos desarrollados por completo en *python*¹³, utilizando la librería *scikit-learn*¹⁴.

Para comenzar, se seleccionó un conjunto de recursos ordinarios en soporte digital que sirvieron luego para entrenar el modelo. Como hemos advertido, los algoritmos de aprendizaje automático no pueden procesar directamente el texto contenido en los documentos, ya que esperan como entrada vectores numéricos. Por lo tanto, resultó necesario representar el contenido de los documentos utilizando solo números. A tal fin, se adoptaron dos métodos, ello en pos de observar el rendimiento y efectividad de cada uno.

En primer término, se aplicó el método de bolsa de palabras comúnmente utilizado en la clasificación de documentos de texto para la extracción de *features*; y en segundo lugar, se definieron los *features* a partir métricas preestablecidas. En ambos casos se obtuvo un vector numérico para representar cada documento.

Para la construcción de las métricas se analizaron las presentadas en [8] y se definieron otras a partir de [9]. Las métricas seleccionadas permitieron identificar buenas prácticas de escritura (cómo se deberían escribir las cosas) y fallas de escritura (que se debería evitar). Esta metodología resultó ser mucho más compleja que la implementada con BoW, ya que cada métrica debió programarse para obtener los valores que permitieron representar cada recurso. La Tabla 1 muestra los features que componen nuestra representación de los documentos.

Tabla 1. Features del documento

Feature	Descripción
M1	Cantidad de palabras con errores ortográficos
M2	Cantidad de palabras informales/coloquiales (“slang”)
M3	Cantidad de párrafos
M4	Cantidad de párrafos extensos (mayores a 6-7 renglones)
M5	Cantidad de palabras repetidas por párrafos
M6	Cantidad de adjetivos con repetición en el texto
M7	Cantidad de adverbios con repetición en el texto
M8	Cantidad de palabras
M9	Longitud promedio de palabras en caracteres
M10	Cantidad de aclaraciones entre comas (o paréntesis) extensas

¹³ Python es un lenguaje de programación interpretado multiplataforma. Disponible: <https://www.python.org>.

¹⁴ Librería software libre para aprendizaje automático para el lenguaje de programación Python. Disponible: <https://scikit-learn.org>.

M11	Cantidad de párrafos que contienen palabras repetidas
M12	Cantidad de párrafos en los que repite comas más de 6 veces

6.3. Entrenamiento y validación

Una vez realizada la selección de los recursos ordinarios y definida la representación de los documentos, se continuó con la etapa de entrenamiento del modelo predictivo, en la cual se utilizó la técnica de aprendizaje supervisado. En primer lugar, para elaborar el dataset de entrenamiento se creó una carpeta que se denominó *train*, compuesta de dos subcarpetas *cumple* y *no-cumple*, en las cuales se guardaron los recursos ordinarios que fueron clasificados por el experto. Dentro de la subcarpeta *cumple* se colocaron los documentos que se ajustaban a los acuerdos de lingüísticos y de estilo, y en la subcarpeta *no-cumple* se guardaron los recursos que no lo hacían. De esta manera se obtuvo un corpus de 75 (29 que CUMPLEN y 46 que NO CUMPLEN) recursos ordinarios en formato .DOC¹⁵.

Luego, se utilizaron dos *scripts* que permitieron extraer el texto y la categoría asociada (carpeta en la que se encontraba el documento) de cada uno. Estos *scripts* generaron los *dataset* de entrenamiento, uno en el que los documentos fueron representados utilizando el método BoW, y otro para el método basado en métricas.

Para entrenar el modelo predictivo se programaron dos *scripts* que reciben como entrada los datos del dataset de entrenamiento. Con estos datos se realizaron una serie de pruebas: para el método basado en métricas se utilizó el clasificador Logistic Regression¹⁶ y para el método basado en BoW, se probaron los clasificadores SGDClassifier¹⁷, Logistic Regression, y SVC¹⁸. Cada uno de los clasificadores debió configurarse adecuadamente para obtener los mejores resultados posibles. La clase GridSearchCV¹⁹ de scikit-learn posibilitó determinar los hiperparámetros²⁰ óptimos para de

¹⁵ Especifica el formato de archivo binario de Word (.doc), usado por Microsoft Word 97, Microsoft Word 2000, Microsoft Word 2002, and Microsoft Office Word 2003. Disponible: <https://msdn.microsoft.com/en-us/library/cc313153%28v=office.12%29>

¹⁶ Clasificador Logistic Regression. Disponible: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

¹⁷ Clasificador SGDClassifier. Disponible: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.

¹⁸ Clasificador SVC. Disponible: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>.

¹⁹ Clase disponible en scikit-learn, permite evaluar y seleccionar de forma sistemática los parámetros de un modelo.

²⁰ Los hiperparámetros son parámetros que no se aprenden directamente dentro de los estimadores. En scikit-learn se pasan como argumentos al constructor de las clases de estimador. Algunos ejemplos típicos pueden ser C, kernel y gamma para Support

cada configuración. Cada *script*, además, permitió generar una salida de un archivo .CSV²¹ en la que se pudo observar la clasificación (probabilidad de pertenecer a la clase CUMPLE y NO CUMPLE) de los documentos que se encuentran en la carpeta *test*, que no fueron utilizados para entrenar el modelo.

Con respecto a la validación, al tratarse de un corpus reducido (75 casos), los modelos fueron validados utilizando la técnica *k-fold cross-validation*, que como se mencionó anteriormente, permite entrenar y validar el modelo con el dataset de entrenamiento completo.

6.4. Análisis de Resultados

Para evaluar el desempeño de cada uno de los modelos se utilizaron las medidas Precision, Recall y F1, y los hiperparámetros usados para cada modelo fueron los obtenidos por medio de una búsqueda *grid search*.

En la Tabla 2 se observan los resultados arrojados de la validación del clasificador Stochastic Gradient Descent (SGD, por sus siglas en inglés). Las pruebas fueron realizadas implementando un pipeline²² en el que se aplicaron las siguientes transformaciones: CountVectorizer, TfidfTransformer y finalmente SGDClassifier.

Tabla 2. Representación utilizando método BoW y clasificador SGDClassifier

	precision	recall	f1-score
cumple	1.00	0.91	0.95
no-cumple	0.92	1.00	0.96
avg / total	0.96	0.96	0.96

En la Tabla 3 se observan los resultados arrojados de la validación para el clasificador Logistic Regression²³. Las pruebas fueron realizadas implementando un pipeline en el que se aplicaron las siguientes transformaciones: CountVectorizer, TfidfTransformer y finalmente Logistic Regression.

Tabla 3. Representación utilizando método BoW y clasificador Logistic Regression

	precision	recall	f1-score
cumple	1.00	0.55	0.71
no-cumple	0.71	1.00	0.83

Vector Classifier, alpha para Lasso, etc. Disponible: https://scikit-learn.org/stable/modules/grid_search.html

²¹ Comma Separated Values File

²² Aplica secuencialmente una lista de transformaciones y un estimador final.

²³ Logistic Regression. Disponible: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

avg / total	0.85	0.78	0.77
--------------------	------	------	------

En la Tabla 4 se observan los resultados arrojados de la validación del clasificador C-Support Vector Classification²⁴. Las pruebas fueron realizadas implementando un pipeline en el que se aplicaron las siguientes transformaciones: CountVectorizer, TfidfTransformer y finalmente C-Support Vector Classification.

Tabla 4. Representación utilizando método BoW y clasificador SVC

	precision	recall	f1-score
cumple	1.00	0.91	0.95
no-cumple	0.92	1.00	0.96
avg / total	0.96	0.96	0.96

Finalmente, en la Tabla 5 se observan los resultados arrojados de la validación del clasificador Logistic Regression basado en métricas.

Tabla 5. Resultados basado en métricas y clasificador Logistic Regression

	precision	recall	f1-score
cumple	0.84	0.88	0.86
no-cumple	0.77	0.71	0.74
avg / total	0.81	0.82	0.81

7. Conclusiones

De acuerdo con lo expuesto anteriormente, puede observarse que los mejores resultados fueron logrados con los modelos en los que se utilizó BoW para la obtención de los *features*. Esto se encuentra en concordancia con lo indicado en una serie de experimentos en los que se ha sostenido que las representaciones más sofisticadas no producen una eficacia significativamente mejor [10, p.10].

Sin embargo, es apresurado adoptar una conclusión tan tajante en nuestro caso, ya que se trata de las primeras pruebas realizadas sobre un modelo predictivo basado en métricas, al cual deben sumarse otras e incorporarse al corpus un mayor número de documentos.

En cualquier caso, la aplicación de estos modelos permite alcanzar una mejora en la actuación estatal y el cumplimiento de sus funciones, sin relegar el control del/a experto/a.

²⁴ C-Support Vector Classification. Disponible: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

El equipo de trabajo se encuentra abocado a mejorar el modelo predictivo y está evaluando la posibilidad de aplicar este tipo de herramientas a otros tipos de escritos judiciales. Asimismo, tiene previsto avanzar hacia la automatización del control vinculado al cumplimiento de las exigencias formales, y finalmente abordar el desafío que presenta el análisis de las cuestiones vinculadas a su sustancia o contenido.

8. Referencias

- [1] Andreas C. Müller & Sarah Guido. Introduction to Machine Learning with Python. A Guide for Data Scientists. O'Reilly (2017).
- [2] Tom M. Mitchell. The Discipline of Machine Learning. July 2006 [Online]. Disponible: <http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf> [Accedido: 24-Mar-2019].
- [3] Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. 2014. Cambridge University Press. [Online]. Disponible: <http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf> [Accedido: 24-Mar-2019].
- [4] Harris, Zellig (1954). "Distributional Structure". Word. 10 (2/3): 146–62.
- [5] Bag-of-words model [Online]. Disponible: https://ipfs.io/ipfs/OmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Bag_of_words_model.html [Accedido: 20-Mar-2019].
- [6] Proyecto Final de Carrera: Aplicación del modelo Bag-of-Words al reconocimiento de imágenes. Julio del 2009. Consultado el 11 de noviembre de 2014. [Online]. Disponible: https://e-archivo.uc3m.es/bitstream/handle/10016/7107/PFC_Sara_Pardo_Feijoo.pdf?sequence=1 [Accedido: 18-Mar-2019].
- [7] Amanda Casari, Alice Zheng. Feature Engineering for Machine Learning. O'Reilly. Abril 2018. [Online]. Disponible: <https://www.oreilly.com/library/view/feature-engineering-for/9781491953235/ch04.html> [Accedido: 21-Mar-2019].
- [8] Edgardo Ferretti, Leticia Cagnina, Viviana Paiz, Sebastián Delle Donne, Rodrigo Zacagnini, Marcelo Errecalde. Quality flaw prediction in Spanish Wikipedia: A case of study with verifiability flaws. Diciembre de 2017.
- [9] Nuevo Manual de Estilo de la Procuración del Tesoro de la Nación (PTN). [Online]. Disponible: <https://www.ptn.gob.ar/images/files/NUEVO%20MANUAL%20ESTILO.pdf>. [Accedido: 17-Mar-2019].
- [10] Machine Learning in Automated Text Categorization. Fabrizio Sebastiani. Diciembre de 1999.
- [11] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[12] Scikit-learn, Cross Validation. [Online]. Disponible: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation. [Accedido: 17-Mar-2019].