Universidad de San Andrés

Las palabras que rigen las leyes

Cómo predecir si una ley será aprobada con métodos de Machine Learning

Matías Güizzo Altube
Santiago Rolón
Santiago Papasidero

Primavera 2020

Big Data
Profesores: Walter Sosa Escudero y Carla Srebot

congreso pueblos bicentenario libres nacional presente sus proyecto este na
mayor esta derechos parte poder senado entre general derecho información p
ollo sin ejecutivo todo años estado personas cada social hasta argentina país
mbién sistema servicios son así ha educación forma señor cuando fue tod
ón día acceso medidas caso desde aplicación tiene través calidad todas fu
ma publicaciones si cámara otros provincia dentro mismo parlamentaria se
rabajo ministerio políticas tanto días ello puede proceso cual federal fin salud a
niños será mediante seguridad articulo actividad público servicio condicion
rotección han convención evaluación nuestro pública ciudad cuenta donde s

I. INTRODUCCIÓN

Cualquier ley emitida por un congreso comienza con un proyecto de ley, un texto escrito y propuesto por un congresista, que es analizado por una comisión o comisiones afines antes de proceder a votación por el cuerpo de legisladores. En Argentina, en el año 2019 fueron presentados 3377 proyectos de ley, de los cuales solo 40 fueron aprobados[1]. Esta tasa de aprobación ínfima podría interpretarse de muchas maneras. Por un lado, podemos pensar que la falta de consenso o la polarización política genera barreras al momento de votar. Por otro, podemos pensarlo como un problema de eficiencia. Durante todo el año, los diputados sesionaron 8 veces y los senadores 7. Es de esperarse que esto genere un cuello de botella significativo en el proceso legislativo. Además, para llegar a votación, cada proyecto debe atravesar las comisiones correspondientes, que pueden rechazarlo o modificarlo antes de llevarlo a votación.

En este trabajo, buscamos explorar una característica relacionada al segundo aspecto. Consideramos que el escrito original presentado por el congresista contiene ya una probabilidad intrínseca de ser aprobado o rechazado, ya sea por su contenido o por su estructura. Nuestro objetivo es entonces poder predecir si un proyecto será aprobado a partir de las palabras con las que fue escrito e identificar cuáles son las palabras con mayor carga positiva y negativa. Para esto, empleamos un modelo *logit* regularizado por *elastic net*, en el que utilizamos como predictores la cantidad de veces que aparece cada palabra en el texto y buscamos estimar la probabilidad de que el proyecto sea aprobado.

Para poner a prueba nuestro modelo, utilizamos una base no exhaustiva de proyectos propuestos por senadores argentinos en un estrecho intervalo de tiempo. Sin embargo, el método sugerido tiene la flexibilidad suficiente como para replicar este análisis en cualquier otro congreso del mundo (y de cualquier nivel), así como incorporar otros elementos para la predicción, relacionados con las condiciones políticas en las que se presenta y evalúa el proyecto, por ejemplo.

[1] Dato obtenido de https://www.cronista.com/economiapolitica/Durante-2019-solo-el-089-de-los-proyectos-se-convirtio-en-ley-20200108-0023.html

II. REVISIÓN DE LITERATURA

El método que utilizamos fue tomado casi directamente de Wu (2018). Esta autora realiza una aplicación muy similar de *logit* regularizado solo por *LASSO*, en la que busca identificar las palabras que mejor predicen si un comentario anónimo en un foro de trabajo es referido a un hombre o una mujer. Así como este tipo de metodología puede utilizarse para identificar discriminación de género, se ha empleado en una gran variedad de contextos, como son el caso de Nemzer y Neymotin (2020) y Peng y Jiang (2016).

En Nemzer y Neymotin (2020), se utiliza una estructura de *machine learning* para examinar la relación entre las palabras utilizadas en la descripción de las películas de comedia de la base de datos de IMDB y su performance. Es decir, observan qué palabras son utilizadas para describir la película y luego observan cuántas entradas vendió esa película, qué reseña le otorgan los usuarios y qué puntaje tiene la película en la página Metacritic. De esta forma, a partir de las descripciones, buscan predecir cómo le fue a la película. Para esto, llevan a cabo regresiones lineales en conjunto con una red neuronal para analizar los textos. En Peng y Jiang (2016), los autores utilizan el método de palabras incrustadas, en conjunto con una red neuronal para analizar las palabras empleadas en las noticias financieras y así poder predecir movimientos de precios en el mercado de acciones.

Ahora, volcándonos más específicamente a la temática de predicciones en legislación, no podemos dejar afuera el trabajo de Nay (2017). Este autor tiene también el objetivo de predecir si un proyecto será aprobado. Para eso, utiliza un modelo de redes neuronales para construir vectores de palabras que analiza con un complejo modelo de lenguajes y árboles de clasificación para estimar una probabilidad de promulgación. Nuestro aporte en este sentido sería no solo simplificar el método de análisis, sino también permitir que los resultados puedan ser interpretados fácilmente. El *accuracy rate* logrado por el autor es de 96%. Nuestro modelo alcanza uno de más de 83% con una simpleza enormemente superior y la ventaja de poder interpretar resultados de forma sencilla.

III. DATOS

El método propuesto puede ser utilizado para analizar cualquier cuerpo de proyectos de ley. En este trabajo, nos restringiremos a una muestra de 120 proyectos propuestos por senadores argentinos entre marzo y agosto de 2015. De estos proyectos, 60 fueron aprobados al menos en Senado y 60 fueron rechazados, ya sea en comisiones o en cámara (o caducaron). De esta muestra, tomamos un 80% para entrenar el modelo (96 proyectos) y dejamos un 20% para evaluarlo (24 proyectos). Además, tomamos 30 proyectos de ley presentados en junio de 2016 como muestra de evaluación pura (de los cuales 15 fueron aprobados). Esta última muestra permite obtener resultados interpretables en términos de validez externa del modelo ajustado.

Nuestra variable de interés es una binaria que toma el valor de 1 para proyectos aprobados y 0 en caso contrario. Nuestros predictores son la cantidad de veces que aparece cada palabra en el escrito original. El archivo del Senado contiene los textos originales de cada proyecto de ley presentado, así como información sobre si la propuesta fue eventualmente aprobada o si fue descartada (ya sea en comisiones o en la cámara). Con herramientas de los paquetes *pdftools* y *tidyverse* de R, generamos una base de datos que indica cada una de las

Tabla 1 - Palabras más frecuentes en proyectos de 2015

Muestra completa		Proyectos aprobados		Proyectos rechazados	
Palabra	Frec. media	Palabra	Frec. media	Palabra	Frec. media
ley	7.11	año	5.1	ley	10.3
artículo	6.88	artículo	4.27	artículo	9.48
año	5.89	es	4.27	año	6.68
es	5.43	pueblos	4.07	es	6.58
congreso	4.76	congreso	3.95	congreso	5.57
pueblos	4.58	ley	3.92	pueblos	5.08
bicentenario	4.42	bicentenario	3.9	libres	4.97
libres	4.4	libres	3.83	bicentenario	4.93
nacional	4.26	nacional	3.62	nacional	4.9
presente	3.74	mayor	3.5	presente	4.85

palabras que aparecen en cada proyecto de ley y la cantidad de veces que aparecen. En la Tabla 1 podemos ver la frecuencia promedio de las 10 palabras más frecuentes de la muestra completa, de los proyectos aprobados y de los proyectos rechazados en la muestra completa de 2015. La cantidad total de palabras que utilizamos como predictores para el modelo es de 16782. Esto nos introduce claramente en un problema de saturación, que buscaremos resolver por medio de *elastic net*, como se explica a continuación.

IV. METODOLOGÍA

Para identificar las palabras que mayor importancia tienen al momento de predecir si un proyecto de ley será aprobado o no, ajustamos los datos al siguiente modelo *logit*.

$$Pr(A_i \mid W_i) = F\left(\beta_0 + \sum_{k=1}^{p} \beta_k W_{ki}\right)$$

Donde A_i es una variable indicadora que vale 1 si el proyecto de ley i fue finalmente aprobado y 0 en caso contrario, W_{ki} es la cantidad de veces que la palabra k aparece escrita en el proyecto de ley i y $F(z) = \dfrac{e^z}{1+e^z}$ es la función de distribución acumulada de la distribución logística. Para estimar los coeficientes β, utilizamos la función de penalidad sugerida por Hastie y Tay (2020) en el paquete *glmnet* de R. Esta función de penalidad tiene un componente de ajuste dado por la contribución (negativa) al logaritmo de la verosimilitud $l_i(A_i, \eta)$ del proyecto de ley i y un componente de complejidad del tipo *elastic net*, regido por los parámetros α y λ. Cuando $\alpha = 1$, se trata de un problema de *LASSO*, mientras que $\alpha = 0$ es un problema de Ridge.

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} l_i(A_i, F(\beta_0 + \beta^T W_i)) + \lambda\left(\frac{1-\alpha}{2} \sum_{k=1}^{p} \beta_k^2 + \alpha \sum_{k=1}^{p} |\beta_k|\right)$$

Los parámetros α y λ pueden ser estimados por *cross-validation*. De esta forma, el componente de *LASSO* llevará a cabo la selección de palabras relevantes para la predicción de la ley, mientras que el componente de Rigde

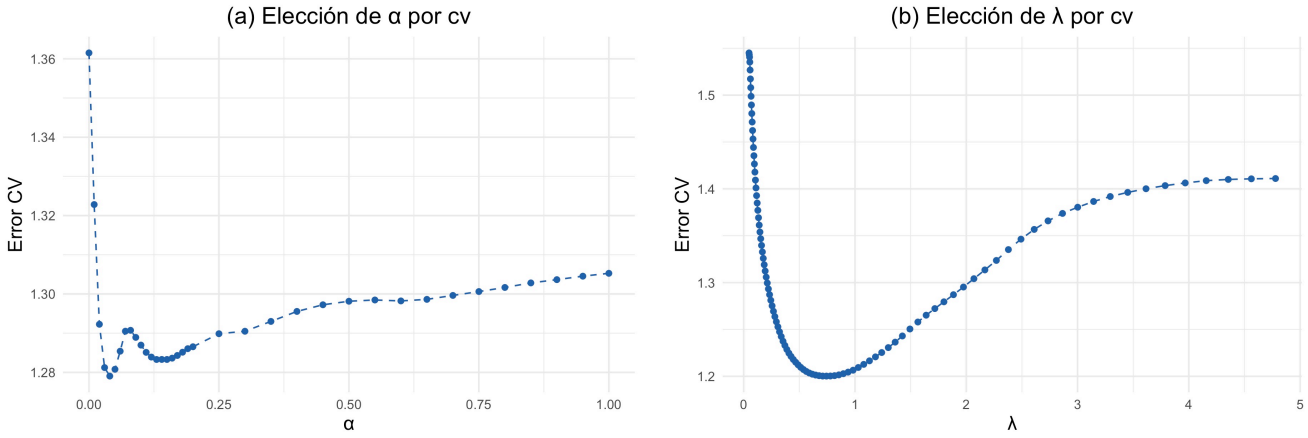

Figura 1. Elección de parámetros por CV

asistirá en esta selección cuando haya palabras que correlacionen mucho entre sí y generen inestabilidad por agrupamiento. Además, estamos en un contexto de saturación, pues la cantidad de palabras excederá la cantidad de proyectos de ley en cualquier caso imaginable. La penalidad de *elastic net* nos permite estimar los parámetros aun cuando tenemos más predictores que observaciones.

La estimación de este modelo permitiría identificar cuáles son las palabras más relevantes para predecir si un proyecto de ley será aprobado o no por un congreso en particular. A continuación veremos las estimaciones para la base presentada en la sección anterior.

V. RESULTADOS PARCIALES Y CHEQUEOS

Al ajustar el modelo a los 96 proyectos de entrenamiento, nuestro primer paso fue estimar los parámetros α y λ por *10 fold cross validation*. En la Figura 1 se puede observar el error por *cross validation* para distintos valores de ambos parámetros.

El mínimo error por *cross validation* obtenido con estos datos surge de $\alpha = 0.04$ y $\lambda = 0.7434$. Esto lleva a que la penalidad de Ridge tenga mucho más peso que la de *LASSO* al momento de ajustar los coeficientes del modelo.

Una vez definidos los valores óptimos de cada parámetro por *cross validation*, podemos identificar las palabras que mejor predicen si un proyecto será aprobado o no según la magnitud de sus coeficientes estimados. En la Tabla 2 puede observarse la lista de las 20 palabras más relevantes con carga positiva y

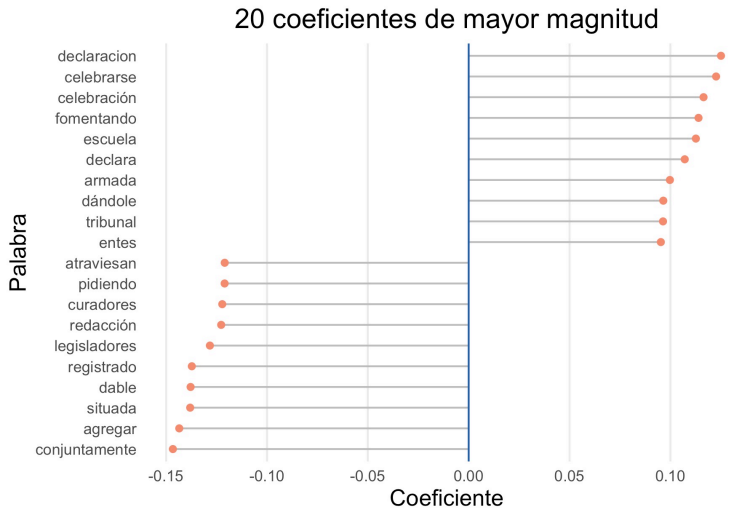

Figura 2. Palabras de mayor relevancia

las 20 con mayor carga negativa. La Figura 2 muestra los coeficientes estimados por *logit* de las 10 palabras positivas y las 10 negativas con coeficientes de mayor magnitud.

Las palabras identificadas no parecerían tener un significado interpretable obvio. Además, no es suficiente quedarnos con esta reducida lista, pues el modelo seleccionó un total de 1074 palabras. Un resultado que sí tiene una interpretación relevante surge de la evaluación del modelo.

Para estimar el poder predictivo del modelo entrenado, primero lo evaluamos con los 24 proyectos de 2015 que reservamos para esto y luego con los

Tabla 2 - Palabras de mayor relevancia predictiva

Efecto positivo		Efecto negativo	
Palabras 1 a 10	Palabras 11 a 20	Palabras 1 a 10	Palabras 11 a 20
declaracion	intenso	conjuntamente	concreta
celebrarse	órdenes	agregar	lleve
celebración	aviación	situada	espíritu
fomentando	talleres	dable	entiendo
escuela	trabajando	registrado	integrar
declara	ubicada	legisladores	2017
armada	disponiendo	redacción	van
dándole	premio	curadores	140
tribunal	precoz	pidiendo	70 %
entes	gubernamental	atraviesan	saludo

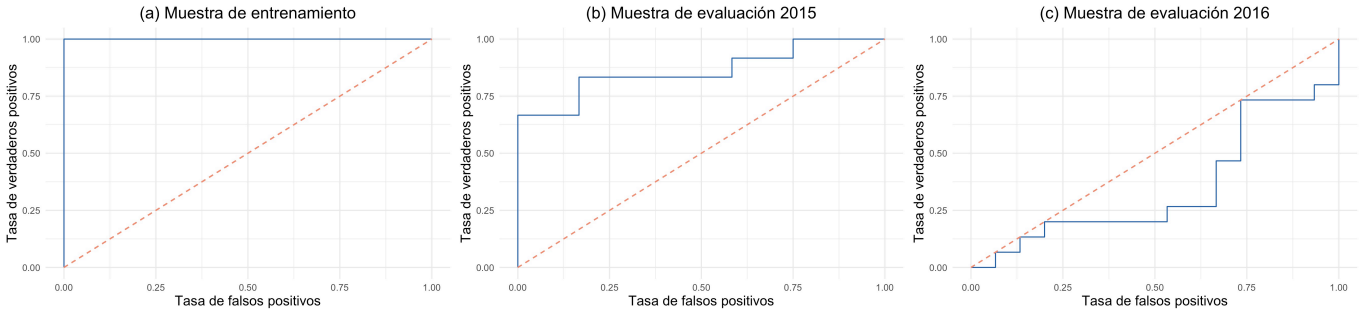

Figura 3. Curvas ROC

30 proyectos de 2016. En la Figura 3 puede observarse la curva ROC para la muestra de entrenamiento (a), la muestra de evaluación de 2015 (b) y la de 2016 (c). Es notorio que, a pesar de ajustar a la perfección a los datos de entrenamiento (lo que sugiere un potencial problema de *overfit*), el modelo también predice relativamente bien en la muestra de evaluación de 2015. Al evaluar el poder predictivo con la muestra de 2016, sin embargo, encontramos que la curva ROC se revierte completamente. Esto implica que lo óptimo sería invertir las clasificaciones. La interpretación de este resultado es de gran relevancia. Entre las fechas de ambas muestras, hubo una elección que revirtió la composición del Senado y el oficialismo pasó a ser oposición. De esta manera, podemos argumentar que, a pesar de no ser interpretable a simple vista, el conjunto de palabras seleccionado por el modelo tiene cierta carga política e ideológica. Por esta razón, al cambiar la ideología mayoritaria en el congreso, se revierte el signo de los coeficientes para las palabras predictoras (al menos en el agregado).

La Figura 4 permite entender mejor el resultado anterior, al representar las distribuciones de las probabilidades estimadas por el modelo para cada muestra. En el panel (a) podemos ver que existen muchos umbrales para los cuales la clasificación es perfecta. En los paneles (b) y (c) las líneas verticales

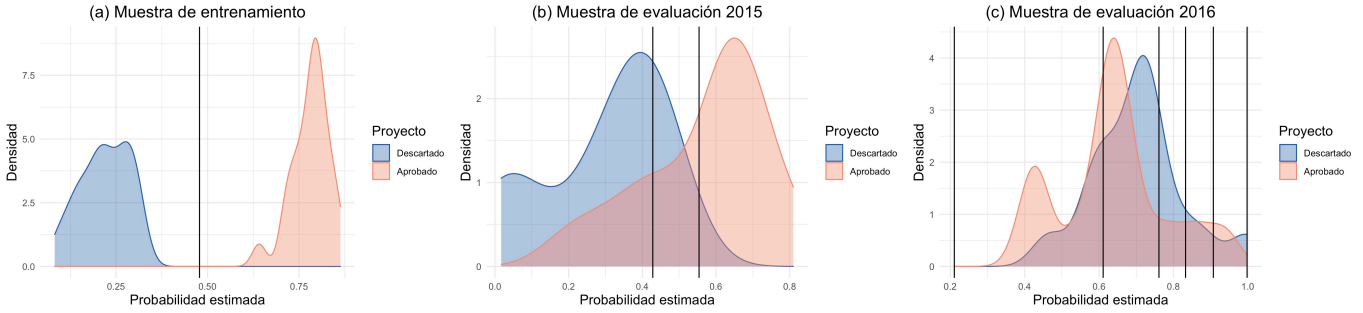

Figura 4. Distribución de probabilidades estimadas

representan los umbrales óptimos que maximizan el *accuracy rate*. Para la muestra de evaluación de 2015, encontramos dos umbrales óptimos. Estos son $p_1 = 0.43$ y $p_2 = 0.55$ y llevan la tasa de precisión a 83.34%. En la muestra de 2016, en cambio, el máximo *accuracy rate* a la que puede aspirarse es del 50%, y se puede alcanzar con 6 umbrales distintos.

V. CONCLUSIONES

Las tasas de proyectos de ley que eventualmente son aprobadas tienden a ser bajas. Al preguntarnos las razones de este fenómeno, comenzamos a pensar en factores externos y del propio proyecto que puedan facilitar o dificultar su llegada a la meta. En este trabajo, presentamos un método simple que permite identificar las palabras más relevantes a la hora de predecir si un proyecto será aprobado o no y que puede ser utilizado para analizar cualquier cuerpo de proyectos de ley, sin importar el país ni el nivel gubernamental que se quiera estudiar.

Nuestros resultados muestran que el método parecería funcionar de forma adecuada siempre que el congreso no cambie su composición. Esto significaría que una vez que el modelo ya fue ajustado con una muestra de entrenamiento, su validez externa para predecir si un proyecto será aprobado o no se limita al intervalo de tiempo en el cual los legisladores que ocupan los escaños se mantienen invariables. Además, los resultados sugieren que las palabras contienen cierta carga ideológica, pues al cambiar el partido que ostenta la mayoría, las predicciones parecerían ir en sentido contrario a la realidad.

Por último, si se quisiera realizar extensiones o réplicas a mayor escala de esta aplicación de *machine learning,* sugeriríamos dos grandes mejoras. Desde la mirada interna, si se tuviera una mayor potencia computacional, podrían agregarse como predictores conjuntos de palabras que aparezcan juntas. Esto no solo incrementaría la flexibilidad del modelo, sino que también podría aumentar su potencial de interpretación, por ejemplo, al permitir que expresiones de más de una palabra tomen relevancia. Por otra parte, desde la mirada externa, sería de gran utilidad añadir predictores como el legislador que presenta el proyecto, el

partido al que pertenece, la temática o comisiones por las que atraviesa el proyecto e incluso variables que contengan información sobre las condiciones generales que se atraviesan en el momento.

Con todos estos agregados, los resultados podrían ser mucho más precisos para predecir. Sin embargo, con nuestros resultados, encontramos que no es necesario un modelo de extremada complejidad para obtener buenas predicciones.

REFERENCIAS

Hastie, T., & Tay, K. (2020). Glm family functions in glmnet 4.0. https://cran.r-project.org/web/packages/glmnet/vignettes/glmnetFamily.pdf

Nay, J. J. (2017). Predicting and understanding law-making with word vectors and an ensemble model. *PLOS one, 12*(5), e0176999.

Nemzer, L. R., & Neymotin, F. (2020). How words matter: Machine learning & movie success. *Applied Economics Letters, 27*(15), 1272-1276.

Peng, Y., & Jiang, H. (2016). Leverage financial news to predict stock price movements using word embeddings and deep neural networks. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 374–379

Wu, A. H. (2018). Gendered language on the economics job market rumors forum. *AEA Papers and Proceedings*, *108*, 175-79.