Ronny Cruz
Catalina Esmeral Flórez
Federico Ramírez
Juan Camilo Sánchez
Marzo 20, 2023
Link al repositorio: Github PS4

Problem Set 4: Predicting Tweets

I. Introducción

El análisis de texto se ha vuelto esencial en la toma de decisiones en muchos ámbitos. El procesamiento de texto implica la extracción de información de fuentes de texto y la transformación de esta información en un formato útil y comprensible para los modelos de texto, los cuales son herramientas de aprendizaje automático que permiten la comprensión de grandes volúmenes de texto y la identificación de patrones y relaciones entre las palabras y los temas.

Una de las fuentes de texto más utilizadas en la actualidad son los tuits, debido a su brevedad y a la gran cantidad de información que se puede obtener a través de ellos. Los tuits contienen información sobre los pensamientos, opiniones y acciones de los usuarios de las redes sociales, y pueden utilizarse para comprender la opinión pública sobre un tema en particular, para identificar tendencias y para predecir eventos futuros. En particular, Twitter surge como una herramienta importante para que figuras públicas como los políticos se comuniquen con sus seguidores y determinen temas de alta discusión.

El objetivo de este ejercicio es predecir el autor de los tuits publicados por tres importantes políticos colombianos: Claudia López, Gustavo Petro y Álvaro Uribe. Esta tarea implica analizar datos textuales y utilizar técnicas de *machine learning* para predecir la fuente de cada tuit. Para ello, se utilizará un conjunto de datos de entrenamiento que consta de tuits de las cuentas de los tres políticos y un conjunto de testeo con tuits sin etiquetar. El objetivo es construir una serie de modelos que logren predecir el autor de cada tuit en el conjunto de testeo.

Dentro de los resultados se encontró que los modelos *Logit* con hiperparámetros y de redes neuronales son los que tienen mejor ajuste y precisión a la hora de predecir el autor de cada tuit, mientras que los modelos de análisis discriminante lineal y de bosques aleatorios tuvieron un bajo poder de predicción en el ejercicio. A continuación, se presenta una sección que describe la obtención y limpieza de los datos, las estadísticas descriptivas que contribuyen a la comprensión del problema y el tratamiento utilizado en las muestras de entrenamiento y testeo; una sección siguiente que describe los modelos estimados y sus resultados, una sección final con conclusiones y recomendaciones que detallan las bondades de los hallazgos y sus principales limitaciones y reflexiones para tener en cuenta en futuros ejercicios.

II. Datos

Previo al análisis de los datos, la limpieza de la base es necesaria para filtrar las palabras de modo que solo queden aquellas palabras clave relacionadas con el modo de hablar de cada político. En una primera fase, se procedió a emplear la función '*stopwords*` sobre el data set de tuits. Esta función contiene un set de palabras que no aportan a nivel semántico a los textos en cuestión, por ejemplo: conectores lógicos, nombres propios, preposiciones, artículos, conjunciones, entre otras.

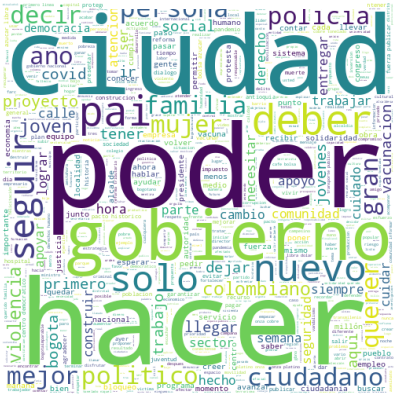
Para el desarrollo de este ejercicio se emplearon los '*stopwords*` del paquete contexto del Departamento Nacional de Planeación (DNP) con el fin de eliminar apellidos y nombres usuales, nombres de municipios y demás palabras recurrentes sin relevancia. Teniendo en cuenta que este paquete fue diseñado para análisis de texto recurrente en los nombres, lenguajes, vocablos, modismos y expresiones más recurrentes en Colombia, se espera que su uso le brinde mayor ajuste a los datos y posteriormente a los modelos estimados.

En una segunda fase de limpieza se removieron hipervínculos, posibles errores de digitación (palabras repetidas), los acentos (tildes, diéresis y virgulillas), los números, además de pasar todas las palabras a minúscula y eliminar todas las palabras que tuvieran menos de cuatro caracteres.

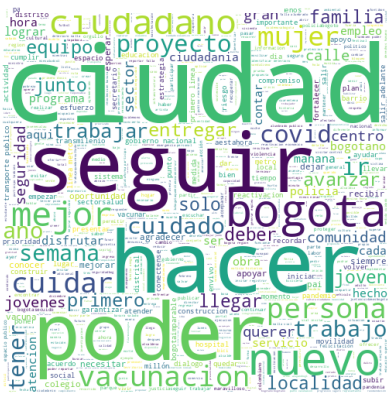
Posteriormente, se aplica un esquema de *Stemming,* el cual reduce las palabras hasta su raíz, por ejemplo: niño, niña, niños, niñitos, se reducen solo a "niñ". Adicionalmente, se aplica la metodología de *lematizador*, que agrupa algunas palabras en una sola como denominador, retomando el ejemplo anterior, estas palabras solo quedarían representadas por "niños". Este proceso de limpieza es fundamental, ya que facilita el proceso de vectorización, puesto que se reducen las dimensiones de la información.

A partir de la limpieza y vectorización de los datos, con la metodología TF-IDF que consiste en asignar un valor numérico a cada palabra para determinar qué tanto se repiten en los tuits, se construyeron las nubes de palabras de la muestra y para cada uno de los políticos. Estas se presentan a continuación:

General



Claudia López



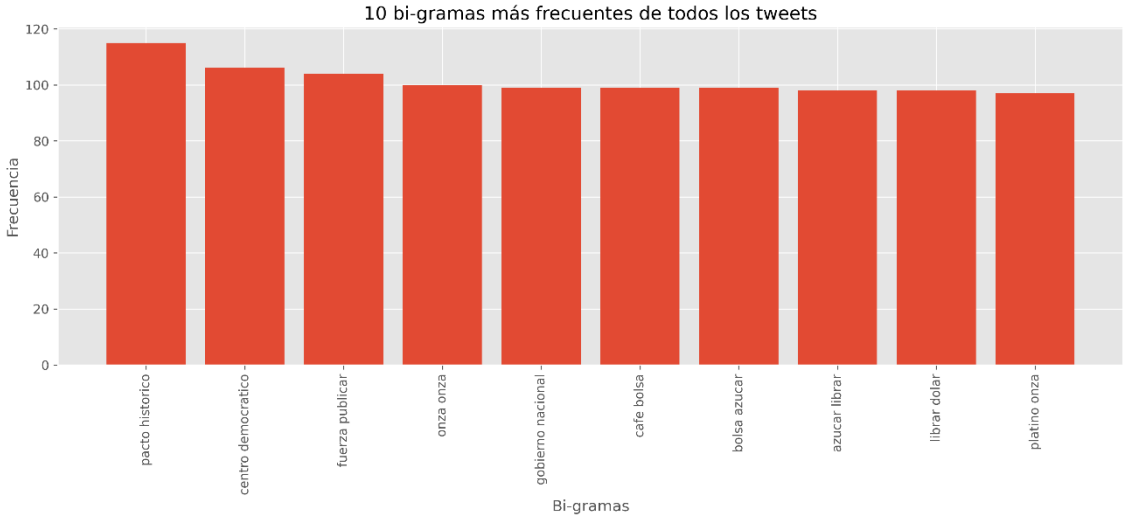
Gustavo Petro

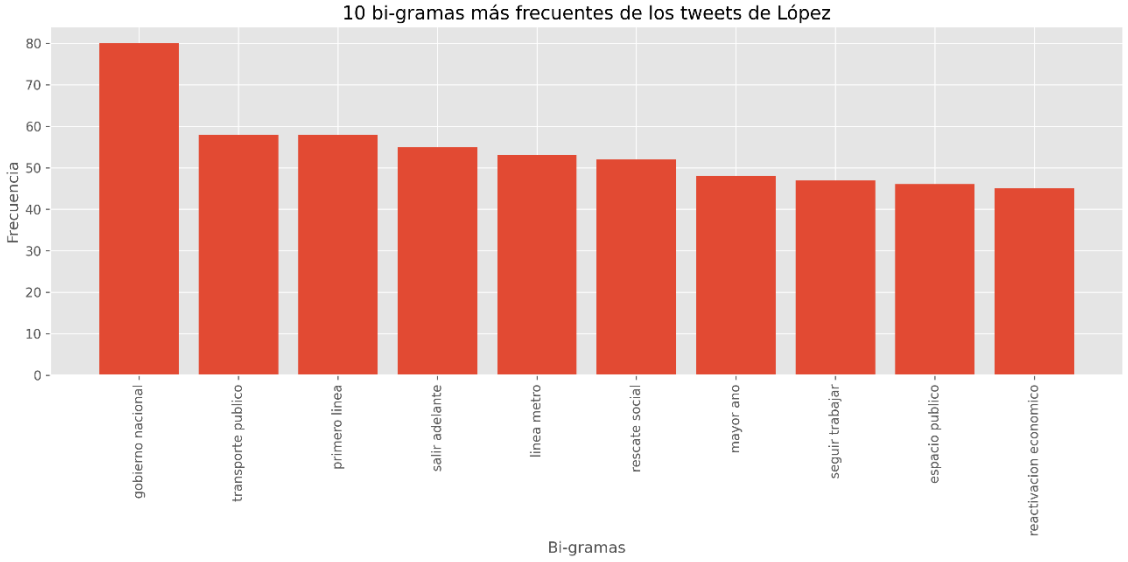
Álvaro Uribe

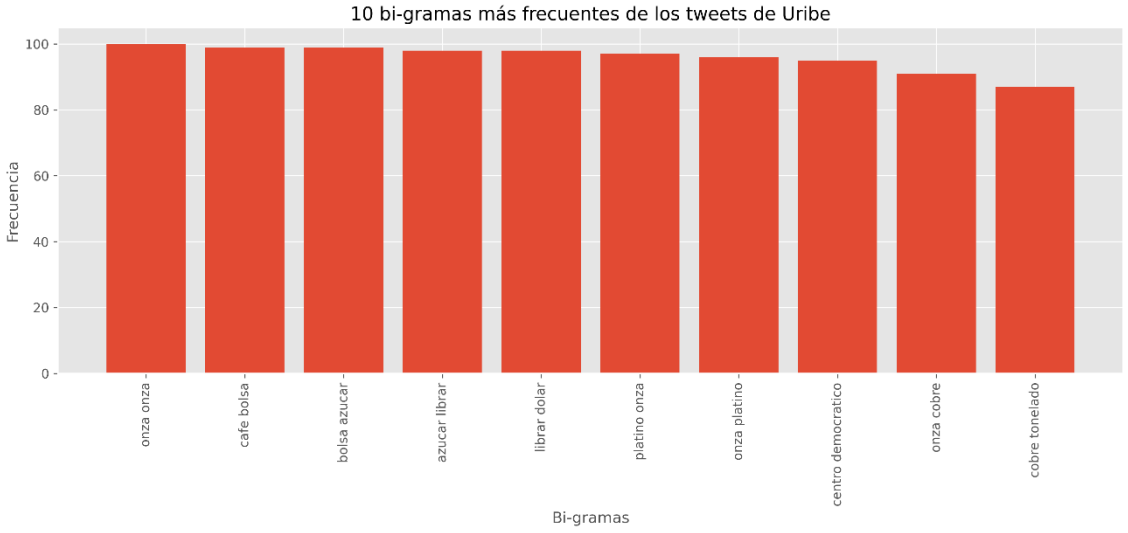
De acuerdo con los gráficos anteriores, se puede observar que las tres palabras más repetidas por todos los políticos en sus tuits son ciudad, poder y hacer. En los tuits de Claudia López también se repite con frecuencia Bogotá, seguir, mujer y nuevo; Gustavo Petro repite gobierno, deber, cambio y solo; Álvaro Uribe repite familia, bloqueo y solidaridad. Este análisis de texto permite relacionar incluso la orientación política de cada uno y los temas más frecuentes en su discurso, asociadas a sus posiciones y funciones políticas y ubicación dentro del espectro político colombiano.

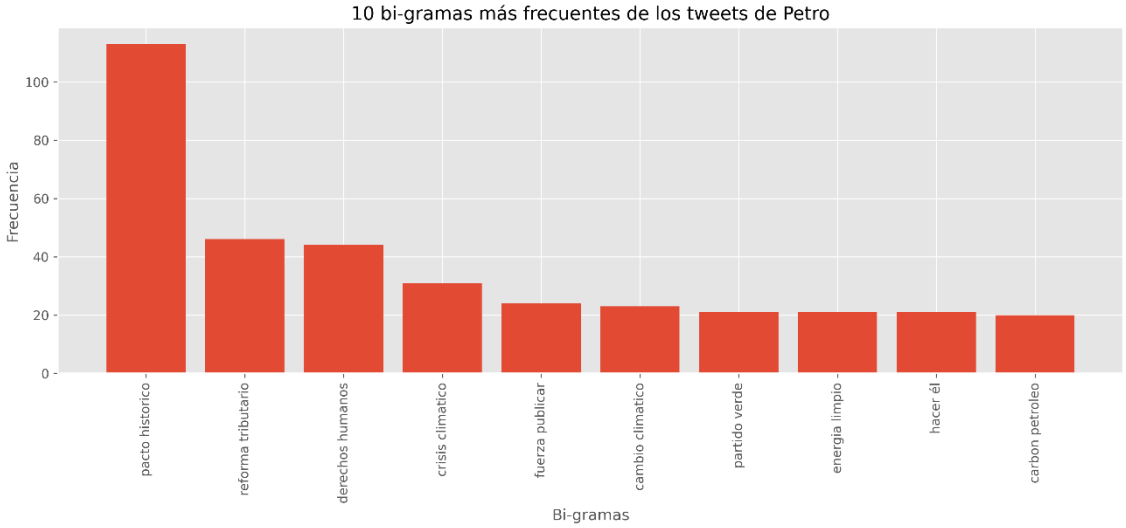
Posteriormente, se construyeron bi-gramas que pudieran relacionar las palabras más frecuentes dentro de los tuits de cada político. A nivel general, se observa que los conjuntos de bi-gramas más frecuentes en los tres políticos son "pacto histórico", "centro democrático" y "fuerza pública", denotando la marcada división de partidos políticos en Colombia en los últimos años, en este caso haciendo referencia al pacto histórico, partido de izquierda y el centro democrático, partido de derecha.

De igual manera, los bi-gramas separados por políticos sugieren que en la época en la que fueron realizados los tuits, Colombia se encontraba en una situación de conflicto o escasez, tal como lo denotan algunos bi-gramas referentes a materias primas, agroinsumos o unidades de medidas como onza, libra, azúcar, platino, entre otros. Adicionalmente, en los bi-gramas de Claudia López se encuentra en alta frecuencia la expresión primera línea, que podría referir a la época del Paro Nacional de 2021, también relacionado con el bi-grama "reforma tributaria" que se encuentra frecuentemente en los tuits de Gustavo Petro, teniendo en cuenta que el proyecto de reforma tributaria de 2021 fue uno de los temas más discutidos dentro del estallido social de ese año.


10 bi-gramas más frecuentes de todos los tweets
Frecuencia
0
20
40
60
80
100
120
pacto historico
centro democratico
fuerza publicar
onza onza
gobierno nacional
cafe bolsa
bolsa azucar
azucar librar
librar dolar
platino onza
Bi-gramas


10 bi-gramas más frecuentes de los tweets de López
Frecuencia
0
10
20
30
40
50
60
70
80
gobierno nacional
transporte publico
primero linea
salir adelante
linea metro
rescate social
mayor ano
seguir trabajar
espacio publico
reactivacion economico
Bi-gramas


10 bi-gramas más frecuentes de los tweets de Uribe
Frecuencia
0
20
40
60
80
100
onza onza
cafe bolsa
bolsa azucar
azucar librar
librar dolar
platino onza
onza platino
centro democratico
onza cobre
cobre tonelado
Bi-gramas



III. Modelos y resultados

Para realizar la predicción del político al que pertenece cada uno de los tuits, se estimaron y probaron modelos *Logit*, Bosques aleatorios (RF), Análisis discriminante lineal (LDA), *Naive Bayes*, Red Neuronal y *XGBoost*. Para ello, a partir de la base de entrenamiento se hizo una partición para tener una base provisional de testeo. Sobre esta base se realiza una primera evaluación para ver el desempeño de los modelos antes de subirlos a la plataforma *Kaggle*.

Contrastando los modelos por la medida de precisión o *accuracy,* se obtuvieron los siguientes resultados:

Modelo	**Accuracy**
Logit	0.81
Redes	0.81
Naive Bayes	0.7833
RF	0.7366
LDA	0.6166

Como se observa, el modelo *Logit* y el modelo de Redes Neuronales obtuvieron el mejor puntaje de precisión con 0.81 para cada uno, seguido de un modelo *Naive Bayes* multinomial, un bosque aleatorio y por último un modelo LDA.

El modelo *XGBoost* no se incluyó dentro de las predicciones subidas a *Kaggle* debido a que la precisión del modelo dentro del conjunto de prueba fue de 0.3219, cuando se tuvieron como parámetros la profundidad máxima de cada árbol de 1500 y *multisoftmax* como función objetivo. Posteriormente se hicieron varios ejercicios cambiando la profundidad máxima de cada árbol pero no se encontró un número claro que pudiera generar mejores resultados que los modelos anteriores. Adicionalmente, al probar este

modelo se sacrificaba tiempo de ejecución y recursos computacionales que se podían invertir en la revisión de otros modelos.

Respecto al modelo de redes neuronales, se construyó un modelo inicial de tres capas en donde la primera capa tuvo 512 neuronas y utilizó la función de activación ReLU. La segunda y tercera capa tuvieron 256 y 3 neuronas, respectivamente, y se utilizó para ambas una función de activación *softmax*. Además, se incluyeron medidas de regularización y optimización para reducir el sobreajuste del modelo. Este modelo se probó y ajustó en varias ocasiones para obtener resultados aceptables en términos de precisión. No obstante, al momento de obtener su puntaje en *Kaggle* no se encontraron diferencias que sugirieran una mejora importante frente a los resultados obtenidos en el modelo *Logit*.

Por esta razón, sumado a las ventajas en sencillez con las que cuenta un modelo *Logit* frente a un modelo de redes neuronales, se decidió correr una búsqueda de hiperparámetros mediante la definición de una grilla para optimizar los resultados del modelo *logit*. Dentro de este ejercicio, se encontraron los siguientes hiperparámetros que optimizaban la precisión del modelo:

Modelo	**Parámetro "c" de regularización**	**Penalidad**	**Algoritmo (solver)**	**Accuracy**
Logit	11.288	L2 (Ridge)	Liblinear	0.8423

De esta manera, se optimizó el modelo *Logit* con un parámetro 'c' de 11.288, que refiere a la regularización y la penalización de los coeficientes, controlando el sobreajuste del modelo, una penalidad L2 del tipo Ridge y un algoritmo *liblinear*. Estos hiperparámetros llevaron a una precisión de 0.8423, que al probar en *Kaggle* se obtuvo el resultado de 0.8466, una notable mejora frente a los otros modelos estimados, siendo este el elegido para la competencia por su puntaje de precisión y su menor dificultad de ejecución frente a los demás modelos.

IV. Conclusiones

A partir de los resultados anteriores, se considera que en términos generales, los modelos propuestos tienen un poder de predicción significativo a la hora de encontrar los autores de los tuits. Que los modelos escogidos para la competición tengan una precisión mayor a 0.5 puede considerarse como un resultado satisfactorio, especialmente con el modelo *Logit* con hiperparámetros que dio un puntaje de precisión cercano a 0.85.

Dentro de este ejercicio se encuentra que la limpieza de datos y sus métodos juegan un papel muy importante dentro de los resultados del modelo; tal vez incluso más importante en estos ejercicios de análisis de texto que las limpiezas que se hacen en datos más convencionales. Por ello, considerando que es un análisis para Colombia, resulta grato contar con

herramientas como aquellas compartidas por el DNP en su paquete contexto, que se ajustan de buena manera a información con características más asociadas al país. Cabe destacar que si se realizara este ejercicio para otro país o espacio social diferente a Colombia, lo más probable es que este paquete no sea el más eficaz para una limpieza de datos adecuada.

En el ámbito computacional, en esta ocasión se probaron herramientas como Google Colab que permitió una mejora en acceso a recursos de hardware y almacenamiento para que el entrenamiento de los modelos fuese mejor y su ejecución más rápida, lo cual habría tenido mayor dificultad si se hubiese ejecutado en un programa local de un computador que se limita a los recursos de este. Adicionalmente, el ejercicio se hizo en Python al ser este el lenguaje en donde están publicados las herramientas del DNP usadas para la limpieza y el paquete 'keras' que se utiliza para ejecutar redes neuronales.