**PREDICIENDO TWEETS**

Problem Set 4 – GRUPO 6

Víctor Dulio Chique

Víctor Iván Sánchez

Natalia Castro Alarcon

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. INTRODUCCIÓN:**

Las opiniones de personalidades políticas, empresariales o que tengan algún tipo de influencia importan. En EEUU por ejemplo, algunos estudios muestran que los mercados financieros tienden a reaccionar sobre los tweets de personalidades políticas, en particular de Donald Trump. [[1]](#footnote-1) Los políticos usan *Tweeter* para defender sus posiciones y precisar sus ideologías. Particularmente aquellos que tienen posiciones contundentes, pueden producir empatía o reacciones adversas de contradictores y por tanto sus posiciones y comentarios pueden identificado, por los patrones inmersos en su lenguaje. Las personas pueden predecir cuál va a ser su punto de vista o no les sorprende que cierta frase pertenezca a un personaje en específico. En ese sentido ¿Pueden los algoritmos de aprendizaje de máquinas predecir correctamente a quién pertenecen los *tweets* que se publican?

En este trabajo buscamos responder a esta pregunta a través de modelos de clasificación que utilizaron los tweets de tres políticos colombianos muy reconocidos: Claudia López, Álvaro Uribe y Gustavo Petro. El objetivo se centró en predecir acertadamente a quién corresponde cada *tweet* identificando el conjunto de palabras que usualmente hacen parte de sus mensajes y buscando el modelo que arrojara la mejor medida de “*accuracy”*, es decir el mayor número de predicciones acertadas sobre el total de predicciones.

**2. DATOS**

Partimos de una base de datos con 9.347 *Tweets* de 3 de los políticos más prominentes de Colombia: Gustavo Petro, Álvaro Uribe y Claudia López. La base solo cuenta con un identificador del tweet, el nombre del usuario que lo realizo y el texto del tweet, de igual manera contamos con una base de datos de test con 500 tweets de cada usuario, esta base es la que debemos predecir.

***2.1 Limpieza de datos***

Dado que tenemos una base de datos que contiene texto crudo, en este caso tweets de políticos, usamos Procesamiento de Lenguaje Natural (NPL), el cual resulta útil para tareas especificas como la estandarización, eliminación de *stop words* o palabras vacías, la tokenizacion y en particular, eliminar del texto todo aquello que no aporte información relevante para la predicción. En las redes sociales, por ejemplo, los usuarios escriben de la forma que quieren, lo que podría resultar en un elevado uso de abreviaturas, signos de puntuación, palabras redundantes, entre otros. Así mismo usamos la técnica de tokenizar los textos, que consiste en dividir el texto en las unidades que lo conforman. En general, el proceso de limpiar y tokenizar el texto consiste, principalmente en convertir el texto en minúscula, eliminar los signos de puntuación, eliminar los números, eliminar espacios en blanco y tokenizar por palabras individuales.

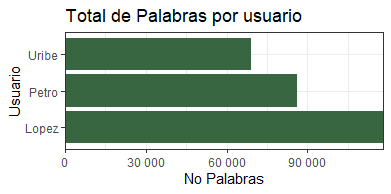
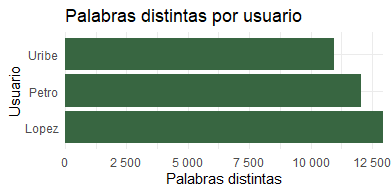
***2.2 Descripción de los datos***

En esta sección realizamos un análisis exploratorio de los datos. La gráfica 1 muestra que Claudia López fue quien más palabras utilizó en los tweets. En el cuadro 1 se puede apreciar que Álvaro Uribe utilizó 41.4% menos palabras que ella y Gustavo Petro 30% menos. De igual manera, Claudia López escribió la mayor cantidad de palabras diferentes pero la diferencia frente a Gustavo Petro y Álvaro Uribe fue solamente de 6.36% y 15% menos respectivamente. En general podemos decir que Claudia López habla sobre una mayor variedad de temas y/o utiliza mayor número de palabras para expresar una idea o informar sobre un acontecimiento. Álvaro Uribe pareciera centrase en ciertos contenidos y los escribe de forma más concisa y Gustavo Petro se encuentra en un punto medio.

|  |  |  |
| --- | --- | --- |
| Usuario | Total palabras utilizadas en tweets | Palabras diferentes |
| Claudia López | 118.104 | 12.876 |
| Gustavo Petro | 86.218 | 12.056 |
| Álvaro Uribe | 69.175 | 10.942 |

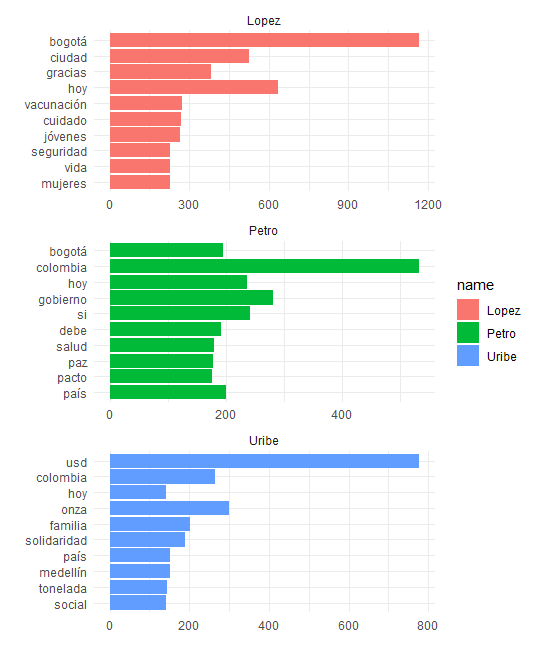
Cuadro 1: Total palabras utilizadas y palabras diferentes

En general podemos decir que Claudia López habla sobre una mayor variedad de temas y/o utiliza mayor número de palabras para expresar una idea o informar sobre un acontecimiento. Álvaro Uribe pareciera centrase en ciertos contenidos y los escribe de forma más concisa y Gustavo Petro se encuentra en un punto medio.

**

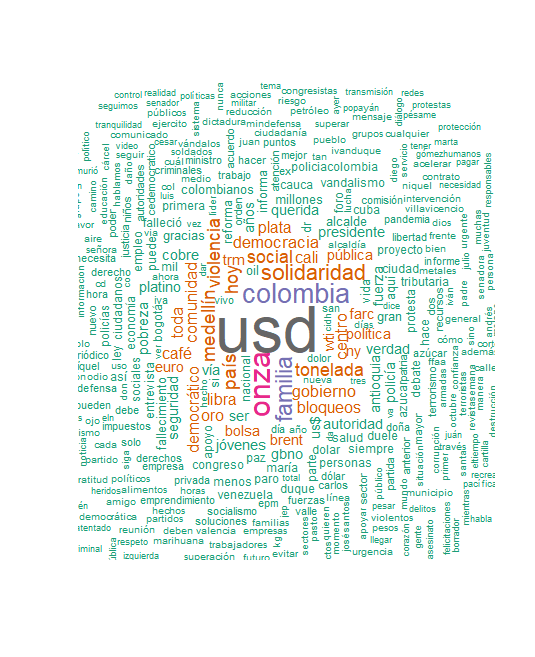
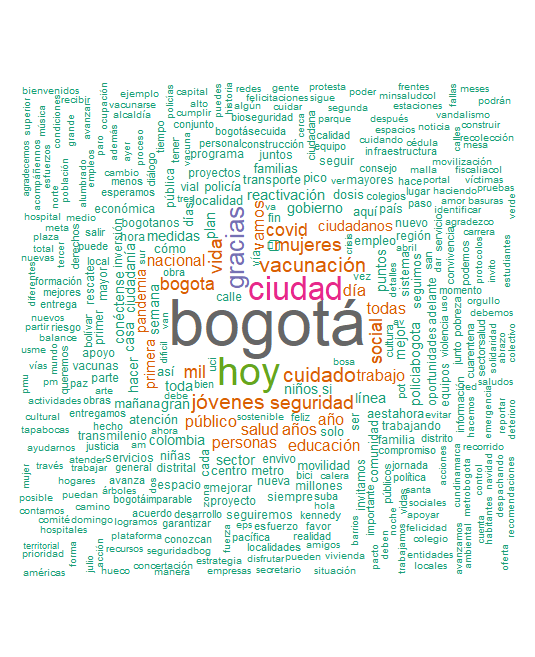
*Gráfica 1 Gráfico 2*

La gráfica 3 muestra cuáles son las palabras que con mayor frecuencia utilizan López, Uribe y Petro. La palabra más utilizada por Claudia López es “Bogotá”, la de Gustavo Petro es “Colombia” y la de Álvaro Uribe “USD”. En general López se centra en lo relacionado con la ciudad, las mujeres, el reconocimiento a quienes ayudan a logra un objetivo, la vacunación (los tweets se tomaron durante el periodo final de la pandemia), los jóvenes y la seguridad. Gustavo Petro se refiere a temas que tienen que ver con el país, el manejo del gobierno, la salud, la paz y hace frecuentemente mención de su movimiento: Pacto Histórico. Finalmente, Álvaro Uribe escribe sobre el dólar lo cual pareciera mostrar especial preocupación por el tema de la devaluación del peso colombiano, menciona frecuentemente a Colombia y la palabra país, y opina sobre la familia, la solidaridad y la ciudad de Medellín.

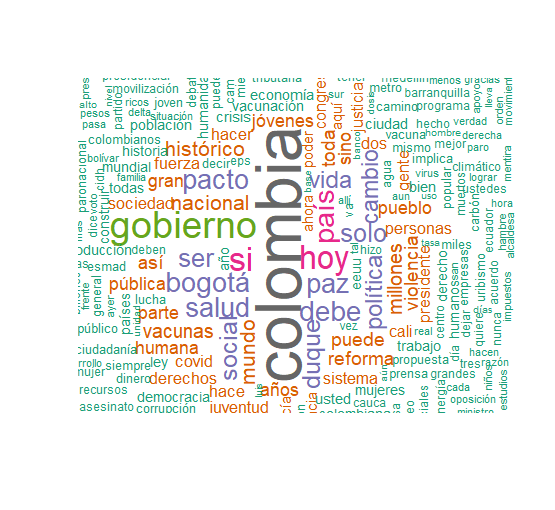


Gráfica 3: Palabras más frecuentes después de Stopwords

En general, el gráfico anterior nos muestra las palabras relevantes usadas por cada político y es una aproximación interesante para conocer las temáticas y patrones que abordan cada uno de ellos, como vemos, cada usuario es diferente en su contenido, en la forma de expresarse y los temas que aborda, por lo que un algoritmo, podría identificar lo anterior y aprender mediante la información suministrada con la finalidad de arrojar predicciones lo suficientemente precisas. A continuación ilustramos la frecuencia de las palabras mediante los siguientes wordclouds:



*Imagen 1. Wordcloud* ***Claudia Lopez*** *Imagen 2. Wordcloud* ***Alvaro Uribe***



*Imagen 3. Wordcloud* ***Gustavo Petro***

**3. Modelos y Resultados**

***3.1 Random Forest***

Los árboles de clasificación buscan predecir la categoría a la que pertenecen las observaciones a través de particiones recursivas binarias que separan en regiones los diferentes predictores que se eligieron. *Random Forest* predice a través de varios árboles de decisión que se construyen a partir de un subconjunto aleatorio de un número determinado de variables predictoras. El objetivo de utilizar *Random Forest* no solamente era lograr predecir las categorías con un *accuracy* alto sino buscar una metodología que nos ayudara a confirmar el número de predictores aleatorios óptimo para otros modelos.

Text

Description automatically generated

En el ejercicio se utilizaron dos conjuntos de predictores. El primero capturó las 50 palabras más utilizadas de los tweets de los tres políticos (después de realizar la limpieza comentada al comienzo del documento) y el segundo las 2644 predictores. Sólo el modelo con 50 predictores logró terminar de correr pero arrojó una predicción muy baja. El modelo con 2644 predictores después de 7 horas continuaba corriendo. Se decidió eliminarlo debido a que el modelo de redes neuronales tomaba menos tiempo computacional y arrojaba resultados muy superiores. La mayor accuracy se obtuvo con 40 predictores aleatorios.

***3.2 Modelo SVM***

Usamos el método de clasificación-regresión Maquinas de Vector Soporte (Vector Support Machines, SVMs), que si bien se ha usado comúnmente para clasificaciones binarias, su aplicación se ha extendido a problemas de clasificación múltiple, para la creación de este algoritmo seguimos los siguientes pasos:

1. Tokenizamos los documentos de entrenamiento y testeo, de acuerdo a los procedimientos descritos en la sección de Datos. Creamos las matriz documento-termino y eliminamos términos que aparecen menos de 5 veces.
2. Convertimos los valores de la matriz de termino a tf-idf para los datos de entrenamiento y testeo
3. Con lo anterior, tenemos las matrices tf-idf necesarias para realizar la predicción.

El modelo nos arrojó un *accuracy* de 82.33% en Kaggle, por tanto, tuvo una buena performance, no obstante con la finalidad de mejorar el *accuracy* del modelo se procedió a realizar una optimización de hiperparametros.

***3.2.1 Optimización de hiperparametros***

El método de SVM cuenta con el hiperparametro **“C”** que establece la penalización por clasificación incorrecta regulando el balance entre bias y varianza, dado que es un hiperparametro su valor óptimo se debe estimar haciendo uso de *validación cruzada.*

|  |  |
| --- | --- |
| Cuadro 6: validación cruzada para estimar C | Grafica 1: Optimización hiper parámetro |

En el cuadro 6, se observa que el proceso de validación cruzada nos indica un valor optimo para el parámetro de C = 0.1 en el cual se encuentra un error de 0.172. La grafica 7, representa visualmente la optimización del hiperparametro.

Procedemos entonces a reajustar el modelo de acuerdo al valor óptimo del hiperparametro C =0.1 y logramos un mejor ajuste del modelo con un accuracy de 85% en Kaggle, convirtiéndose en nuestra mejor predicción y en una muy buena performance del modelo.

* 1. ***Redes Neuronales***

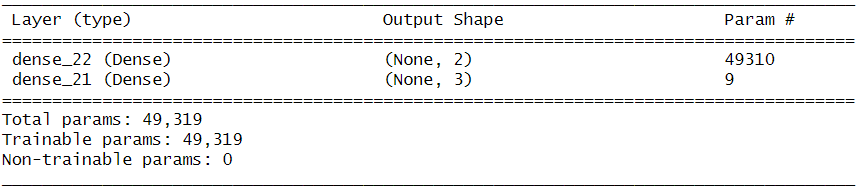
El primer paso en el preprocesamiento de los datos fue la inspección del contenido de las bases, no encontrándose observaciones faltantes. Luego se procedió a la limpieza del texto, tokenización, lematización y la construcción de la matriz de la matriz TF-IDF. Durante este proceso en la matriz de la base train se perdió más de 60 observaciones, y en la dase test se eliminan por encima de 10 observaciones, con lo cual es imposible cargar la predicción en el kaggle (1500 observaciones). Adicionalmente, las dimensiones de las matrices TF-IDF fueron considerablemente distintas en ambas bases. Por lo cual, este procedimiento resultó inviable.

Luego se procedió a identificar la razón por la que se eliminan las observaciones en la base test, encontrándose que un tweet contiene solamente emojis, lo que genera la eliminación de observaciones. En consecuencia, se procedió a imputar los emojis, en la base test, por la palabra más frecuente en la base de datos test, en este caso “bogota”. Además, se encontró que lematizando se eliminaba observaciones en la base test.

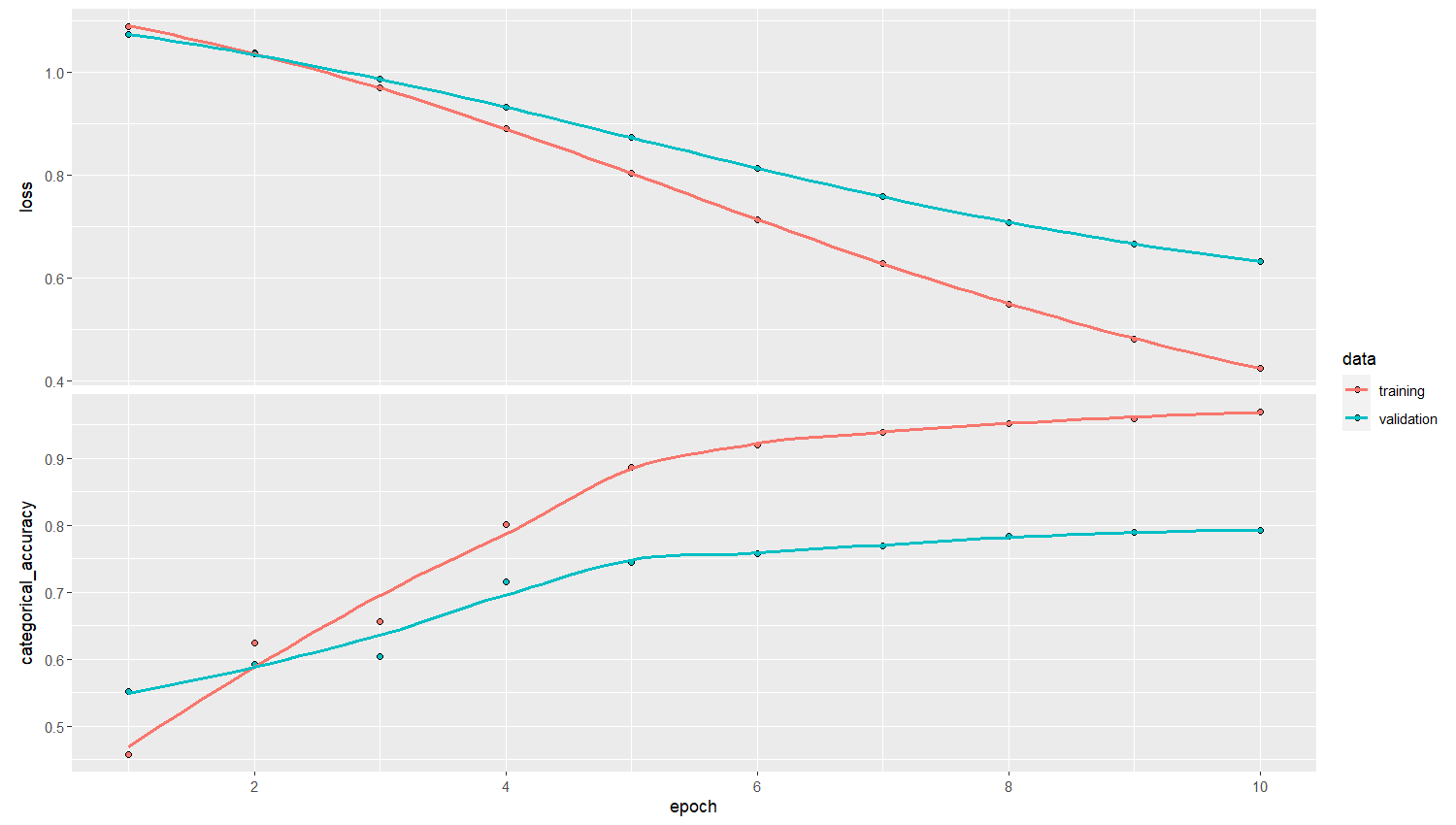
Por lo tanto, para solucionar los problemas señalados, se optó por unir las dos bases para obtener una matriz TF-IDF con igual dimensión en las columnas y no se lematizo para evitar la pérdida de observaciones en la base test.

El mejor modelo de redes neuronales consiguió un score en kaggle de 0.81, pero no supera a la predicción del modelo Support Vector Machines (SVM).

La arquitectura del modelo de red neuronal se construye con una matriz de 9343 observaciones con 24654 columnas, la primera capa corresponde al total de columnas de la matriz, la segunda capa es una capa densa con 2 neuronas y una función de activación “relu”, y la capa de salida corresponde a tres categorías (autores de los tweets: Lopez, Uribe y Petro).



*Cuadro : Parámetros del modelo*



*Gráfica : Función de pérdida y accuracy*

El otro modelo de redes neuronales obtuvo un score de 0.78 en kaggle, considera dos capas intermedias, la primer con 10 neuronas y la segunda con 2 neuronas.

**3. Conclusiones y Recomendaciones**

La predicción multi categórica utilizando palabras provenientes de textos es compleja y requiere utilizar modelos no lineales que capten interacciones. Dentro de estos modelos se encuentran *Random Forest*, el Vector de Soporte de Máquinas (SVM) y las redes neuronales.

Encontramos que *Random Forest* no predice bien en esta clase de problema probablemente porque cuando toma subconjuntos de predictores, no mantiene el más significativo en todos los árboles y justamente las palabras más importantes deben mantener su mayor peso en este tipo de predicción. Adicionalmente requiere gran tiempo computacional.

El Vector de Soporte de Máquinas es reconocido por ser efectivo en espacios *high dimensional* como el de este estudio que para este ejercicio utilizó 158844 palabras. Para este modelo en específico se utilizaron . Adicionalmente es una metodología que busca maximizar el margen entre la línea separadora de clases y los vectores de soporte. Encontramos que separar las clases por distancia utilizando los hiper planos es más efectivo y preciso que separarlas por condiciones por nodos.

Las Redes Neuronales por su parte permiten utilizar funciones de activación como ReLU que son computacionalmente muy eficientes. De esta manera cuando se tiene un alto número de predictores resultan muy atractivas. Sin embargo, necesita un gran trabajo de *tuning* de hiper parámetros para obtener la mejor predicción. Para los tres modelos es fundamental el trabajo de limpieza de datos.

**4. Bibliografía**

Llorente, Juan. (2020). El impacto de los tweets de Donald Trump sobre los mercados financieros.

1. Llorente, Juan. (2020). El impacto de los tweets de Donald Trump sobre los mercados financieros. [↑](#footnote-ref-1)