2 DE ABRIL DE 2023

Departamento de Ingeniería de Sistemas y Computación
ISIS 3301 – Inteligencia de Negocios
Proyecto Analítica de textos

Estudiantes:

Julian Andres Mendez Melo	201920623	j.mendezm@uniandes.edu.co
María Alejandra Vargas Torres	201123148	ma.vargas73@uniandes.edu.co
Omar Esteban Vargas Salamanca	201921271	o.vargas@uniandes.edu.co

Tabla de contenido

1. Caso de estudio

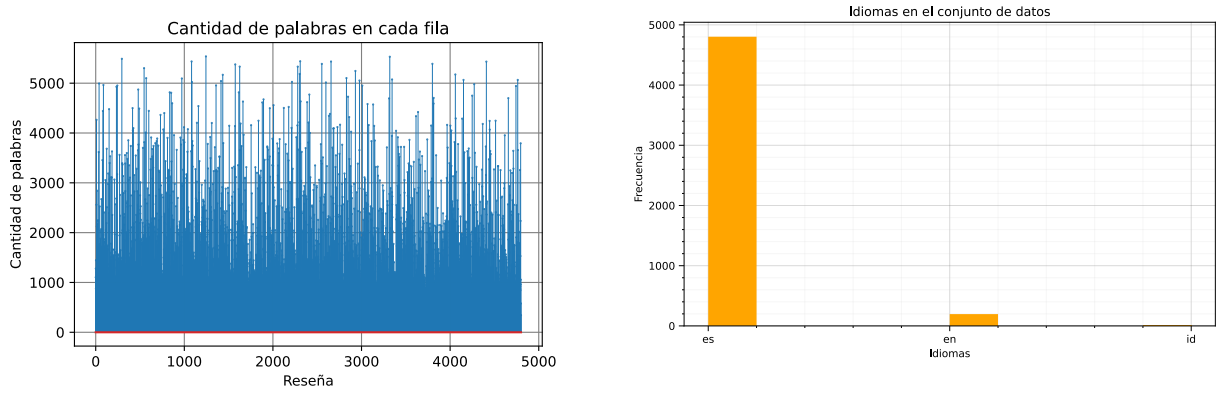
Oportunidad | Problema de negocio	El negocio quiere identificar si la extensión del texto influye en la evaluación y las palabras significativas de cada categoría. **Objetivo General** Comprender el problema y utilizar los algoritmos seleccionados (Regresión Logística, MultinomialNB, Red neuronal) para así poder dar respuesta a si la reseña se ve afectada por el número de caracteres e identificar el género de la película. **Objetivos específicos** 1. Entender el conjunto de datos dado por el negocio. 2. Realizar de forma apropiada la preparación y transformación de los datos para asegurar la calidad de los datos. 3. Identificar los algoritmos que pueden ser utilizados para resolver la problemática de la clasificación de las películas según las reseñas. 4. Implementar los algoritmos propuestos de forma correcta.
Enfoque Analítico	Identificar si la extensión del texto influye en la clasificación de la reseña y encontrar las palabras significativas cada categoría.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Organización: Productoras de cine. Rol: Inversionistas, director, guionistas.
Técnicas y algoritmos a utilizar	- Regresión Logística - Red neuronal - MultinomialDB

2. Entendimiento y preparación de los datos

2.1. Perfilamiento y Análisis

En la carga de datos se encontró que el conjunto de datos tiene tres columnas: el identificador de la reseña, la reseña y el sentimiento. Debido a que se quiere identificar según el contenido qué sentimiento se genera se eliminó la primera columna ya que no es relevante para el análisis. Seguido de lo anterior utilizando `.shape` en el Data Frame se obtuvo que se tienen 5000 filas. El siguiente paso fue utilizar estadísticas descriptivas para conocer de mejor forma el conjunto de datos en los cuales se obtuvo el máximo y

mínimo de palabras junto con el lenguaje en el que estaba escrito, para este último se utilizó la librería `langdetect`. La figura 1 muestra la cantidad de palabras en cada fila y la distribución de los lenguajes que se encontraban en el conjunto de datos.



a) Máximo de palabras **b) Idiomas**

Figura 1. Distribución máximo de palabras y de los idiomas en el conjunto de datos

En la anterior figura se observa que la mayoría de los datos se encuentra en español, también para la mayoría de las filas se tiene que hay más de 1000 palabras. Teniendo en cuenta lo especificado en el diccionario solo se deben dejar las columnas que se encuentran en español. Por otra parte, para evaluar la calidad de los datos se generó un reporte utilizando la herramienta de `pandas_profiling`, en la que se encontró que había dos duplicados, de estas solo se dejó la primera ocurrencia, la completitud de los datos es del 100%, en cuanto a la consistencia se evidenció un problema con el contenido en los idiomas y en la validez no se debieron realizar correcciones.

2.2. Preparación y transformaciones

Se definieron las funciones de la tabla 1 para preparar los datos después la tokenización:

Tabla 1. Funciones utilizadas después de la tokenización

Función	Definición
`remove_non_ascii`	Remover los caracteres que no sean ASCII
`to_lowercase`	Convertir cada letra de la palabra a minúscula
`remove_punctuation`	Remover la puntuación
`replace_numbers`	Responder los números por su equivalente textual
`remove_stopwords`	Remover las palabras vacías (*stop words*)
`Remove_blank_space`	Remueve los espacios en blanco luego de realizar la tokenización

Para cada una dentro de su respectiva función se creó una lista dentro de la cual a medida que se le pasaba cada palabra realizaba la operación respectiva a la columna `review_es`, generando como resultado la columna `palabras`. Ya con las palabras tokenizadas se encontró la moda, siendo esta la palabra que más se repite del conjunto que se tienen dentro de la reseña. El siguiente paso fue normalizar los datos, es decir, retirar los prefijos y sufijos de cada palabra para después realizar una lematización, para

esto se utilizó la librería spacy, con esta para que funcionara de forma correcta se debieron unir los datos anteriormente transformados para que después realizara la lematización, es necesario recalcar que en la documentación se tiene que hay un 84% de probabilidad de que funcione adecuadamente, por lo cual se evidencia que en algunos valores se tienen palabras como película y peliculo.

Para utilizar los algoritmos y poder predecir la variable objetivo que en este caso es el sentimiento se deben transformar las columnas a valores numéricos. En la columna *sentimiento* si era positivo se transformó a 0 y negativo a 1, para las reseñas se utilizó la librería de sklearn `feature_extraction.text` con la función `TfidfVectorizer`[1], este está une dos funciones de sklearn, `CountVectorizer` y `TfidfTransformer`, la primera toma el texto y realiza una tokenización en la cual se tiene en cuenta la frecuencia con la que aparece cada variable, esto por cada fila y el segundo realiza una transformación teniendo en cuenta la frecuencia de las ocurrencias.

```
['hola cómo estar',
'caer trabajar hola']
```

$\rightarrow Se\ representa\ como \rightarrow$

```
[[0, 1, 1, 1, 0],
 [1, 0, 0, 1, 1]]
['caer', 'cómo', 'estar',
 'hola', 'trabajar']
```

3. Modelado y Evaluación

Con la anterior preparación de datos se procedió a seleccionar los algoritmos y encontrar el que tenga el mejor score, es decir el que tenga una mejor precisión en la predicción con los datos dados. Para la selección se tuvo en cuenta que debía ser supervisado ya que en el conjunto de datos se tienen datos etiquetados, y ya que se quiere predecir un atributo categórico se utilizaron algoritmos de clasificación.

Algoritmo 1: Regresión logística [Omar Vargas]

un modelo de aprendizaje supervisado utilizado para la clasificación binaria, es decir, para predecir si una muestra pertenece a una de 2 instancias posibles [1]. En el caso de clasificación de películas como buenas o malas a partir de reviews.

Los modelos de regresión logística tienen como objetivo:

Establecer por medio de coeficientes de probabilidad la relación entre covariables y la instancia objetivo.

Determinar si existe interacción y confusión entre las covariables en relación a la variable dependiente (es decir, los odds ratio correspondientes a cada covariable) para una correcta interpretación del modelo.

El algoritmo de regresión logística puede ser una buena opción para la clasificación de películas como buenas o malas a partir de reviews debido a sus ventajas en términos de simplicidad, interpretabilidad y capacidad para modelar relaciones no lineales y la cantidad de datos. No obstante, puede tener problemas cuando se usan palabras en

[1] Dependiendo de el algoritmo se le especifica que la transformación sea binaría o no.

sentido irónico, oximoron,doble sentido o sarcásticas. Por ello entre mayor número de palabras vectorizadas el algoritmo puede mejorar, pero también a sobre ajustarse. Para este modelo en particular se relacionaron las palabras que más impacto tienen sobre los coeficientes `lr_model.coef_[0]` y relacionándolas con el orden predefinido del vectorizador de palabras para obtener el top 20 de palabras que más tienen impacto en la clasificación de las reviews ya sea con clasificación positiva o negativa.

En general, cuanto más negativo sea el coeficiente de una variable independiente en la regresión logística, más fuerte será su efecto en la probabilidad del evento. Es decir, en que la review sea positiva.

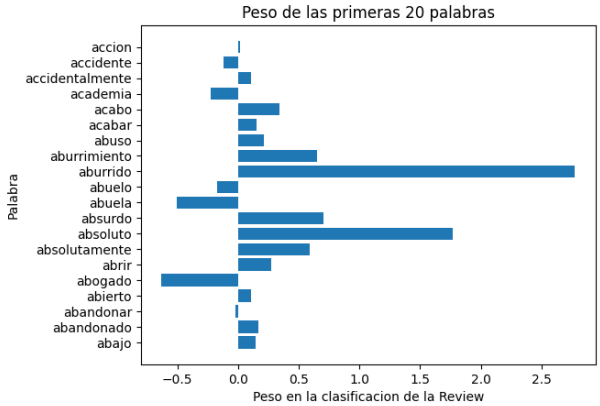


Figura 1. Primeras 20 muestras ponderadas

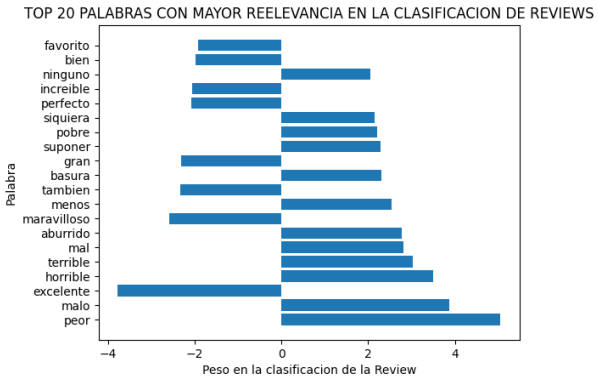


Figura 2. Palabras con máEn s impacto en la clasificación

Algoritmo 2: Red Neuronal [Julian Mendez]

Una red neuronal se encuentra en el área de aprendizaje profundo de machine learning. Esta se caracteriza por utilizar nodos en una estructura de capas que se asimila a el cerebro humano. Ahora la bien, la razón de ser escogida recae en que por medio de una red neuronal es posible realizar procesamiento de lenguaje natural y, por tanto, permite realizar análisis de sentimientos a partir de los datos y documentos de texto. Gracias a esto, es posible diseñar una red neuronal tal que permita clasificar comentarios de positivo y negativo de comentarios de pelicular que permitan obtener resultados claros para el negocio.

La red neuronal fue implementada con Keras, la cual es una librería de alto nivel de python que trabaja de la mano con tensorflow para la creación de las capas de red que hacen parte de una red neuronal. Con esto en mente, la red creada consta de un modelo Sequential, que se refiere a una serie de capas de neuronas secuenciales, en donde hay en total 5 capas y sus parámetros fueron hallados por el criterio de *GridSearchCV().* Para empezar, la primera capa oculta presenta 4 neuronas con función de activación ReLu, seguida de esta viene otra capa de dropout de 0.2 para la regularización. Cabe añadir que la dimensión de entrada de los datos se especifica en la primera capa, la cual equivale a la forma de la matriz de características de entrada. Luego se añade otra capa adicional con el fin de mejorar el rendimiento del modelo con 8 neuronas con función de

activación ReLu donde le sigue otra capa oculta con un dropout de 0.2 para la regularización. Finalmente, se tiene una capa de salida con únicamente 1 neurona con función de activación sigmoide para la clasificación binaria, además un optimizador de adam y una función de perdida utilizada que es la entropía cruzada. Para finalizar, es importante mencionar que se usó la función de activación ReLu porque es una de las funciones más populares y utilizadas en redes neuronales a causa de su simplicidad y eficacia en el modelo, además es fácil de calcular porque no requiere funciones exponenciales y, como consecuencia, sirve para aprendizaje profundo en grandes conjuntos de datos.

Por un lado, en el análisis cualitativo se obtuvo el número de características positivas por cada clase, así mismo se obtuvo el número de características negativas por cada clase. A continuación, en la figura 3 se presenta la evidencia de las primeras 5 filas del número de características positivas y negativas de dichos resultados.

	Negativos
hamlet	111.348155
dino	59.162347
larry	56.180002
mirada	56.119827
kung	55.364856

a) Reseñas negativas

	Positivos
hamlet	101.102055
fabrica	59.027667
mirada	55.592850
dino	51.759832
kung	48.481445

b) Reseñas positivas

Figura 3. Primeras 5 muestras

Por otro lado, en el análisis cuantitativo, se realizó una extracción de información del modelo en términos de cuánto peso le asigna a cada palabra al momento de realizar el entrenamiento. Es importante destacar que el peso es extraído de la última capa de la red neuronal (capa 5) y que las palabras se extraen directamente del objeto *TfiddVectorizer().* En este sentido, si el peso es negativo representa una variable relacionada con el sentimiento negativo y, por el contrario, si el peso es positivo, entonces representa una variable con el sentimiento positivo. En la figura 4 se presenta las 5 palabras con más incidencia al sentimiento positivo y negativo tenidas en cuenta en el entramiento.

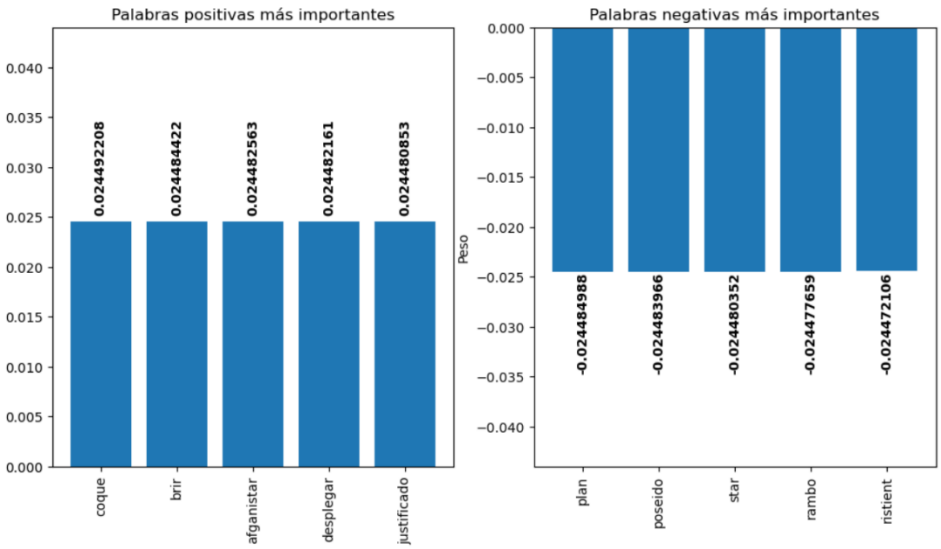


Figura 4. Pesos de características incidentes en el entrenamiento de la red neuronal

Cabe añadir que los resultados dependen del conjunto de datos extraídos de la vectorización de palabras. Por lo tanto, pueden variar si se cambia el número de características.

Algoritmo 3: MultinomialNB [Alejandra Vargas]

El modelo pertenece a los modelos bayesianos, sus predicciones se realizan basadas en las probabilidades. Para este se asume una distribución multinomial, teniendo que cada uno de sus resultados son independientes y la suma de las probabilidades es igual a uno [2]. Este tiene los siguientes pasos:

- Se calcula la probabilidad de cada una de las clases, en este caso si es positiva o negativa la reseña.
- Se mira la probabilidad de que se de cada palabra dentro de la clase.
- Para nuevos datos se evalúa si la clase es posible, calculando la probabilidad de que los nuevos datos, las palabras, pertenezcan a la clase.

En el caso de estudio se realiza una tabla en la que se tienen los atributos que se quieren predecir y en las demás columnas son las palabras, los valores de cada una es cuántas veces han aparecido para cada una:

Reseña	Palabra 1	Palabra 2	Palabra …	Palabra n
Positiva	$\#_1$	$\#_3$	…	$\#_n$
Negativa	$\#_2$	$\#_4$	…	$\#_n$

Para el cálculo de probabilidad de cada una de las palabras en cada categoría se utiliza la siguiente ecuación:

$$P(Palabra\ n|cateoría) = \frac{\#_n}{\#_1 + \#_3 + \cdots + \#_n | \#_2 + \#_4 + \cdots + \#_n}$$

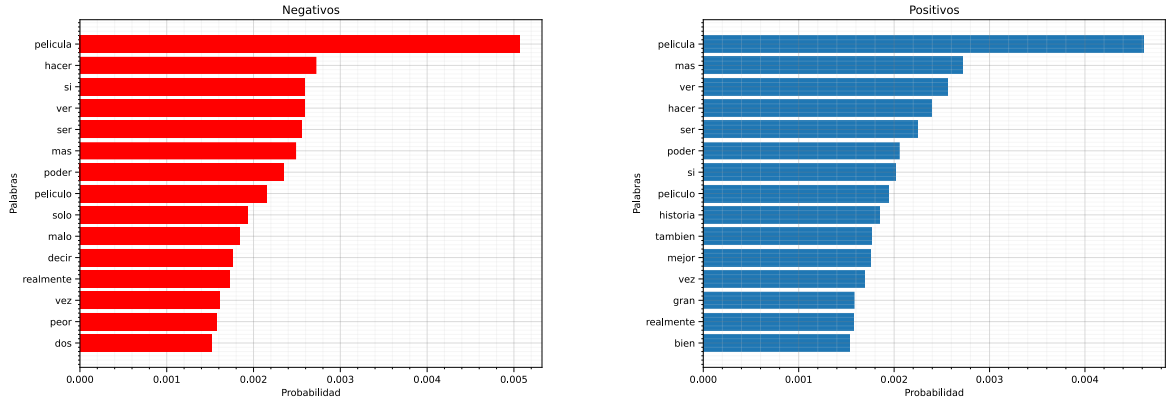
Otra operación que se realiza es calcular la probabilidad de cada categoría, teniendo así que:

$$P(Categoria) = \frac{\sum Datos\ de\ las\ palabras\ de\ la\ cateria}{\sum Datos\ de\ todas\ las\ palabras} = \frac{\#_1 + \#_3 + \cdots + \#_n}{\#_1 + \#_3 + \cdots + \#_n + \#_2 + \#_4 + \cdots + \#_n}$$

Con las operaciones anteriores se calcula el producto de las probabilidades con el cual se escoge la que sea mayor:

$$P(Categoria) \times P(Palabra\ 1_{\text{cat}}) \times P(Palabra\ 2_{\text{cat}}) \times \ldots \times P(Palabra\ n_{cat}) = rta$$

Para este modelo se obtuvieron las gráficas 5 y 6, en la primera se encontraron las probabilidades de ocurrencia de cada una de las palabras, estos corresponden al eje x, y para obtener los datos del eje y que serían los nombres de cada una se utilizó `get_feature_names_out()` en la variable en la que se define `TfidfVectorizer()`. Para la segunda gráfica se utilizó `.feature_count_` la cual toma el número de muestras encontradas después de realizado el ajuste para cada una de las variables en este caso las palabras, y así ponderarlo con el peso o el total de la muestra. Teniendo así que entre mayor sea la probabilidad de ocurrencia esta tiene un valor mayor, sin embargo, para evaluar las palabras que no se encontraban en el conjunto de datos positivos o negativos se tomaron los datos en los que este valor era 0 para cada categoría. En el que se encontraron palabras como apestoso, bostezo, inapropiado, entre otras que lleva a que se tenga una reseña negativa, mientras que para una positiva se tienen creación, delicado, delicioso, y un gran contenido de nombres.



a) Reseñas negativas **b) Reseñas positivas**

Figura 5. Primeras 15 muestras ponderadas

a) Reseñas positivas **b) Reseñas negativas**

Figura 6. Palabras de las reaseñas

Conclusión algoritmos

A continuación, se muestra el score obtenido por cada uno de los modelos utilizados:

Tabla 2. Tabla de scores

Algoritmo	Score en Entrenamiento	Score en Prueba
Red Neuronal	0.999702	0.850104
Regresión Logística	0.935714	0.848022
MultinomialNB	0.931548	0.831367

Note que el color verde indica los puntajes mayores en entrenamiento y prueba dependiendo del algoritmo. En conclusión, se puede evidenciar que el mayor puntaje sobre el entrenamiento fue 0.999702 y sobre el test fue 0.850104 en el algoritmo de Red Neuronal.

Uno de los objetivos de la construcción de modelos era la de conocer que palabras tienen más impacto en la clasificación de la review, es decir conocer aquellas palabras con las que el público define mayoritariamente una review como buena o mala. Para este tipo de tarea que requiere una extracción e interpretación de los datos los modelos que funcionan en base a la probabilidad o la asignación de pesos ponderados suelen ser una buena fuente de información para mirar el impacto de las palabras. Aunque estos pesos pueden variar según el modelo implementándolo que cualquier persona esperaría sería una lista de adjetivos calificativos y cualitativos. Para el modelo de Regresión logística teniendo que fue el que obtuvo un mayor score, la lista de las 20 palabras con más influencia son.

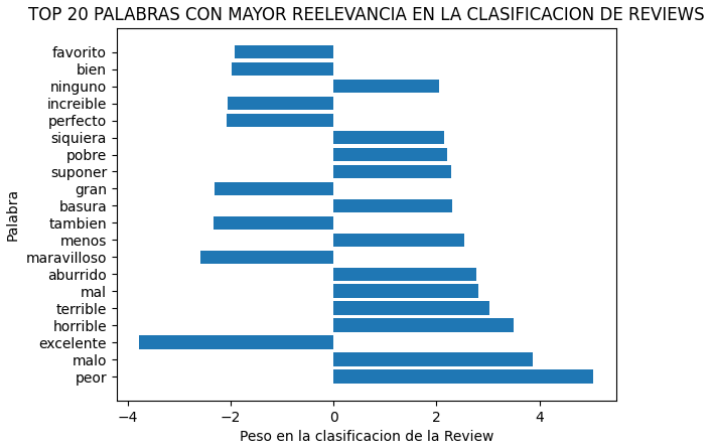
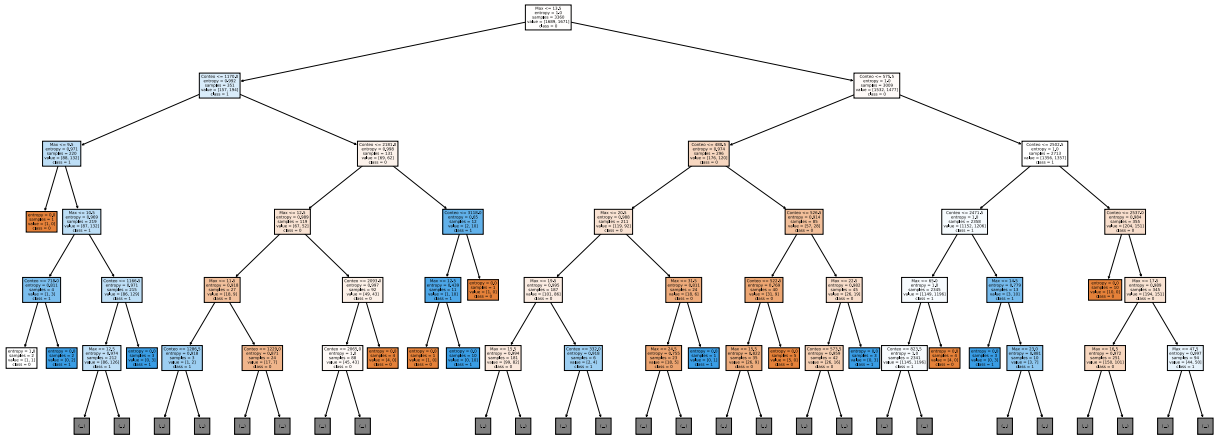


Figura LLL. Palabras con más impacto en la clasificación

Es de notar que en esta lista preponderan palabras comunes en el español, generalmente adverbios y adjetivos comparativos y destaca la ausencia de sustantivos por lo que se puede concluir que el modelo es coherente con las respuestas esperadas en un contexto real.

La segunda hipótesis que se quería evaluar era si la cantidad de texto que daba algún indicio si la reseña era positiva o negativa, para evaluar esto de una forma gráfica se

utilizó un árbol de decisión, dentro del cual se consideraros las columnas Max y Conteo, que son el número de caracteres que tiene la palabra más larga de la fila y el número de palabras de la fila respectivamente.



Aunque un análisis de las primeras ramas del árbol nos indica que efectivamente hay una relación entre el número de palabras y la clasificación de la review (la cual indica que a mayor cantidad de palabras la review es positiva). No obstante, a medida que se analiza el árbol en profundidad se observa que no existe una relación tan directa entre estas ya que otras instancias empiezan a obtener más relevancia en la clasificación. Esto se debe a que un 89,5 % de las reviews están por debajo de las 575 palabras; en este caso sí parece existir una relación con el porcentaje restante de reviews que poseen más de 575 palabras, pero menos de 1170 las cuales son positivas. Con los datos que se pueden visualizar se tiene que la gran mayoría de reviews positivas no pasan de 824 palabras y no son menores de 575. Para las reviews negativas no se puede establecer un compartimiento basado en el número de palabras.

4. Conclusión

Para los modelos utilizados se encontró que la regresión logística es la que da un mejor score de los datos junto con la red neuronal, esto se pudo deber a la cantidad de palabras tokenizadas. También se encontró con los modelos aplicados que la película tiene una alta probabilidad de ser catalogada negativamente si el espectador se siente aburrido, también si dentro del texto se tienen palabras negativas constantemente como horrible, mal, terrible (ver figura 2). Para las reseñas clasificadas como positivas se tiene que esto último ocurre al contrario, las palabras que más se tienen son maravilloso, gran, increíble y perfecto. Comparando este modelo con los otros dos, se tiene que las palabras calificativas se repiten en la red neuronal con la palabra despreciable (ver figura 3). Comparando con el modelo de multinomial se tiene que es el menos preciso de los tres modelos y aunque arroja las palabras que más frecuencia tienen no se obtienen palabras que puedan ayudar a tener un mejor entendimiento de los datos con las probabilidades. Por lo cual, si se toma un análisis de las palabras que no se encuentran en cada categoría

si se tiene que se acercan a los valores de los anteriores algoritmos teniendo palabras como bostezo

Como se evidencia en los resultados, se puede obtener no solo parte del comportamiento de las personas dentro de las películas, sino la percepción que tienen sobre las mismas. Pudiendo llevar esa información a los que las realizan como los directores y guionistas. Además, utilizando las reseñas positivas se pueden encontrar los temas de interés para las personas pudiendo generar más contenido de estas.

Reunión persona grupo expertos

En la reunión se dio una retroalimentación de las correcciones que se le debían realizar al documento para que quedase con una estructura más clara, también se habló de cómo con las gráficas y tablas ayudaban a ejemplificar de mejor manera los resultados para cada modelo y a realizar una comparación entre estos.

Referencias

[1] «Logistic Regression 3-class Classifier,» [En línea]. Available: https://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html.

[2] A. F. Jaurequi, «Machine Learning Data Science Cloud,» [En línea]. Available: https://anderfernandez.com/blog/naive-bayes-en-python/.

[3] Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional, México, 41, 12-16.