Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

# Proyecto 1 – Análisis de Sentimiento de Películas

## Inteligencia de Negocios - Grupo 19

## Tabla de Contenidos

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

# 0. Entendimiento del Negocio y Enfoque Analítico

| | |
|---|---|
| **Oportunidad / Problema de Negocio** | Identificar, basado en comentarios de películas, el sentimiento (positivo/negativo) atribuido a estas de millones de usuarios de manera automática, eficaz y rápida. |
| **Enfoque Analítico (Descripción del requerimiento desde el punto de vista de aprendizaje automático)** | Dada una muestra de datos de comentarios de películas y sus sentimientos atribuidos respectivos (pos/neg) queremos entrenar un modelo que logre predecir la categoría sentimental de un comentario mediante un análisis de texto en español. Haremos uso de técnicas de procesamiento de lenguaje natural así como modelos de machine learning para predecir el sentimiento del comentario. |
| **Organización y rol dentro de ella que se beneficia con la oportunidad definida** | Una organización encargada de la operación de los cines (e.g. CineColombia, RoyalFilms) podría utilizar estos modelos para determinar por cuanto tiempo debería permanecer una película en cines. Por ejemplo, una película con un buen número de comentarios positivos podría mantenerse en cine por más tiempo que una con mayor numero comentarios negativos. Por otro lado, podría funcionar también como insumo para plataformas que le recomiendan películas a sus usuarios, desde servicios de streaming como Netflix hasta portales informativos para amantes del cine como IMDB. |
| **Técnicas y algoritmos para utilizar** | Lemmatización<br>Bag of Words para extraer las features del texto<br>TFID Transformer<br><br>Aprendizaje Supervisado - Clasificación:<br>SGD<br>Naive Bayes<br>Regresión Logística<br>Random Forest<br>SVC<br>Gradient Boosting Classifier |

# 1. Entendimiento y Preparación de los Datos

*La sección que corresponde en el notebook tiene el mismo nombre y está en paréntesis.*

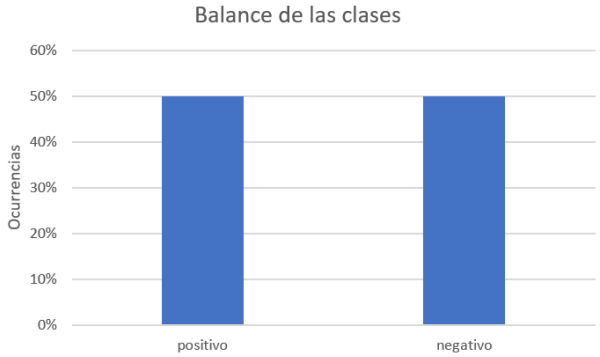
En este primer vistazo a los datos, logramos cargarlos de manera apropiada. El encoding para importar los datos es importante, ya que con UTF8 podemos visualizar las tildes, la cual no es la opción por defecto de herramientas como Excel.

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

Enseguida notamos de unos posibles errores en cuanto al lenguaje: la entrada 4998 que apareció en el primer vistazo estaba en inglés. Notamos también algunos tipos gramaticales que probablemente no deberían ser incluidos en el modelo final, como comillas o puntos.

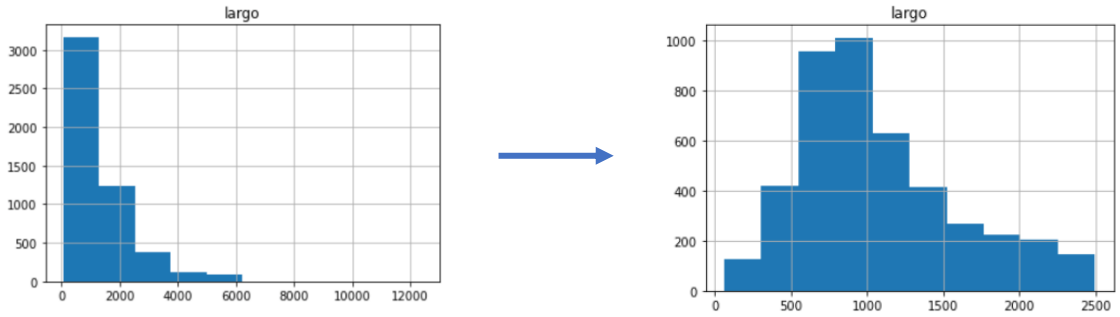
## 1.1 Clases Balanceadas o Imbalanceadas

Lo primero que hicimos fue ver si las categorías de clases para la variable objetivo (sentimiento) estaban apropiadamente balanceadas. Afortunadamente, la distribución era de un nivel apropiado, 50-50 de cada lado.




## 1.2 Largo de las Cadenas

Después de ver algunos ejemplos de entradas en la muestra de datos que nos presentaron, logramos ver que la cadena de texto como entrada puede variar mucho. Por eso, utilizamos un análisis estadístico sobre el largo de cada cadena de texto para ver si hay posibles problemas. Inmediatamente notamos que hay un valor máximo de 12375 caracteres, cual era muy inusual. Además, varios comentarios con tamaños mucho más grandes que el 75% de 1700 (algunos mayor a 3000, por ejemplo). También notamos que, al quitar estos valores, la distribución de tamaño es mucho más apropiada.

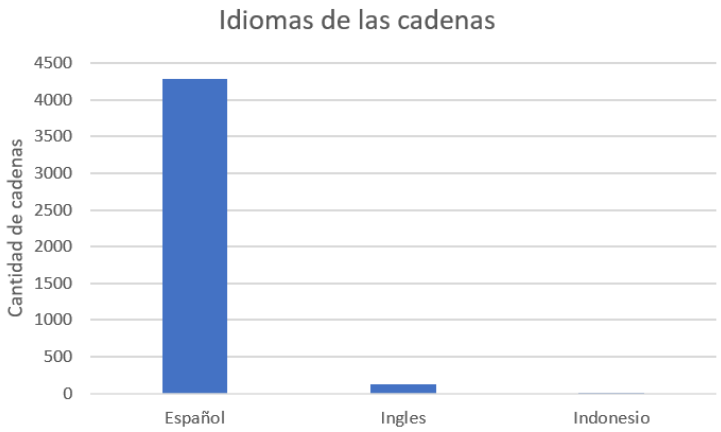


Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

### 1.3 Idioma de las Cadenas

Mencionado anteriormente, se presentaron casos de comentarios en inglés. Utilizamos una librería externa para analizar el idioma de los comentarios, y determinó que, en efecto, hay varias entradas que estaban en inglés. Incluso, una entrada con el idioma en indonesio. Estas también fueron filtradas fuera de los datos para entrenar debido a que puede afectar el desempeño del modelo.

```
es    4281
en     118
id       1
Name: idioma, dtype: int64
```




## 2. Preprocesamiento para análisis de texto

Para el análisis de texto, no solo basta con hacer la limpieza usual de datos que hemos hecho en laboratorios anteriores. Principalmente tenemos solo un variable que usar, “review”, y un variable objetivo “sentimiento”. Pero en verdad, no es así: queremos extraer variables del texto para poder crear un modelo apropiado para la clasificación. Por ende, utilizamos una técnica de preprocesamiento llamada *Bag of Words* para poder extraer features del texto. Esta técnica extrae las palabras de los textos y los convierte en variables binarias, 1 si está presente en el texto y 0 si no. Por ende, también es importante reconocer que esta técnica crea un alto numero de variables para los modelos.

### 2.1 Limpieza y Lemmatización

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

Primero, tenemos que limpiar el texto de entrada para que la creación del bag of words sea apropiada. Esto consta con la lematización, cual tiene como objetivo limpiar el texto para que solo quede una lista de palabras separadas por espacio, limpiando objetos gramaticales como puntuación o diferencias en cuanto a mayúsculas y minúsculas. Principalmente:

- Eliminó caracteres especiales
- Eliminó caracteres singulares (a, y, o)
- Eliminó caracteres singulares del principio del texto
- Eliminó stopwords del texto
- Eliminar el prefijo b de algunas oraciones en Python (ya que son bytecoded)
- Convertir todo a minúsculas

Lemmatización: Reducir un conjunto de palabras a solo una representación dependiendo de distintas formas morfológicas. Se llevo a cabo una lematización para reducir la dimensionalidad de el conjunto de datos a través del lematizador WordNetLemmatizer() de sklearn.

## 2.2 Bag of Words

En este paso, se utilizó la técnica de Bag of Words para extraer las features utilizando la lematización previa, y la librería de Scikit Learn. En este modelo, se considera cada documento como un conjunto no ordenado de palabras y se cuenta el número de veces que aparece cada palabra en el documento.


## 2.3 TFID Transformer

Este último paso para la preparación de datos es mover la matriz de recuento a una normalizada de TFIDF, es decir, una que tiene en cuenta *term frequency* y *inverse document frequency*. Esto es para atribuirle mayor peso a palabras que no son tan recurrentes en una parte del texto (y quitarle importancia a palabras comunes como conectores, preposiciones, etc.)

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

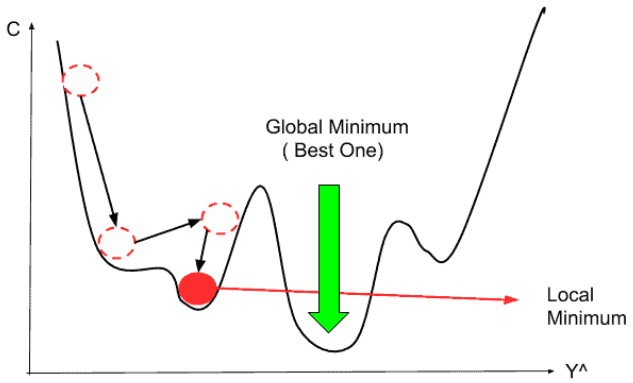
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

# 3. Modelado y Evaluación

Aunque se requerían tres algoritmos mínimos, decidimos probar dos algoritmos cada uno, y asimismo escoger el modelo más apropiado de los tres para tener como nuestra base de tres algoritmos, profundizando su análisis mediante GridSearch. De ahí, nos reunimos otra vez en grupo para seleccionar el modelo “ganador”: el mas apropiado y que tiene mayor eficiencia. En este punto también tratamos de indagar con el experto en estadística que modelos se adecuan más para la tarea que estamos tratando de explorar y de que manera estos logran un rendimiento alto con tareas de lenguaje natural.

## 3.1 Stochastic Gradient Descent Classifier – Camilo Falla

El SGD Classifier (Stochastic Gradient Descent Classifier) es un algoritmo de aprendizaje automático utilizado para la clasificación de texto en procesamiento de lenguaje natural (NLP). Es un tipo de clasificador lineal que utiliza el descenso de gradiente estocástico para optimizar la función de pérdida y encontrar los mejores pesos para las características del modelo.



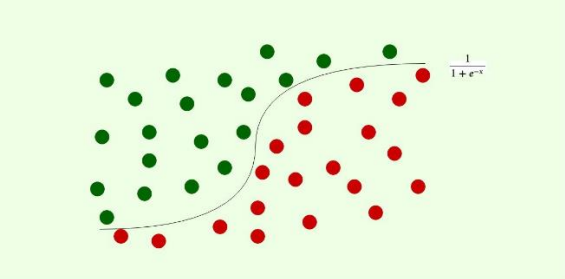

El SGD Classifier se utiliza para clasificar los textos en dos o más categorías, y se entrena utilizando un conjunto de datos de entrenamiento etiquetado. Durante el

entrenamiento, el algoritmo actualiza iterativamente los pesos de las características del modelo para minimizar la función de pérdida, que mide la discrepancia entre las predicciones del modelo y las etiquetas verdaderas.

El modelo SGD Classifier también admite la regularización L1 y L2, que son técnicas para evitar el sobreajuste del modelo. Además, también es posible utilizar diferentes funciones de pérdida y diferentes configuraciones de hiperparámetros para adaptar el modelo a diferentes conjuntos de datos y mejorar su rendimiento.

## 3.2 Logistic Regression – Pablo Pastrana

La Logistic Regression Classifier, también conocida como regresión logística, es un modelo de clasificación utilizado en el aprendizaje automático. A diferencia de la regresión lineal, que se utiliza para predecir valores continuos, la regresión logística se utiliza para predecir la probabilidad de que una instancia pertenezca a una de las dos o más categorías.



La regresión logística utiliza una función sigmoide para transformar una entrada lineal en una salida entre 0 y 1, que representa la probabilidad de que una instancia pertenezca a una de las categorías.

## 3.3 Naïve Bayes (Multinomial) – Santiago Campo

El algoritmo de Naive Bayes es un modelo probabilístico en la cual se utiliza ampliamente el teorema de Bayes, y su respectiva formula, para hacer predicciones probabilísticas al asociar una entrada con las respectivas clases.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Se le denomina *naive* por el hecho de que, para utilizar probabilidad bayesiana, se asumen que las variables de entrada son independientes entre sí, reduciendo la formula a una regla de clasificación:

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

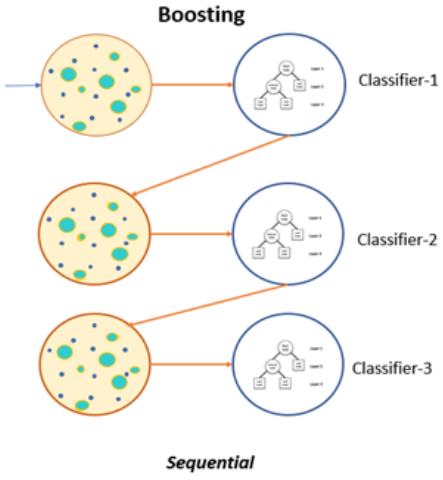
$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

Utilicé la versión multinomial, más apropiada para la clasificación de texto. Asigna la probabilidad de que cada feature pertenezca en un ejemplo de cada clase (en nuestro caso, probabilidad de que la palabra aparezca en un ejemplo negativo / positivo). Tiene como hiperparámetros *Alpha* (parámetro de smoothing), *fit prior* y *class prior*. Utilicé GridSearch para ver cuáles eran los más apropiados.

## 3.4 Gradient Boosting Classifier - Camilo Falla

El Gradient Boosting Classifier es un modelo de aprendizaje automático supervisado utilizado para problemas de clasificación. Es una técnica de boosting, lo que significa que combina varios modelos de aprendizaje débiles (conocidos como estimadores base) para construir un modelo más fuerte y preciso.

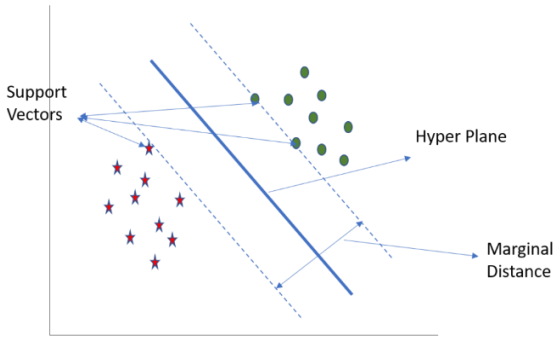



El Gradient Boosting Classifier funciona de manera iterativa, en cada iteración ajusta un nuevo estimador base en los errores del modelo anterior. En cada iteración, el modelo intenta corregir los errores del modelo anterior, ajustando el siguiente modelo base en la dirección del gradiente de la función de pérdida. En resumen, el modelo aprende de sus errores y ajusta los pesos de las características para minimizar la función de pérdida.

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

## 3.5 Support Vector Machine – Pablo Pastrana

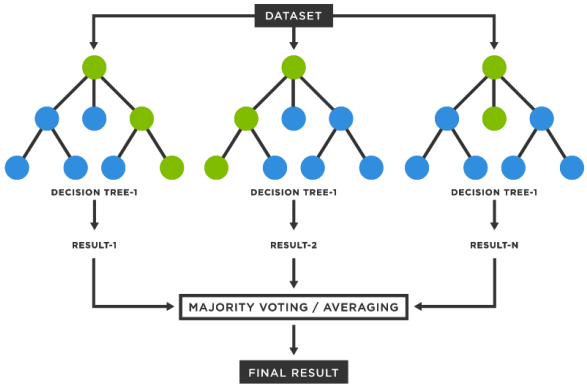
El Support Vector Classifier (SVC) es un modelo de clasificación utilizado en el aprendizaje automático. Se basa en la idea de encontrar un hiperplano en un espacio de alta dimensión que separe las clases de datos de entrada.




El SVC utiliza vectores de soporte para definir el hiperplano. Los vectores de soporte son las instancias que están más cerca del hiperplano y que ayudan a definir la frontera de decisión. La distancia entre los vectores de soporte y el hiperplano se llama margen, y el objetivo del SVC es maximizar este margen para mejorar la capacidad de generalización del modelo.

## 3.6 Random Forest – Santiago Campo

Random Forest es un *meta estimator* debido a que en sí es una agrupación de otro método de clasificación. Consiste en un grupo de árboles de decisión agrupados y ajustados a diferentes subparticiones de los datos y utiliza promedios de los resultados para mejorar la accuracy y controlar el overfitting.



Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

## 4. Resultados

En una de nuestras reuniones de seguimiento, mostramos los modelos desarrollados a los demás integrantes del grupo y decidimos que el mejor modelo para aplicar a la problemática en particular era el de SGD (Stochastic Gradient Descent). No solo obtuvo el mejor desempeño en cuanto al reporte de clasificación y matriz de confusión, sino que también tiene una buena reputación como un algoritmo versátil que es comúnmente utilizado en tareas de clasificación de texto y el Natural Language Processing en general. En este punto también intentamos indagar con el experto en estadística cuales herramientas existen para evaluar modelos y si es suficiente con el análisis de precisión, recall, f1 score y matrices de confusión.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negativo | 0.85 | 0.87 | 0.86 | 424 |
| positivo | 0.87 | 0.85 | 0.86 | 433 |
| accuracy | | | 0.86 | 857 |
| macro avg | 0.86 | 0.86 | 0.86 | 857 |
| weighted avg | 0.86 | 0.86 | 0.86 | 857 |

Accuracy: 0.86

| | Actual Pos | Actual Neg |
|---|---|---|
| Pred Pos | 370 | 54 |
| Pred Neg | 66 | 367 |

**Impacto al negocio:** Los resultados encontrados tendrán un gran impacto en el negocio. Este podrá mejorar la experiencia del usuario al mostrarle comentarios buenos o malos dependiendo de la preferencia que cada usuario tenga. De este modo, la satisfacción general de los usuarios aumentará, lo cual será bastante beneficioso para la rentabilidad general de la organización. Además, la empresa también podrá hacer uso de la categorización automática de los comentarios para saber que series están teniendo mayor aceptación y de esta manera, poder mantenerlas en oferta por un mayor tiempo.

## 5. Trabajo en Equipo

Santiago Campo 201921995
Camilo Falla 201821059
Pablo Pastrana 201822920

## 5.1 Distribución de Roles

Santiago Campo – Líder de negocio

Camilo Falla – Líder del Proyecto, Líder de analítica

Pablo Pastrana – Líder de datos

## 5.2 Retos Enfrentados y Puntos a Mejorar

Afortunadamente, no presentamos retos mayores en la realización del proyecto, pero hubo algunos puntos imprevistos en la cual se necesitaba la atención de todo el grupo. Por ejemplo, al notar que había comentarios en inglés, decidimos todos buscar una librería apropiada que pudiera resolver el problema y realizar una limpieza eficiente de los datos.

En cuanto a puntos a mejorar, quisiéramos aprovechar mejor el trabajo interdisciplinario y la pareja escogida para el desarrollo del proyecto, ya que pueden aportar a la selección de modelos y realizar un análisis cuantitativo más apropiado. Esperamos contar con su apoyo para la siguiente etapa.

## 5.3 Distribución de Puntos

Decidimos distribuir los puntos equitativamente, ya que logramos desarrollar el trabajo en equipo de una manera bien distribuida. Esto quedaría con todos teniendo 33 puntos, pero, con 1 punto de sobra debido a los 100 puntos, decidimos dar un punto al líder del proyecto por la organización de la entrega (Camilo).

## 5.4 Reuniones y Tareas

Además de las reuniones recomendadas en el enunciado (Lanzamiento, Ideación, Seguimiento y Finalización), tuvimos una reunión de seguimiento especial para exponer los mejores modelos de cada uno y elegir el modelo final.

En cuanto a tareas, primero hicimos una revisión inicial de los datos como grupo y el líder de datos se encargó de hacer la preparación adecuada de estos. Luego, dividimos el trabajo de algoritmos para que cada uno desarrolle dos, y definimos que al menos uno de esos utilice GridSearch para la búsqueda de hiperparámetros. Al seleccionar el modelo definitivo final en una reunión, trabajamos en los resultados y la presentación de estos juntamente con el líder de negocio.