Universidad de los Andes

Proyecto 1, parte 1:

Construcción de modelos de analítica de textos

Leandro Yara Ramirez

Daniel Bernal Caceres

Mateo Lopez Cespedes

Universidad de los Andes

Inteligencia de negocios

Bogotá, Abril 2

Universidad de los Andes

Roles de proyecto:

Lider de proyecto: Leandro Esteban Yara Ramírez

Lider de negocio: Leandro Esteban Yara Ramírez

Lider de datos: Mateo Lopez Cespedes

Lider de analitica: Daniel Andrés Bernal Cáceres

Algoritmos realizados:

SVC: Daniel Andrés Bernal Cáceres

Bayes ingenuo multinomial: Leandro Esteban Yara Ramírez

KNN: Mateo López Céspedes

Entendimiento del negocio y enfoque analitico:

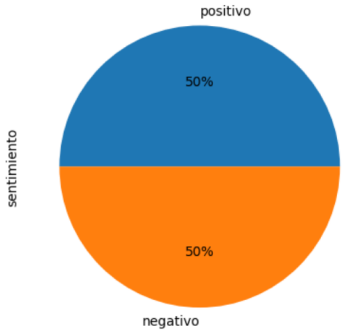
Oportunidad/ problema negocio	La necesidad de realizar analítica de textos, en este caso análisis de sentimientos, para conocer así las opiniones de los clientes respecto a un aspecto que hace parte de la operación del negocio y que puede impactar la forma en como operan o como generan ganancias.
Enfoque analítico	Implementar un modelo de aprendizaje automático para analizar los textos que proporciona la organización con el objetivo de determinar los aspectos que hacen que una opinión sea positiva o negativa respecto a una película.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Si se asume que la organización es un negocio dedicado a las películas, este trabajo de analítica influenciará a la división de contenidos de la empresa pues podrán saber qué clase de película es la que más gusta y la que menos gusta, lo que puede guiarlos en decidir sobre qué géneros vale la pena invertir.
Técnicas y algoritmos a utilizar	Tipo de aprendizaje: Supervisado Tarea de aprendizaje: Clasificación

	Algoritmos: KNN, Multinomial Naive-Bayes, SCV.

Entendimiento y preparación de los datos:

Entendimiento de datos:

- El número de reseñas positivas es igual al número de reseñas negativas, por lo que la distribución de datos está balanceada. Esto nos dice que la métrica de exactitud será relevante en la evaluación de los modelos.



- En el top 20 de palabras más usadas en las reseñas se identifican 19 palabras son conjunciones. Esto indica que se necesita hacer una limpieza de palabras para reducir el número de similitudes que hay entre las reseñas positivas y negativas.

```
[('de', 63856),
 ('que', 36188),
 ('la', 34760),
 ('y', 26879),
 ('en', 25251),
 ('a', 20734),
 ('el', 19636),
 ('un', 17788),
 ('es', 16077),
 ('una', 15773),
 ('los', 13193),
 ('se', 12130),
 ('no', 11373),
 ('película', 11003),
 ('para', 8935),
 ('con', 8544),
 ('lo', 8412),
 ('su', 8169),
 ('las', 8090),
 ('por', 7823)]
```

- En el top 20 de palabras menos usadas en las reseñas se identifican 9 palabras mal puntuadas, 4 palabras con paréntesis mal ubicados y 2 palabras mal escritas. Esto nos dice que hay palabras que no permiten realizar una generalización entre las reseñas para clasificarlas.

```
[('hija.Con', 1),
 ('felicidad.En', 1),
 ('Teer)', 1),
 ('(Mc', 1),
 ('Generally,', 1),
 ('bump', 1),
 ('schlock', 1),
 ('late-hour', 1),
 ('wee', 1),
 ('TV-movie', 1),
 ('película".Lástima', 1),
 ('ansiedad.Supongo', 1),
 ('mejoraría', 1),
 ('a.movies', 1),
 ('disco.Si', 1),
 ('turn).', 1),
 ('(sans', 1),
 ('D.V.Pero', 1),
 ('inconsciente.No', 1),
 ('cine.Pero', 1)]
```

Preparación de los datos:

Luego de analizar los datos, se identifica que la mejor estrategia es eliminar las palabras que generen problemas y vectorizar los datos para poder ser usados en los modelos de predicción, por lo que:

- Se importa de ntlk la lista de palabras irrelevantes en español y se crea un stemmer que permitirá simplificar el análisis al reducir las palabras hasta su raíz lingüística.
- Se crea un vectorizador que indica que el análisis se hará sobre las 2500 palabras significativas más usadas y no pueden aparecer en más del 80% de las reseñas (para evitar redundancia).
- Se aplica un método en el que se eliminan todos los caracteres de una reseña que no son letras del alfabeto, se convierten todas las letras en minúscula, se excluyen las palabras insignificantes de una reseña y las restantes son transformadas en su raíz lingüística.
- El anterior método se aplica para todas las reseñas que existen en el archivo de datos.
- Se eliminan todas las palabras que aparecen 10 o menos veces en todo el conjunto de reseñas (significancia estadística).
- Se ejecuta el vectorizador sobre el conjunto de reseñas para obtener un dataframe que nos indica cuántas veces aparece cada palabra en cada reseña.
- A partir del dataframe modificado, se generan los conjuntos de entrenamiento y prueba.

Modelado y evaluación:

SVM:

SVM es un modelo matemático que se utiliza para clasificar los datos en dos o más categorías. El algoritmo busca un hiperplano que separa los datos en las distintas categorías de manera óptima, maximizando la distancia entre los puntos de datos más cercanos al hiperplano de cada categoría.

La razón por la cual se usa SVM es porque es una técnica efectiva en la clasificación de datos en una variedad de campos. Además, SVM es capaz de manejar conjuntos de datos de alta dimensionalidad y puede trabajar con datos tanto linealmente separables como no linealmente separables utilizando diferentes técnicas de kernel. En el caso del proyecto se utiliza un kernel de núcleo lineal, osea se utilizó una función lineal para transformar los datos sin agregar nuevas características a estos.

De acuerdo a la precisión del modelo SVM lineal es del 83,7%, lo que significa que el modelo clasifica correctamente el sentimiento de aproximadamente el 84% de las muestras del conjunto de prueba.

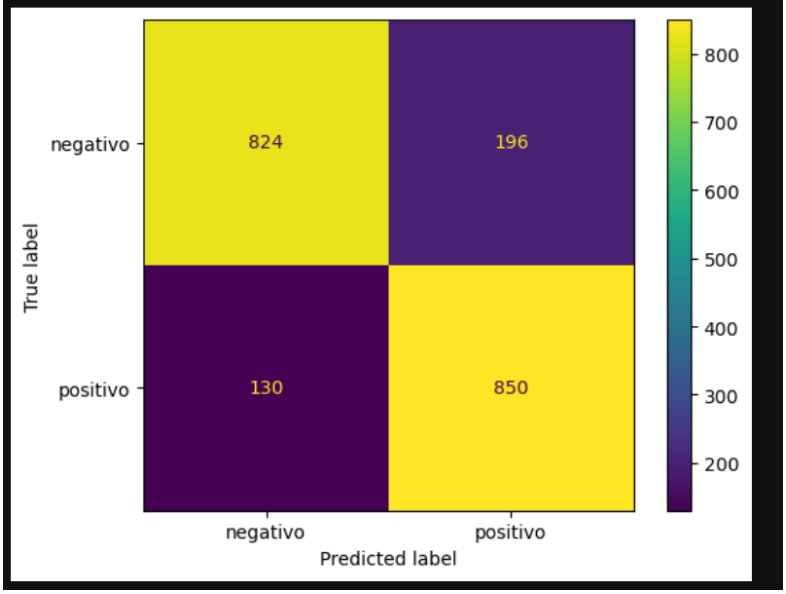
```
# Predecir los valores en el conjunto de prueba
y_pred = clf.predict(X_test)

# Calcular la precisión
accuracy = clf.score(X_test, y_test)
print('Precisión:', accuracy)
```

```
Precisión: 0.837
```

```
# Se genera la matriz de confusión
ConfusionMatrixDisplay.from_predictions(y_test, y_pred)
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x15425b7de70>
```



Bayes ingenuo multinomial:

Bayes ingenuo es un algoritmo predictivo de clasificación basado en el Teorema de Bayes. Este consiste en calcular la probabilidad de un evento dadas las condiciones relacionadas al mismo. En este caso, el evento es el tag de clasificación que indica si la reseña es positiva o no, mientras que las condiciones son las palabras contenidas en la reseña. Se considera ingenuo porque asume independencia entre la probabilidad de los eventos, osea, la probabilidad de que una reseña sea de x o y tipo no es afectada por la probabilidad de otra reseña.

La versión multinomial de este algoritmo se basa en la frecuencia de aparición de las palabras en otras reseñas para asignar la etiqueta de clasificación. Este proceso se vuelve más eficiente luego de eliminar las palabras con baja significancia estadística y palabras que aparecen demasiado.

Según los resultados del modelo, se clasifican correctamente el 83,2% de las reseñas que son analizadas por el algoritmo. Esto representa un porcentaje adecuado y realista de predicción (entre 70 y 90%). Además, el nivel de falsos positivos y falsos negativos es balanceado (130 - 121), por lo que parece no haber problemas graves de significancia entre palabras.

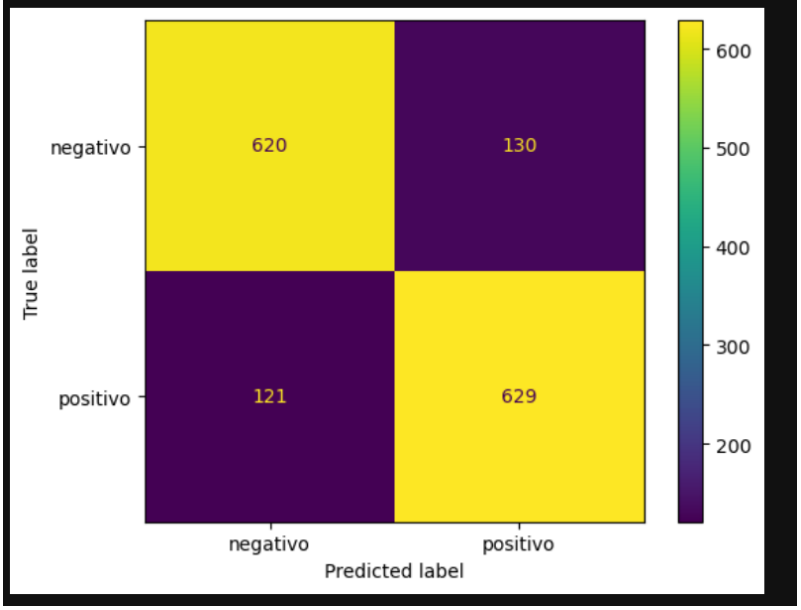
```
#Evaluacion del modelo:
print("Puntuacion de entrenamiento: %f" % nb.score(X_train, Y_train))
print("Puntuación de prueba: %f" % nb.score(X_test, Y_test))
```

```
Puntuacion de entrenamiento: 0.876286
Puntuación de prueba: 0.832667
```

```
# Se predicen los resultados del conjunto de prueba
Y_pred = nb.predict(X_test)
```

```
# Se genera la matriz de confusión
ConfusionMatrixDisplay.from_predictions(Y_test, Y_pred)
```

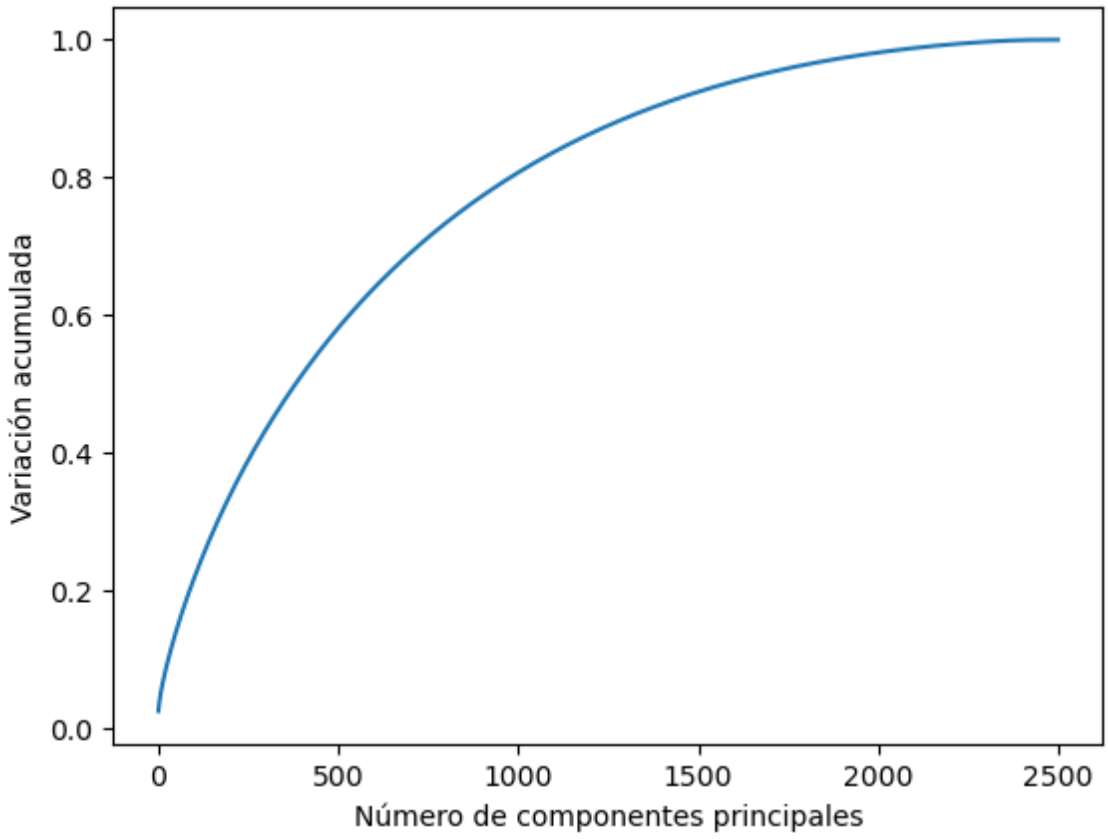
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1542242d690>
```



KNN:

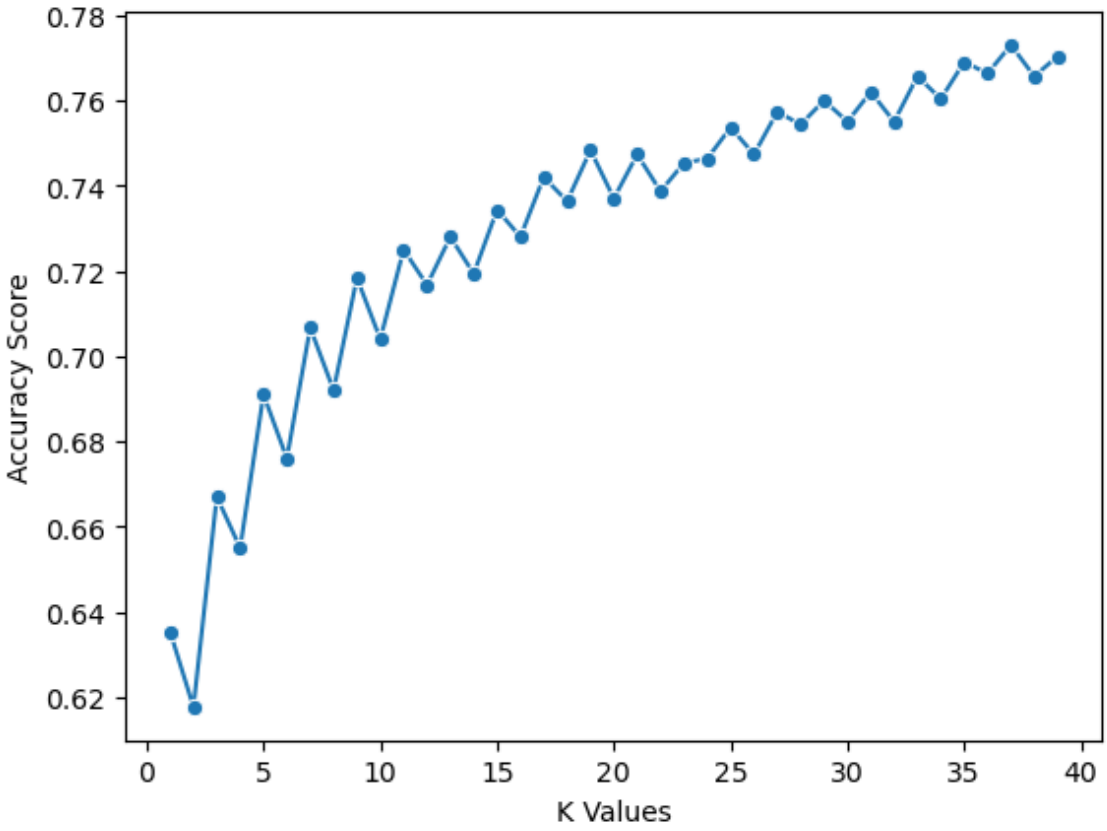
K-Nearest-Neighbors es un algoritmo de aprendizaje automático supervisado cuyo objetivo principal es la clasificación de nuevos datos en categorías definidas por unos datos de entrenamiento. KNN funciona de forma que toma como métricas la distancia de un nuevo dato a sus k vecinos más cercanos y se clasifica a este de acuerdo a la categoría de sus vecinos respectivos. Se decidió usar este algoritmo pues por un lado los resultados que puede obtener se alinean con los objetivos de negocio, además, la forma en que funciona resulta indicada para análisis de emociones en texto, pues es de esperar que palabras que describan una emoción sean vecinas si se graficaran.

En nuestro caso, para implementarlo primero realizamos un PCA o Principal Component Analysis con el fin de reducir la dimensionalidad del grupo de datos resultante de la limpieza, pues tenía cerca de 2500 columnas.



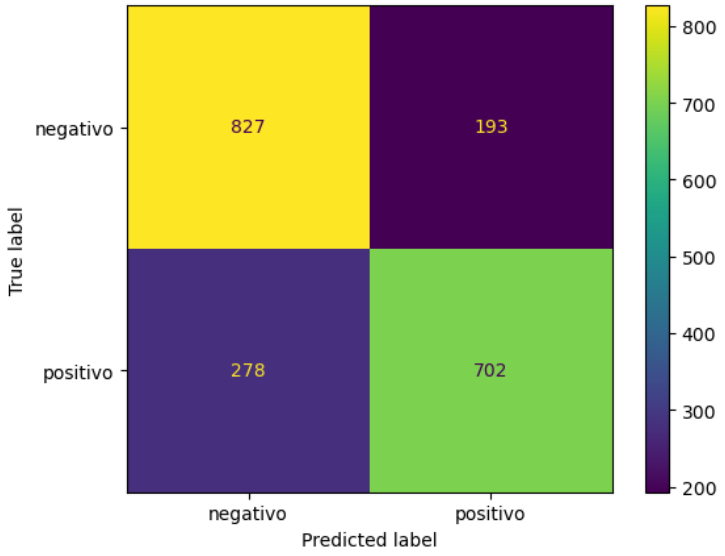
El resultado de esto fue que podíamos efectivamente reducir el número de características de 2500 a 1750 pues con este número se explicaba más del 95% de la variabilidad de los datos.

Una vez tuvimos el conjunto de datos lo más condensado posible procedimos a buscar los hiperparametros de KNN que resultaron en la mayor precisión:



Lo que dio como resultado que el modelo tenía la mejor precisión cuando k=38, la cual corresponde a 76.4%

Por último, encontramos la matriz de confusión:



Conclusiones:

- El algoritmo de predicción más certero para el análisis de reseñas es SVM (83.7% de exactitud). Además, permite analizar gráficamente cual es la categoría más acertada para una reseña.
- La compañía debería definir un mínimo de palabras que se deben usar para generar una reseña. Esto con el fin de que las reseñas puedan ofrecer información más significativa respecto al número de palabras que se pueden analizar.
- Según el wordcloud generado, dentro del análisis de datos, las personas hablan en mayor medida de los personajes, el final y la historia de la película.

Reflexión del trabajo en equipo:

El trabajo fue repartido de forma que cada integrante tuvo varias tareas por completar de acuerdo a su rol asignado en la reunión inicial. Leandro Yara fue el líder del negocio, desarrolló el modelo de Bayes y se encargó del entendimiento del negocio y los datos. Dedico aproximadamente 6 horas en total. Mateo López fue el líder de datos, se encargó del análisis de los datos y del desarrollo del modelo de KNN, dedicó 6 horas a sus tareas. Daniel fue el líder de analítica y se encargó de realizar el modelo SVC y de puntualizar los resultados. Dedicó 6 horas al desarrollo de sus tareas.

Si tuviéramos que repartir 100 puntos entre los integrantes, los repartiríamos equitativamente pues todos tuvimos una carga de trabajo equitativa y los aportes de todos fueron importantes para lograr el objetivo del proyecto. Si tuviésemos que corregir algo para otro proyecto mejoramos la planeación de las actividades y la comunicación.