Daniel Esteban Aguilera Figueroa – 202010592
Laura Vanessa Martínez Prieto – 202012624
Cristian Armando Sánchez Ocampo – 202022112

Informe: Proyecto 1 – Etapa 1

1. Introducción

La Analítica de Textos (AT) es un campo de estudio que combina el aprendizaje automático y el procesamiento de lenguaje natural para procesar grandes cantidades de textos y extraer conocimiento útil para la toma de decisiones. Este proyecto en específico se enfoca en fortalecer las competencias necesarias para una aplicación efectiva del proceso de descubrimiento de conocimiento a partir de textos, utilizando una metodología propia del mundo de la analítica. Una de las opciones de problemáticas a abordar en este proyecto es el análisis de sentimientos de películas en español, se tienen comentarios de películas que deben ser clasificados en las categorías de positivo o negativo.

2. Entendimiento del negocio y enfoque analítico

Oportunidad/problema Negocio	La oportunidad y/o problema de negocio se basa en brindar soluciones que permitan a las empresas automatizar el análisis de grandes bases de datos y textos provenientes de diversas fuentes, como la Web, redes sociales, librerías digitales, entre otros. En este sentido, el proyecto específico busca desarrollar una herramienta de análisis automático de sentimientos de comentarios de películas en español, la cual podría ser de gran utilidad en la industria del cine.
Enfoque analítico (Descripción del requerimiento desde el punto de vista de aprendizaje automático)	El proyecto propuesto, que se enfoca en el análisis de comentarios de películas, se trata claramente de un problema de clasificación binaria, en el cual se utiliza el aprendizaje supervisado para entrenar un modelo con un conjunto de datos etiquetados y, posteriormente, se emplea dicho modelo para predecir la polaridad de nuevos comentarios. Para lograr clasificar los comentarios de películas en categorías de positivo o negativo, se aplican técnicas de aprendizaje automático y procesamiento de lenguaje natural en los modelos utilizados.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Teniendo en cuenta el contexto específico del proyecto de análisis de comentarios de películas, la organización que se beneficiaría sería aquella que tiene interés en analizar los

	comentarios o reseñas de películas en español para clasificarlos en positivos o negativos. Algunos ejemplos de esto podrían ser una empresa cinematográfica que desea obtener información sobre cómo sus películas son percibidas por el público o una plataforma de streaming que busca mejorar su algoritmo de recomendación de películas.
Técnicas y algoritmos a utilizar	Se utilizarán técnicas de análisis de texto, como la tokenización, la eliminación de stopwords, la lematización, la vectorización y la selección de características. Asimismo, se han elegido tres algoritmos: Naive Bayes, Redes Neuronales y Random Forest.

3. Entendimiento y preparación de los datos

Durante el inicio del entendimiento de los datos se tuvieron en cuenta las columnas mencionadas en el archivo DiccionarioPeliculas.xlsx, las cuales fueron usadas durante este proceso. Estas son:

- Id – identificador de la revisión
- review_es – Contenido de la revisión a procesar
- sentimiento – Sentimiento relacionado a la revisión (positivo-negativo)

Para el entendimiento de estas columnas bastó con la revisión de las revisiones propuestas, para las cuales se obtuvo que se encontraban en español, lo cual muestra un desarrollo diferente a si estas estuvieran en inglés debido a los diferentes caracteres especiales presentes. Adicionalmente, el contenido de la revisión es lo que define si un sentimiento es positivo o negativo.

En cuanto la preparación de los datos se siguieron los siguientes pasos para el desarrollo:

1. Transformar todas las palabras a minúsculas.
2. Eliminar signos de puntuación y caracteres especiales.
3. Transformar los caracteres con tilde a su forma base.
4. Eliminar las palabras que no tienen una representación relevante para el modelo (Stopwords).
5. Reemplazar los caracteres numéricos que posiblemente tienen una representación textual.
6. Tokenización, Lematización y Stemming a todas las palabras para su representación base y fructífera para el modelo.
7. Transformación de la variable objetivo para una representación numérica (1 – Positivo, 0 – Negativo)

Estos pasos por seguir se llevaron a cabo con el fin de tener únicamente las palabras relevantes para el modelo y así tener un menor ruido en los datos y mejores resultados.

4. Modelado y evaluación

a. Naive Bayes – Laura Martínez

Luego de la preparación de los datos se desarrolló del modelo de Naive Bayes y se comenzó separando los datos en conjuntos de entrenamiento y prueba. Luego, se vectorizan los datos con ayuda de CountVectorizer. Asimismo, se entrenó y se predijo el modelo con ayuda de MultinomialNB() de la librería Sklearn y estos fueron los resultados:

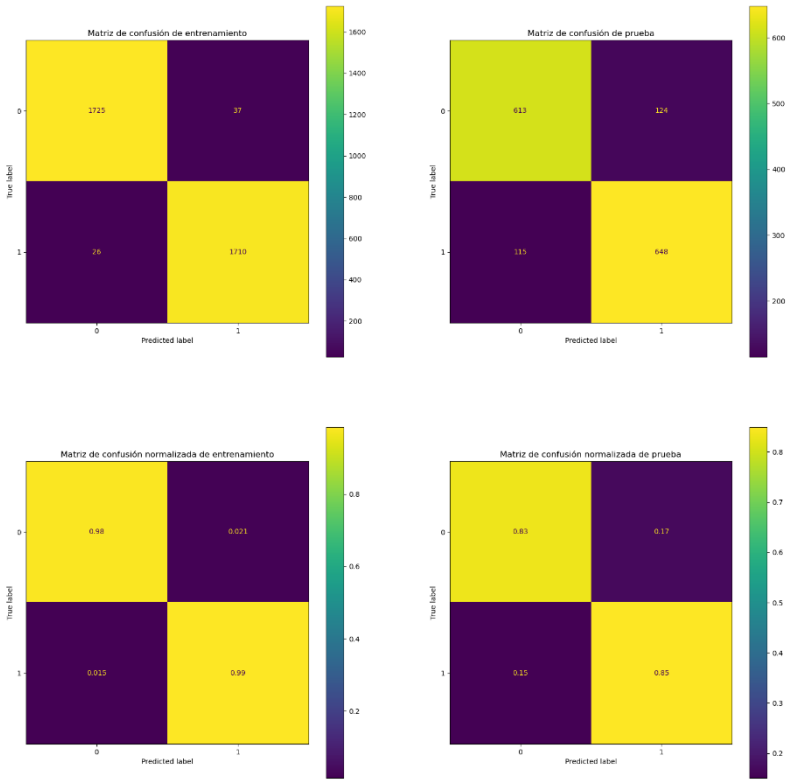
```
Exactitud: 0.82
Recall: 0.7575360419397117
Precisión: 0.8639760837070254
Puntuación F1: 0.8072625698324023
```

	precision	recall	f1-score	support
0	0.78	0.88	0.82	737
1	0.86	0.76	0.81	763
accuracy			0.82	1500
macro avg	0.82	0.82	0.82	1500
weighted avg	0.82	0.82	0.82	1500

b. Neural Network – Daniel Aguilera

Para el desarrollo de este modelo, se cargaron los datos preprocesados anteriormente y se vectorizaron las palabras una vez cargadas para así poder iniciar con el modelo. Se utilizaron dos formas de vectorizar las palabras (CountVectorizer y TF-IDF) y se probaron estas dos para saber cual generaba un mejor resultado. En cuanto al modelo, se usó MLPClassifier de la librería Sklearn. En cuanto a los hiperparametros usados para el modelo, se hizo una prueba manual para encontrar la combinación que llevaba a un mejor resultado.

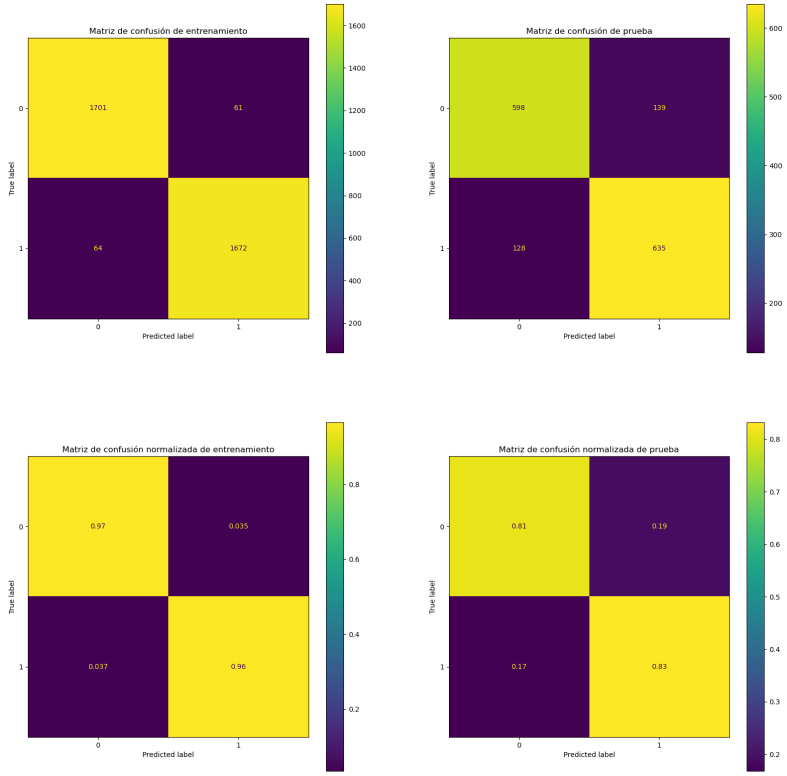
- **Prueba 1:** Count Vectorizer



Resultados:

	Accuracy	F1	Precision	Recall
Entrenamiento	0.981990	0.981990	0.982010	0.981990
Prueba	0.840667	0.840644	0.840683	0.840667

- **Prueba 2:** TF-IDF



Resultados

	Accuracy	F1	Precision	Recall
Entrenamiento	0.964265	0.964265	0.964266	0.964265
Prueba	0.822000	0.821968	0.822024	0.822000

Luego de realizar distintas pruebas con ambas formas de vectorizar y la red neuronal, podemos concluir que el modelo con CountVectorizer es el que mejor resultados nos da, por lo que podemos concluir que el modelo de redes neuronales es el que mejor se adapta a este problema en particular. Se tiene en cuenta que esta red neuronal en específico tiene algunos problemas con la jerarquía de los datos, ya que no se puede tomar en cuenta la relación entre las palabras. Aun así, tenemos un resultado del F1 Score de 84%.

c. Random Forest – Cristian Sánchez

Para el desarrollo del modelo *Random Forest* se utilizó los datos

previamente tratados. Con estos datos ya cargados se procedió a realizar el modelo con los siguientes pasos:

- Se hace la separación de los datos en datos de entrenamiento y prueba.
- Se vectorizan las palabras ya tratadas para utilizar en el modelo. Se hace uso el método de TF-IDF para realizar esta tarea. Se hace tanto la vectorización del conjunto de prueba como el de entrenamiento.
- Se corre el modelo RandomForestClassifier() con los datos vectorizados y teniendo los siguientes resultados:

```
[0.77       0.786      0.79       0.766      0.79759519]

0.7819190380761524
```

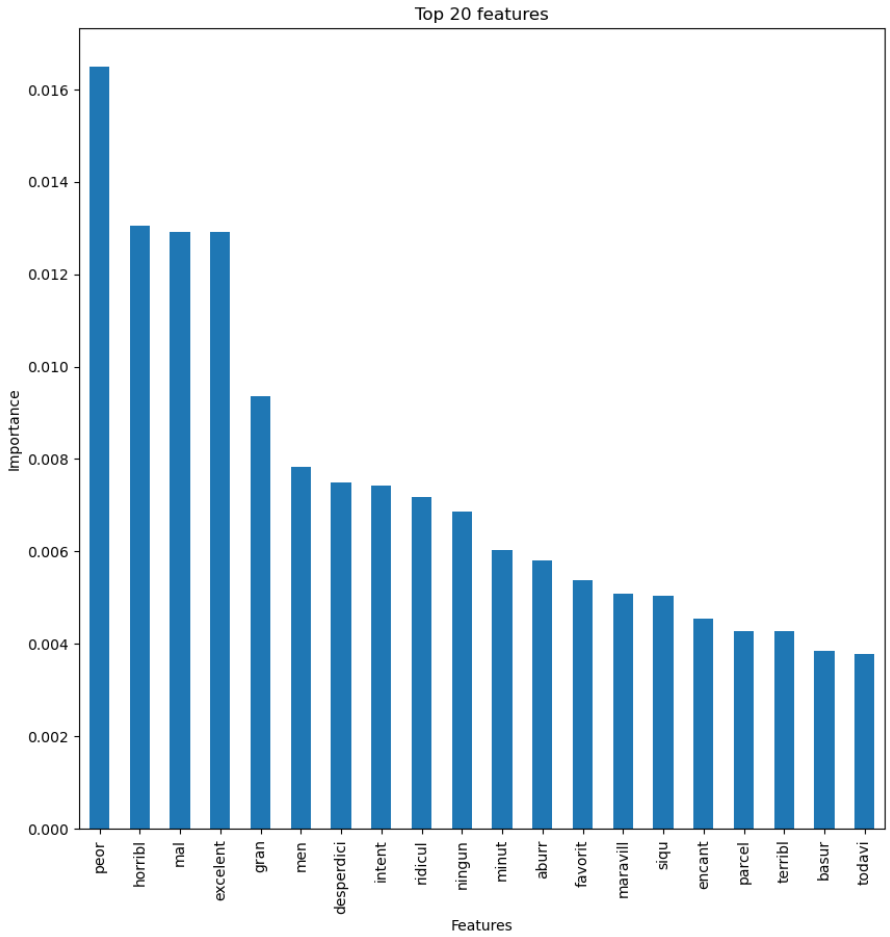
- Se buscan los mejores hiperparametros, utilizando GridSearchCV, para usar en el modelo. De los muchos resultados obtenido en este apartado se sacan 3 combinaciones y se evalúa, obteniendo lo siguiente:

```
MAX DEPTH: 20 / # OF EST: 100 -- A: 0.806 / P: 0.845 / R: 0.757
MAX DEPTH: None / # OF EST: 100 -- A: 0.796 / P: 0.824 / R: 0.76
MAX DEPTH: None / # OF EST: 5 -- A: 0.651 / P: 0.657 / R: 0.653
```

- Al identificar como mejor hiperparametro 20 y 100 usamos este como parámetros finales para el modelo, ya que nos mejora la respuesta de predicción.

```
MAX DEPTH: 20 / # OF EST: 100 -- A: 0.806 / P: 0.845 / R: 0.757 / F1: 0.799
```

Con este modelo, aunque no sea el más alto podemos identificar varias cosas. En primera instancia, los hiperparametros ayudan a encontrar la mejor combinación para el modelo y así mejorar su rendimiento. Por último, este modelo nos ayuda a identificar las palabras más significativas. La mayoría de ellas tienen son adjetivos positivos y negativos, lo que tiene sentido con el problema.



5. Resultados

Modelo	Accuracy	Recall	Precision	F1 Score
Naive Bayes	**0.82**	**0.82**	**0.82**	**0.82**
Neural Network	**0.841**	**0.841**	**0.841**	**0.841**
Random Forest	**0.806**	**0.757**	**0.845**	**0.799**

En cuanto a los resultados obtenidos es posible observar que el modelo con un rendimiento más alto es la red neuronal, con un 84.1%, siendo el más alto de los 3. Adicionalmente, en cuanto a las otras métricas, es posible observar una consistencia en estas lo cual lleva a un F1 score similar llevando a un resultado esperado.

Por otro lado, es posible considerar nuestro modelo como efectivo en la tarea de clasificación y de gran ayuda para el análisis de sentimiento de opiniones, ya que logra superar el desempeño de los otros dos modelos evaluados. Sin embargo, es importante tener en cuenta que siempre hay margen de mejora y se podrían explorar diferentes técnicas y enfoques para lograr un mejor desempeño en la tarea en cuestión. Además, también se debe considerar la interpretación de los resultados y su aplicación en situaciones donde la cantidad de datos pueda llegar a ser más extensa.

En cuanto a nuestros demás modelos, estos pueden llegar a ser de gran utilidad para el análisis textual en otros contextos. Por un lado, RandomForest nos permite saber la importancia de las palabras en los textos analizados, siendo esto de gran ayuda en caso de que se quiera identificar todas estas palabras que tienen más influencia sobre si una opinión es negativa o positiva. Por otro lado, Naive Bayes permite un acercamiento más eficiente y flexible en cuanto a el nivel presente en las palabras.

En cuanto a las decisiones a considerar por la organización, se pueden considerar diversas opciones. Por ejemplo, la organización podría utilizar el modelo de red neuronal para analizar las opiniones con el fin de clasificar estas y agrupar información en base a esto. Asimismo, La organización podría utilizar el modelo RandomForest para identificar las palabras clave y las características más importantes en las revisiones analizadas, lo cual podría ser útil para mejorar la estrategia de marketing y publicidad de la empresa, o incluso ver las tendencias presentes en las opiniones.

En conclusión, los tres modelos presentados ofrecen distintas ventajas y pueden ser útiles en diferentes contextos. En el caso específico abordado en esta entrega, el modelo de red neuronal ha demostrado un mejor desempeño en la tarea evaluada en comparación con los otros modelos considerados.

6. Trabajo en equipo

a. Roles desempeñados en el equipo

Rol	Nombre	Código	Usuario Github
Líder de proyecto	Daniel Aguilera	202010592	Daniagui12
Líder de negocio	Laura Martinez	202012624	lmartinezp2003
Líder de datos	Cristian Sánchez	202022112	Panis26
Líder de analítica	Daniel Aguilera	202010592	Daniagui12

b. Distribución del trabajo

Tema	Nombre
Entendimiento del negocio y enfoque analítico	Todos los integrantes
Entendimiento y preparación de los datos	Todos los integrantes
Modelado y evaluación (Modelo Naive Bayes)	Laura Martinez
Modelado y evaluación (Modelo Neural Network)	Daniel Aguilera
Modelado y evaluación (Modelo Random Forest)	Cristian Sanchez
Resultados	Todos los integrantes

c. Coevaluación

Los 100 puntos para repartir serian repartidos de tal forma que los 3 estudiantes cuenten con la misma cantidad de puntos. Esto debido a que la

cantidad de trabajo y la calidad de este fue relativamente similar, mostrando que todos trabajamos por igual y presentamos un buen trabajo de forma consistente. Esto diciendo que cada uno tendría alrededor de 33.3 puntos.

Puntos por mejorar:

- Comunicación: Es necesario para la siguiente etapa tener una comunicación mas efectiva para así evitar confusiones y que nuestra calidad de trabajo sea mucho mejor
- Establecer objetivos: Es crucial mejorar el establecimiento de objetivos para trabajar de forma constante y que pueda haber revisiones más seguidas.
- Tiempo de trabajo: Es importante iniciar a trabajar con mayor anterioridad para así evitar días de trabajo pocos días antes de la entrega.

Referencias

https://www.kaggle.com/code/onadegibert/sentiment-analysis-with-tfidf-and-random-forest

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f

https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

https://www.kaggle.com/code/prashant111/naive-bayes-classifier-in-python