Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ingeniería Eléctrica (DIE)

Procesamiento del Lenguaje Natural (2950)

Semestre 2023-2

Programa 01. Naive Bayes, Árbol de Decisión, SVM con Bag of Words y TF-IDF

Nombre de los Alumnos:	Basile Álvarez Andrés José Keller Ascencio Rodolfo Andrés		
Números de Cuenta:	316617187 316515746		
Correos Electrónicos:	andresbasile123@gmail.com rodolfoa.kellerascencio@gmail.com		
Grupo:	1	**Profesor:** Ing. Emilio Alejandro Morales Juárez	**Calificación:**
Semestre:	2023-2	**Fecha de Entrega:** 07. Abril. 2023	

Contenido

Introducción

A lo largo de esta práctica se estará trabajando con las técnicas de soporte vectorial Bolsa de Palabras (BOW, Bag Of Words) y TF-IDF, así como con los algoritmos de aprendizaje máquina Naive Bayes, Árboles de Decisión y Máquinas de Soporte Vectorial (SVM, Support Vector Machines).

Para esto, se utilizó un conjunto de datos de entrenamiento y prueba relacionado con la reseña de películas, siendo estas reseñas positivas y negativas, donde fue necesario trabajar con estos conjuntos de datos de entrenamiento y prueba que contaban con doce mil quinientos registros de reseñas positivas y doce mil quinientos registros de reseñas negativas, tanto para datos de entrenamiento como para realización de pruebas, es decir, se trabajó con un total de cincuenta mil registros de reseñas que fueron divididas de forma equitativa al momento de obtener un conjunto de entrenamiento de veinticinco mil registros, y un registro de prueba de veinticinco mil registros.

Para reconocer las características de las diferentes técnicas de soporte vectorial y de los algoritmos de aprendizaje máquina fue necesario construir y ajustar los modelos de clasificación, los cuales a su vez contaban con diferentes características al momento de hacer uso de las reseñas, pues además se trabajó con técnicas de recuperación de datos en los sistemas de información como lo es el *Stemming* y la lematización, de esta forma pudimos evaluar la precisión y eficiencia de los algoritmos para realizar una comparativa y reconocer el tipo de algoritmo que mejor se ajustaría a nuestro conjunto de datos, el cual nos proporcionara una mejor clasificación de las reseñas.

El procedimiento que se siguió en la práctica fue, inicialmente, la obtención de los conjuntos de datos y el preprocesamiento de los mismos, trabajando con la selección de características y las distintas técnicas de recuperación de datos así como la reducción de nuestro conjunto a través de la eliminación de *stopwords*. Posteriormente se buscó definir la técnica de soporte vectorial a utilizar, al igual que el algoritmo de lenguaje máquina, con lo cual se prepararon los distintos modelos para realizar su comparación a través de la medición de tiempos de entrenamiento y clasificación, al igual que reconociendo la precisión y exactitud de los modelos a través de reportes de clasificación y matrices de confusión.

Metodología

En este apartado se describirán brevemente la mayoría de los métodos utilizados a lo largo de la realización de este programa.

Bag of Words

Es un algoritmo que utiliza la frecuencia de palabras, es decir, donde cada palabra se relaciona con su número de apariciones en el conjunto de entrenamiento para poder reconocer la probabilidad de aparición de la palabra en un texto. De esta forma, para poder realizar la clasificación de un texto, o en este caso de las reseñas de las películas, el algoritmo contabiliza cada palabra del texto que se desea clasificar y predice dentro de su modelo, a partir de una matriz de frecuencia, qué tipo de clasificación sería la más adecuada y relevante para el texto.

En este sentido, este algoritmo se puede utilizar cuando se necesite clasificar el texto en función de la frecuencia de aparición de determinadas palabras. (IBM, 2022)

Figura 1. Representación gráfica de la Bolsa de Palabras. (Kumar, 2021)

TF-IDF

Este es un algoritmo o método indicador de peso semántico que realiza la asignación de peso de una palabra como el producto entre dos medias, la frecuencia de término y la frecuencia invertida de documento. IDF es la proporción inversa entre el número de documentos totales (en este caso reseñas) y el número de documentos (o reseñas) en donde la palabra aparece al menos una vez.

$$W_{ij} = f_{ij} * \log(N/n_i)$$

Donde N es el número de documentos totales y ni es el número de documentos en donde la palabra ocurre al menos una vez. (Cifuentes, 2016, P. 11)

Con el objetivo de determinar el nivel de presencia de la temática analizada dentro de las reseñas, se hace uso del cálculo del factor TF y del factor IDF, siendo un método general de listas de palabras clave con una calificación o peso que indica que tan relevante es la palabra con respecto a la reseña seleccionada y al corpus en general. De esta manera es posible establecer una ponderación de cada término del árbol de pertenencias en cada reseña o documento de la colección. (Vuotto, 2015, p.8)

Stemming

El *Stemming* es una técnica de normalización de texto utilizada en el procesamiento de lenguaje natural para reducir las palabras a su raíz. Esta técnica elimina los afijos de las palabras, lo cual puede llegar a generar palabras no válidas o existentes, pero reducidas en longitud, agrupando, en general, a palabras que pertenecen a la misma familia.

Por lo general, el *stemming* es utilizado en la búsqueda de información, donde las palabras reducidas a su raíz se utilizan como sinónimos para aumentar los criterios de búsqueda, o también son utilizadas para la reducción de la dimensionalidad, ayudando a reducir la cantidad de palabras que se rastrearán y utilizarán en la generación de modelos con algoritmos de Machine Learning. (MatLab, 2023)

Lematización

La lematización es otra técnica de normalización de texto utilizada en el procesamiento del lenguaje natural. Esta técnica utiliza tanto vocabulario como análisis morfológico, donde busca devolver palabras con el formato que aparecen en el diccionario. (MatLab, 2023)

Palabra	Lematización	Stemming
Pensando	Pensar	Pensa
Pensado	Pensar	Pensa
Pensamiento	Pensamiento	Pensa

Figura 2. Diferencias entre *Stemming* y lematización. (MatLab, 2023)

Naive Bayes

Naive Bayes es uno de los algoritmos de aprendizaje automático más eficientes y efectivos, siendo un algoritmo clasificador probabilístico simple con fuerte suposición de independencia, debido a que simplifica considerablemente el aprendizaje mediante el supuesto de independencia de los atributos. Este algoritmo se considera fácil de construir y no necesita de esquemas iterativos de estimación de parámetros, lo cual implica que puede ser aplicado a grandes bases de datos. (Mosquera, 2018)

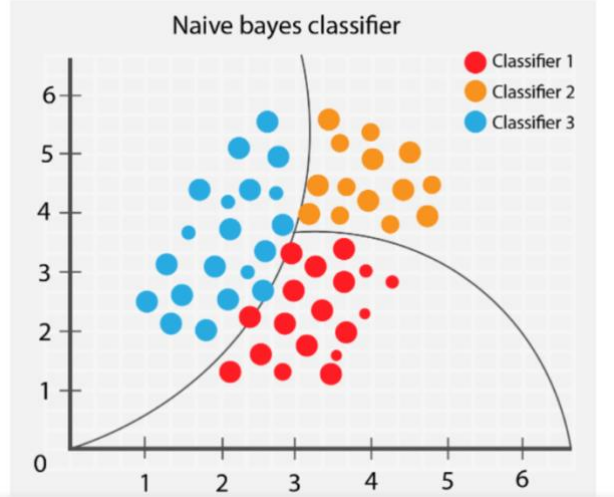


Figura 3. Clasificador Naive Bayes. (Chaudhuri, 2022)

Árbol de Decisión

El algoritmo de árboles de decisión es otro algoritmo de aprendizaje basado en similaridades, donde se siguen considerando como uno de los algoritmos más sencillos y fáciles de implementar, siendo a su vez bastante precisos y poderosos. Este algoritmo genera un árbol de decisión de manera recursiva al considerar el criterio de la mayor proporción de ganancia de la información conocido también como *gain ratio*, siendo este el atributo de los datos que mejor los clasifica. (Valero, 2005)

En este sentido, un árbol de decisión es un modelo de predicción que nos permite representar y categorizar una serie de condiciones que suceden de forma sucesiva para la solución de un problema, donde dicho conocimiento se representa mediante un árbol que cuenta con un conjunto de nodos, hojas y ramas. (Barrientos, 2009, p.20)

Dentro del nodo principal se representa a toda la población de nuestros datos; en el nodo de decisión encontramos una división de los datos a partir de una condición; en el nodo hoja podemos encontrar la decisión final de la clasificación.

Una desventaja y problema común de los árboles de decisión es el sobreajuste o sobreaprendizaje, el cual se da cuando se genera un árbol que cubra todos los posibles casos de los datos y que ya no permita generar correctamente nuevas predicciones, por lo que para evitar estos problemas es necesario definir restricciones sobre el tamaño o profundidad del árbol o podarlo. (Molero, 2022)

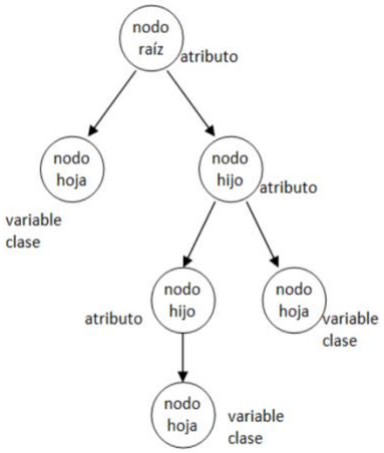


Figura 4. Estructura de un árbol de decisión. (Barrientos, 2009, p.20)

Máquina de Soporte Vectorial

Inicialmente, las máquinas de soporte vectorial fueron diseñadas para tratar con problemas binarios, sin embargo, actualmente se pueden utilizar para múltiples propósitos. Este es un algoritmo que toma los datos como entrada y genera una línea que separa a estos datos en dos clases, donde a esta línea se le conoce como hiperplano y busca la separación óptima de las clases a través de la mayor distancia entre vectores de soporte de las clases.

Para encontrar la máxima separación de las clases es posible generar infinitas líneas o hiperplanos, sin embargo, el separador de margen máximo se determina por un subconjunto de datos conocidos como vectores de soporte, pues entre más lejanos se encuentren dichos vectores de soporte, mayor será lña probabilidad de clasificar correctamente las clases.

Las máquinas de soporte vectorial pueden utilizarse tanto para conjuntos de datos linealmente separables como para los que no son linealmente separables, e inclusive se puede modificar la fórmula del kernel para trabajar con transformaciones matemáticas que mapeen de mejor forma los datos.

Este algoritmo suele ser mucho más complejo que los descritos anteriormente, debido a que a mayor cantidad de elementos, mayor suele ser su complejidad, siendo que una de las mayores desventajas para este algoritmo es que al trabajar con conjuntos grandes de datos el tiempo de procesamiento, entrenamiento y predicción suele ser mucho mayor, siendo poco efectivo, pese a ser uno de los algoritmos que mejor separación y precisión le puede dar a los datos al trabajar con un kernel correcto. (Molero, 2022)

Experimentos y Resultados

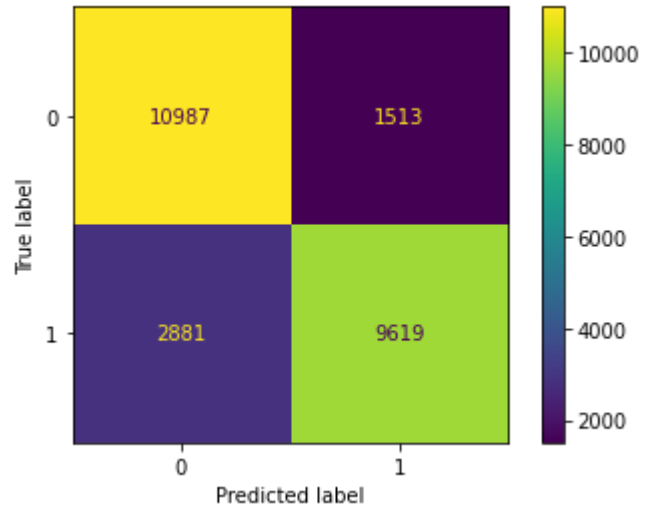
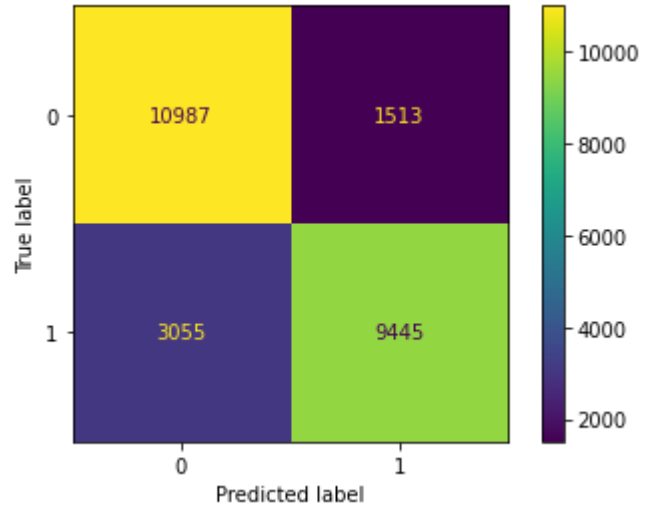
En este apartado se busca crear tablas que resuman los resultados obtenidos a través de nuestro código, mostrando métricas, tiempos de entrenamiento y evaluación.

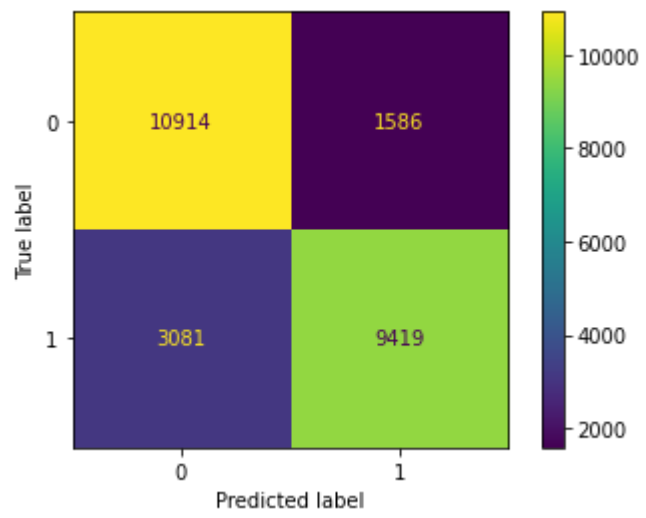
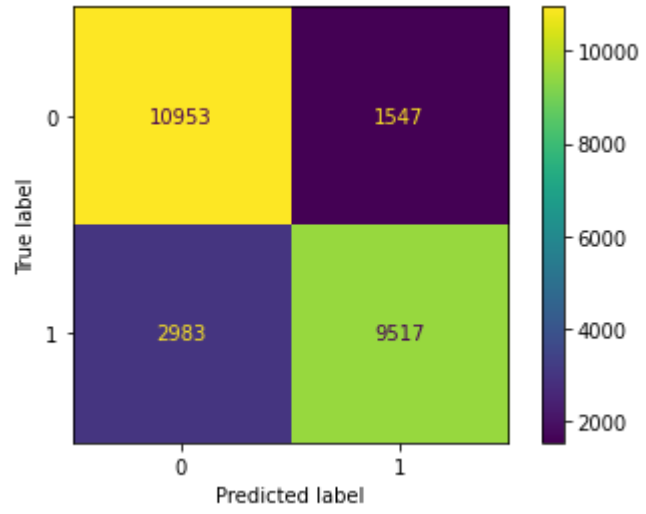
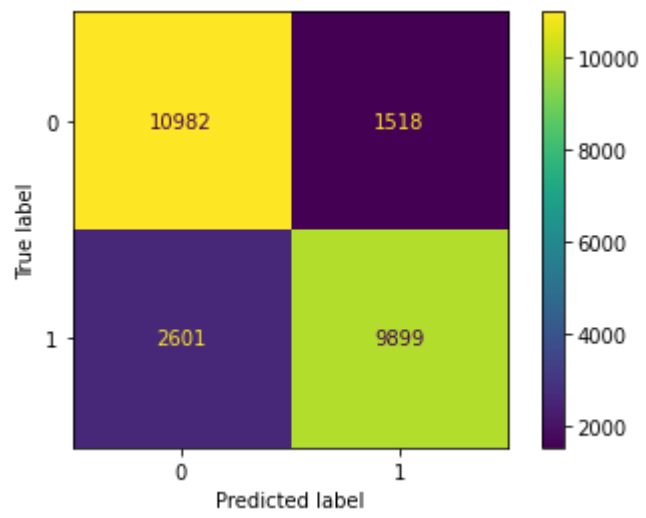
Naive Bayes tiempos y precisión

Tipo de modelo	Train Time	Predict Time	Precision
NB Bow S/Stw	0.026520	0.025522	0.864086
NB Bow C/Stw	0.036520	0.034516	0.861927
NB Bow Stemming S/Stw	0.027949	0.024989	0.855884
NB Bow Lematización S/Stw	0.024530	0.026440	0.860177
NB TF-IDF S/Stw	0.039560	0.019537	0.867040
NB TF-IDF C/Stw	0.039013	0.025968	0.879223
NB TF-IDF Stemming S/Stw	0.029557	0.017476	0.862632
NB TF-IDF Lematización S/Stw	0.034662	0.019475	0.866915
Promedio	**0.032288875**	**0.024240375**	**0.8647355**

Tabla 1. Resultados obtenidos de tiempos de entrenamiento, tiempos de predicción y precisión de algoritmo Naive Bayes.

Naive Bayes Matrices de confusión

Tipo de modelo	Matriz de confusión
NB Bow S/Stw	
NB Bow C/Stw	

NB Bow Stemming S/Stw	
NB Bow Lematización S/Stw	
NB TF-IDF S/Stw	

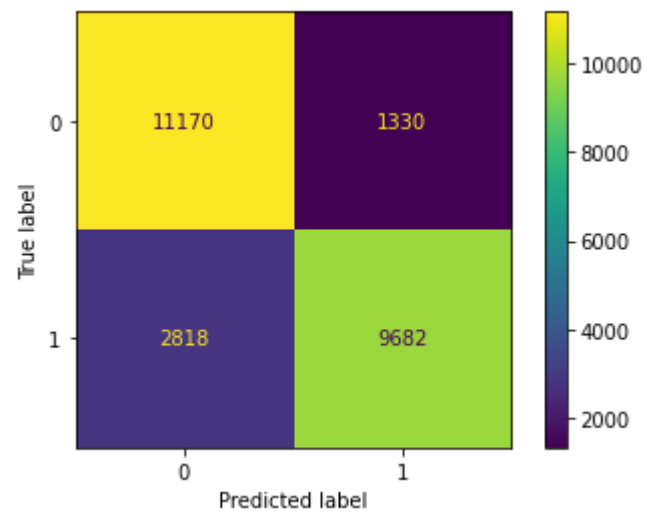
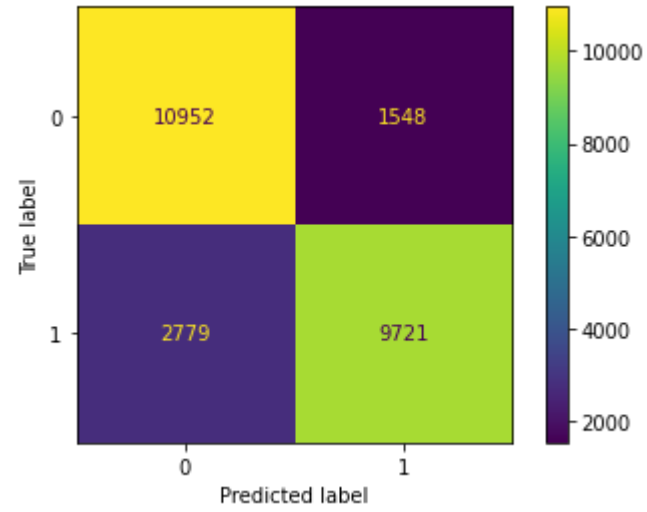
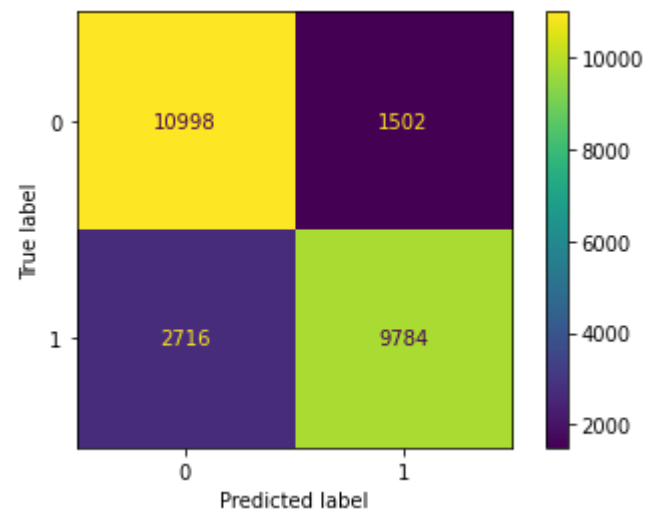
NB TF-IDF C/Stw	
NB TF-IDF Stemming S/Stw	
NB TF-IDF Lematización S/Stw	

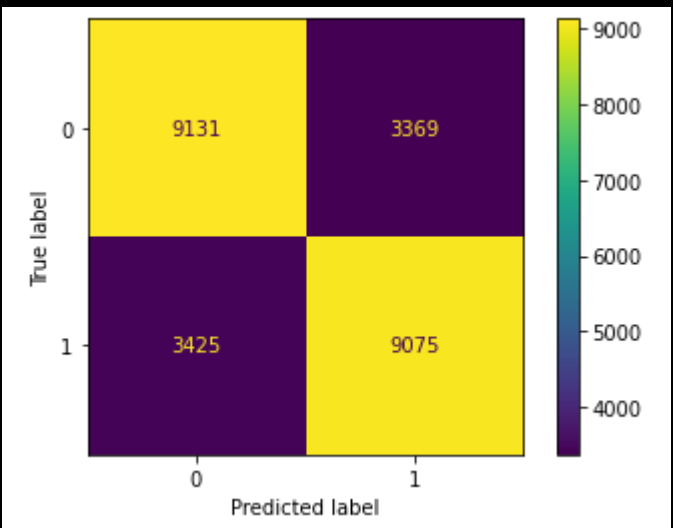
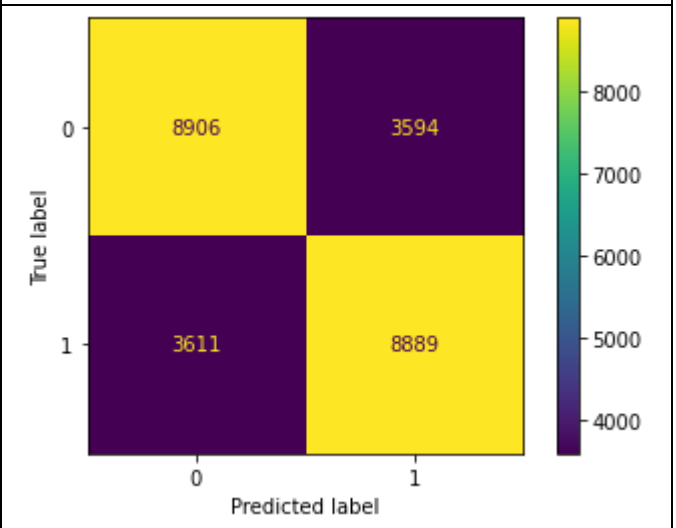
Tabla 2. Resultados obtenidos de matrices de confusión de algoritmo Naive Bayes.

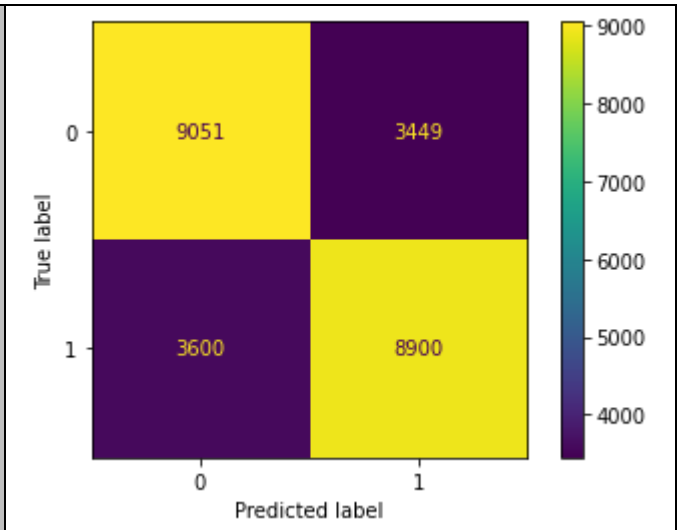
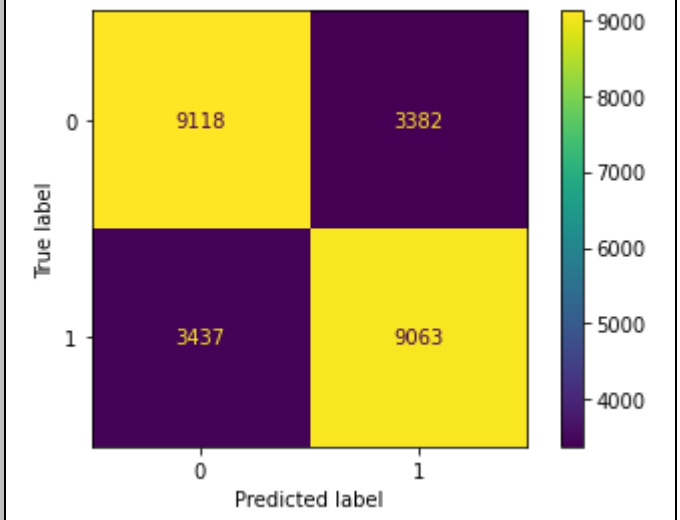
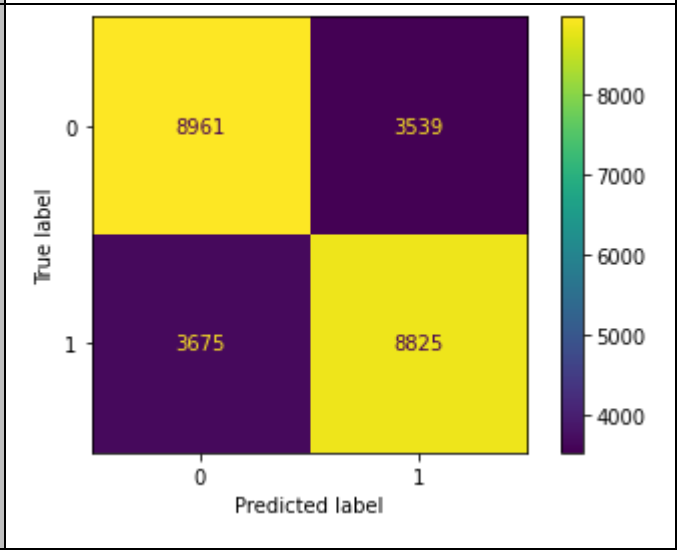
Árbol de Decisión tiempos y precisión

Tipo de modelo	Train Time	Predict Time	Precision
Árbol Bow S/Stw	28.542007	0.041974	0.729267
Árbol Bow C/Stw	33.369805	0.052446	0.712088
Árbol Bow Stemming S/Stw	23.932797	0.042481	0.720706
Árbol Bow Lematización S/Stw	25.144222	0.034993	0.728244
Árbol TF-IDF S/Stw	36.193511	0.081888	0.713766
Árbol TF-IDF C/Stw	34.119844	0.103303	0.695968
Árbol TF-IDF Stemming S/Stw	27.582786	0.096496	0.716360
Árbol TF-IDF Lematización S/Stw	28.504585	0.099866	0.716039
Promedio	**29.83536429**	**0.073067571**	**0.714738714**

Tabla 3. Resultados obtenidos de tiempos de entrenamiento, tiempos de predicción y precisión de algoritmo Árbol de Decisión.

Árboles de Decisión matrices de confusión

Tipo de modelo	Matriz de confusión
Árbol Bow S/Stw	
Árbol Bow C/Stw	

Árbol Bow Stemming S/Stw	
Árbol Bow Lematización S/Stw	
Árbol TF-IDF S/Stw	

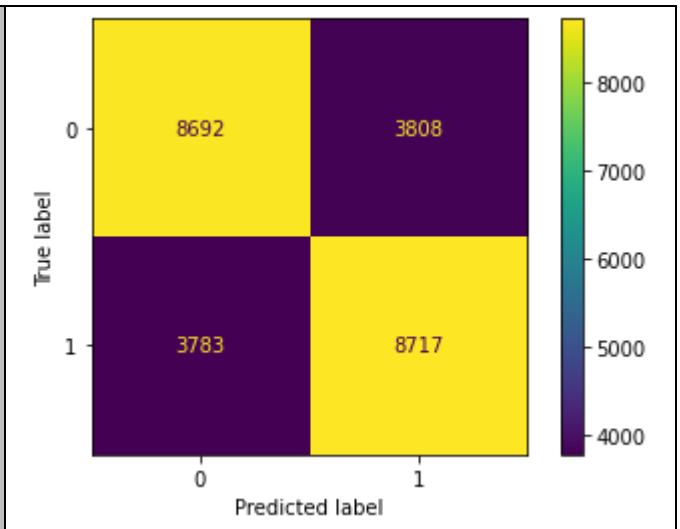
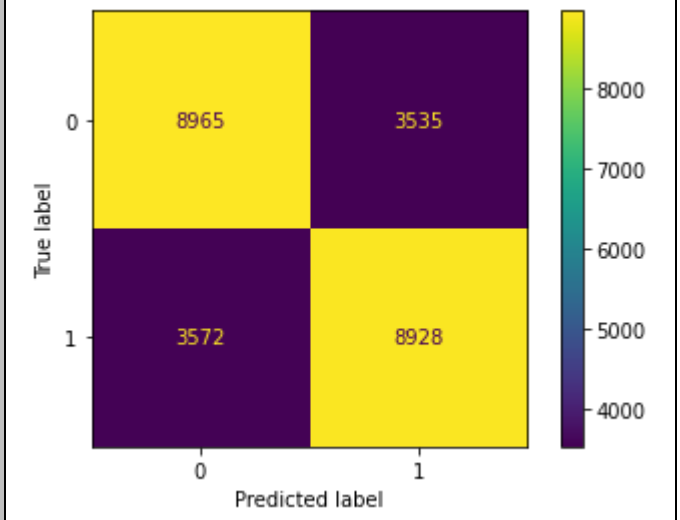
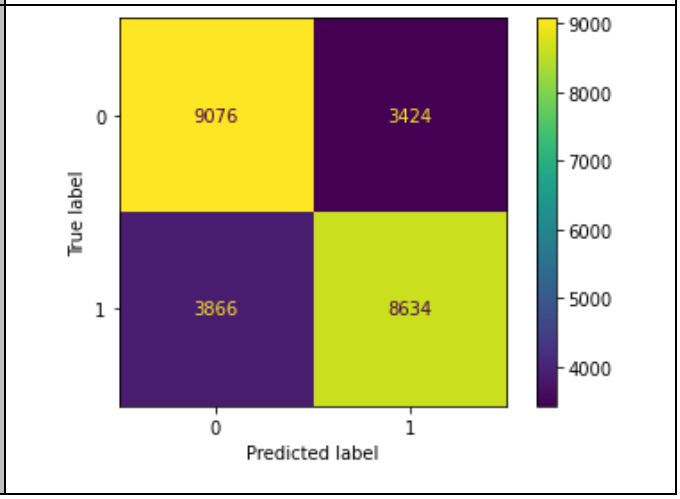
Árbol TF-IDF C/Stw	
Árbol TF-IDF Stemming S/Stw	
Árbol TF-IDF Lematización S/Stw	

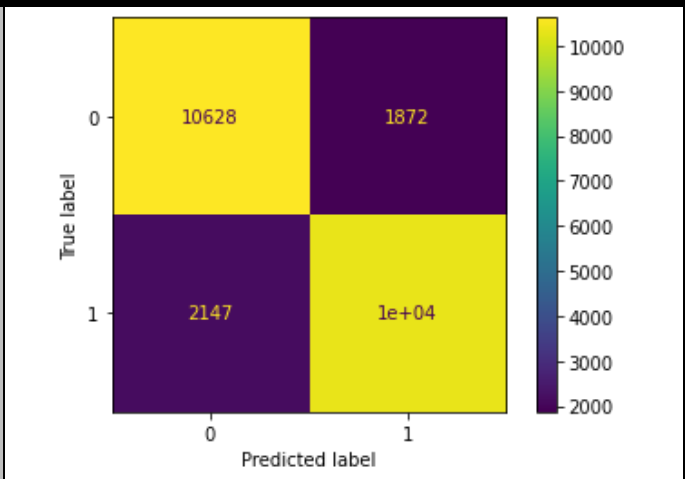
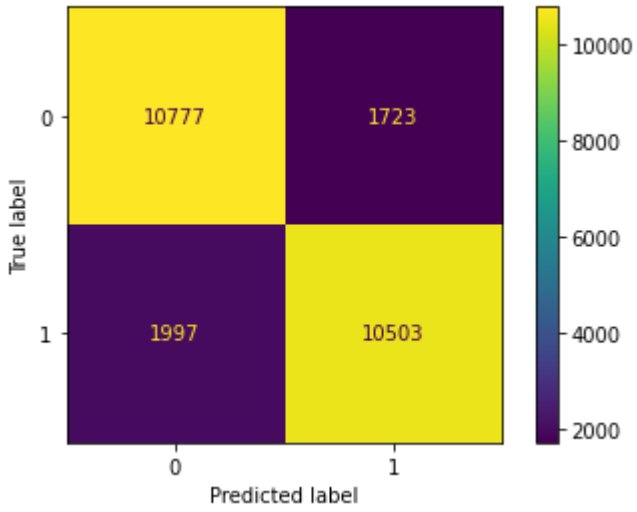
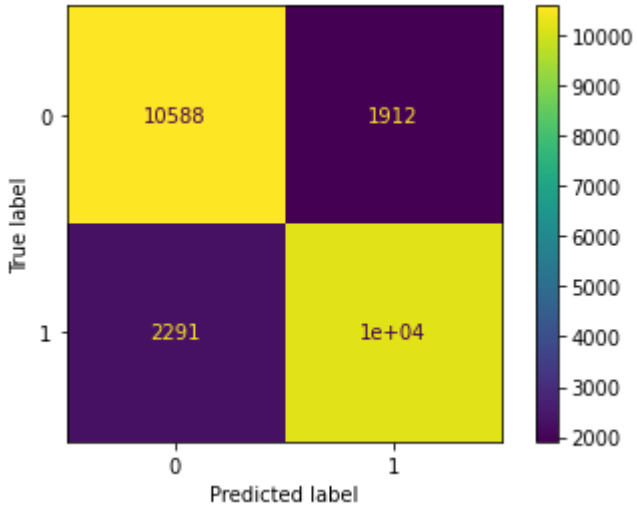
Tabla 4. Resultados obtenidos de matrices de confusión de algoritmo de Árbol de Decisión.

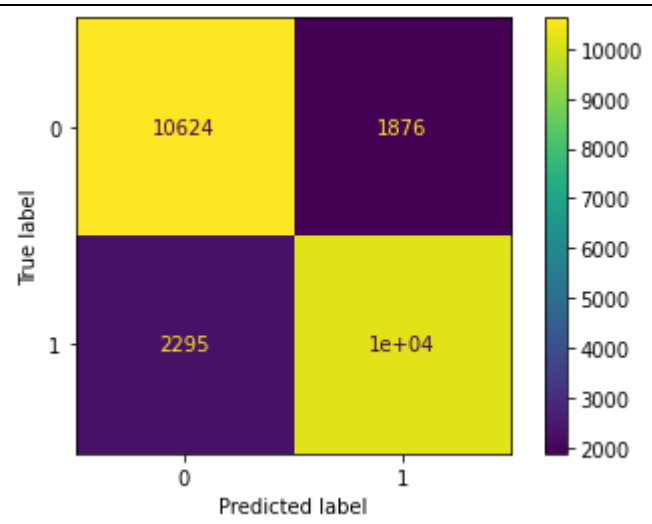
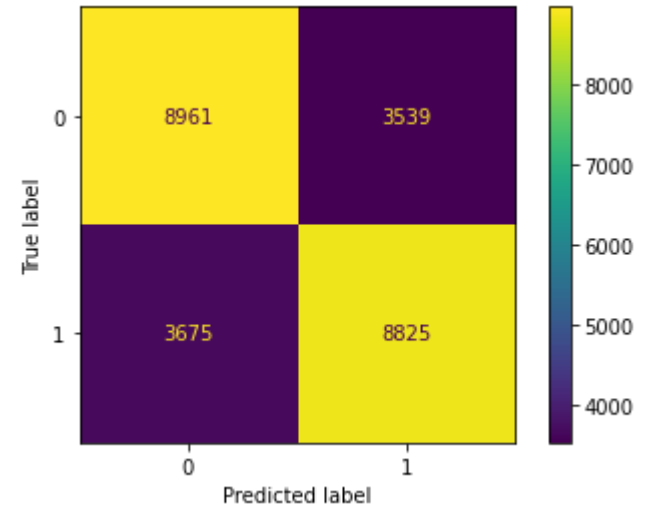
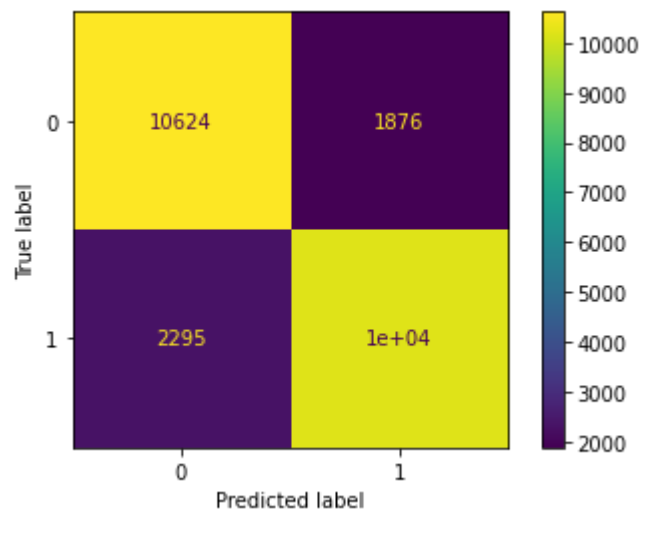
Máquinas de Soporte Vectorial tiempos y precisión

Tipo de modelo	Train Time	Predict Time	Precision
SVM Bow S/Stw	1083.498923	129.170357	0.83924
SVM Bow C/Stw	2405.155325	201.661782	0.85120
SVM Bow Stemming S/Stw	1245.849867	123.066634	0.83188
SVM Bow Lematización S/Stw	1085.880910	119.127320	0.83316
SVM TF-IDF S/Stw	1080.616822	118.847229	0.83316
SVM TF-IDF C/Stw	1051.827193	116.980485	0.83316
SVM TF-IDF Stemming S/Stw	1111.696929	128.914931	0.83316
SVM TF-IDF Lematización S/Stw	1110.531911	143.662559	0.83316
Promedio	**1271.882235**	**135.1789121**	**0.836015**

Tabla 5. Resultados obtenidos de tiempos de entrenamiento, tiempos de predicción y precisión de algoritmo Máquinas de Soporte Vectorial.

Máquinas de Soporte Vectorial matrices de confusión

Tipo de modelo	Matriz de confusión
SVM Bow S/Stw	
SVM Bow C/Stw	
SVM Bow Stemming S/Stw	

SVM Bow Lematización S/Stw	
SVM TF-IDF S/Stw	
SVM TF-IDF C/Stw	

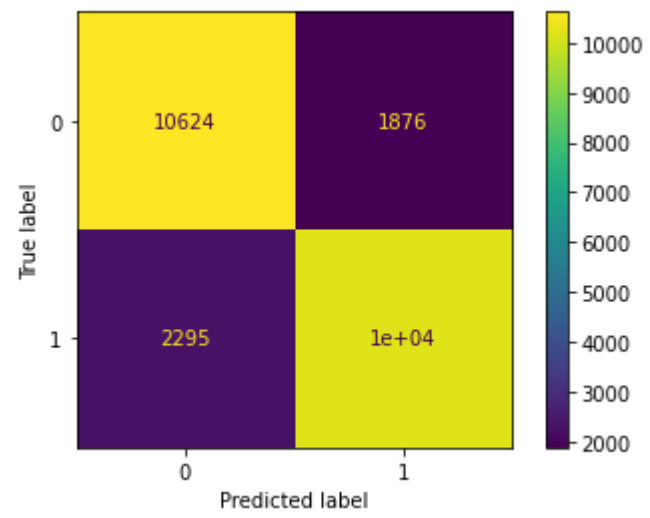
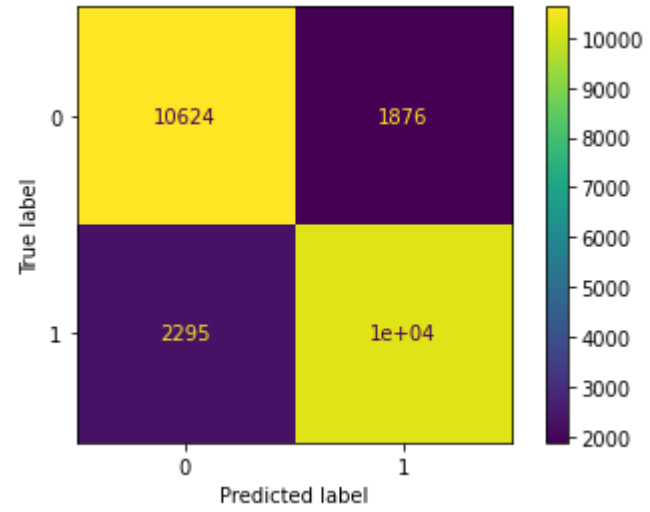
SVM TF-IDF Stemming S/Stw	
SVM TF-IDF Lematización S/Stw	

Tabla 5. Resultados obtenidos de matrices de confusión de algoritmo Máquinas de Soporte Vectorial.

Resultados obtenidos por el programa

	Tipo de modelo	Train Time	Predict Time	Precision
0	NB Bow S/Stw	0.026520	0.025522	0.864086
1	NB Bow C/Stw	0.036520	0.034516	0.861927
2	NB Bow Stemming S/Stw	0.027949	0.024989	0.855884
3	NB Bow Lematización S/Stw	0.024530	0.026440	0.860177
4	NB TF-IDF S/Stw	0.039560	0.019537	0.867040
5	NB TF-IDF C/Stw	0.039013	0.025968	0.879223
6	NB TF-IDF Stemming S/Stw	0.029557	0.017476	0.862632
7	NB TF-IDF Lematización S/Stw	0.034662	0.019475	0.866915
0	Arbol Bow S/Stw	28.542007	0.041974	0.729267
1	Arbol Bow C/Stw	33.369805	0.052446	0.712088
2	Arbol Bow Stemming S/Stw	23.932797	0.042481	0.720706
3	Arbol Bow Lematización S/Stw	25.144222	0.034993	0.728244
4	Arbol TF-IDF S/Stw	36.193511	0.081888	0.713766
5	Arbol TF-IDF C/Stw	34.119844	0.103303	0.695968
6	Arbol TF-IDF Stemming S/Stw	27.582786	0.096496	0.716360
7	Arbol TF-IDF Lematización S/Stw	28.504585	0.099866	0.716039
0	SVM Bow S/Stw	1083.498923	129.170357	0.83924
1	SVM Bow C/Stw	2405.155325	201.661782	0.85120
2	SVM Bow Stemming S/Stw	1245.849867	123.066634	0.83188
3	SVM Bow Lematización S/Stw	1085.880910	119.127320	0.83316
4	SVM TF-IDF S/Stw	1080.616822	118.847229	0.83316
5	SVM TF-IDF C/Stw	1051.827193	116.980485	0.83316
6	SVM TF-IDF Stemming S/Stw	1111.696929	128.914931	0.83316
7	SVM TF-IDF Lematización S/Stw	1110.531911	143.662559	0.83316

Figura 5. Resultados obtenidos en la ejecución de nuestro programa

Conclusiones

Andrés José Basile Álvarez

Durante la realización de este ejercicio, trabajamos con representaciones vectoriales "BOW" y "TF-IDF", aplicando varios algoritmos de clasificación para, a partir de un conjunto de datos de reseñas de películas, obtener su respectiva clasificación dentro de las categorías: "positiva" o "negativa".

Con tal objetivo en mente, realizamos distintas variaciones en la aplicación de los algoritmos: incluyendo "stopwords", eliminando "stopwords", realizando "stemming" y eliminando "stopwords y realizando "lematización" y eliminando "stopwords". A partir de estos modelos, observamos que los que tuvieron mayor precisión fueron los que utilizaron el algoritmo de Naive Bayes y la Máquina de Soporte Vectorial, puntuando en alrededor de 86% y 83% de precisión, respectivamente. No obstante, el gran aumento de tiempo necesario para entrenar y predecir nuevas clasificaciones utilizando la máquina de soporte vectorial, comparado con el algoritmo de Naive Bayes, hace que éste último sea mucho más adecuado para esta aplicación.

Se había pensado que para futuras pruebas sería recomendable analizar la posibilidad de trabajar con el conjunto de datos completo debido a todos los problemas para la realización del ejercicio gracias al excesivo tiempo de entrenamiento y ejecución, por lo que en un inicio se decidió únicamente utilizar el conjunto de prueba donde se dividió este conjunto en dos subconjuntos, uno para entrenamiento conteniendo el 75% de los datos y otro de prueba conteniendo el 25 % de los datos, sin embargo, en esta nueva etapa ya se realizaron las pruebas con los conjuntos completos de datos, aumentando la precisión del algoritmo Naive Bayes, al igual que aumentando considerablemente los tiempos de entrenamiento y prueba para las Máquinas de Soporte Vectorial.

Rodolfo Andrés Keller Ascencio

A lo largo de este ejercicio se estuvo trabajando con diferentes técnicas de representación vectorial, tanto Bag Of Words (bolsa de palabras), como TF-IDF, así como con tres algoritmos de aprendizaje máquina, siendo estos el Naive Bayes, Árboles

de Decisión y la Máquina de Soporte Vectorial (SVM). De misma manera, se trabajó con dos conjuntos de datos, uno de entrenamiento y otro de prueba, donde estos datos hacían referencia a la reseña de películas, de tal forma que se pueda realizar una clasificación con respecto a si una reseña es positiva o negativa, a partir de las palabras contenidas dentro de la misma.

Para esto, se realizaron ocho diferentes pruebas con cada uno de los tres diferentes algoritmos de aprendizaje máquina, donde se trabajó con BOW y TF-IDF, analizando la diferencia entre estos dos, así como la diferencia del uso de procedimientos como lo es la eliminación de stopwords, el uso de stemming y el uso de lematización, teniendo como resultado 24 distintos modelos a analizar.

Como conclusión, a partir de la tabla de resultados obtenida se aprecia cómo los algoritmos de Naive Bayes y SVM fueron los más precisos, siendo el Naive Bayes TF-IDF el que obtuvo una mayor precisión al momento de realizar la clasificación de las reseñas, teniendo un 87% de precisión. El uso del algoritmo de árboles de decisión resultó ser el menos preciso con un 71% promedio.

Por otro lado, con respecto a los tiempos de entrenamiento y predicción la SVM resultó ser el algoritmo de mayor tiempo, con un tiempo promedio de entrenamiento de veintiún minutos frente a los nueve minutos veinte segundos que habíamos tenido al trabajar con conjuntos más pequeños de datos y un tiempo de predicción promedio de dos minutos quince segundos frente a los treinta y dos segundos obtenidos al trabajar con un conjunto de datos reducido, siendo sumamente mayor con respecto al Naive Bayes y a los Árboles de decisión. Al hacer uso de Naive Bayes se tuvo un tiempo de entrenamiento promedio de menos de 0.03 segundos y un tiempo de predicción promedio menor a 0.02 segundos.

A partir de estos resultados podemos concluir que el mejor tipo de algoritmo y procedimiento para la clasificación de reseñas positivas y negativas de películas sería el Naive Bayes TF-IDF, donde no hubo mucha diferencia con respecto al uso de stopwords, o el procedimiento de stemming o lematización, por lo cual deberíamos realizar un estudio más profundo con respecto a las implicaciones que traería el uso de estos

procedimientos para la clasificación de las reseñas y de esta manera, seleccionar el modelo que mejor se ajuste a nuestras necesidades.

A manera de observación, en un inicio se consideró que en el futuro realizáramos la selección de otro algoritmo de aprendizaje de máquina distinto a la Máquina de Soporte Vectorial para que, de esta manera, podamos hacer uso del conjunto completo de datos de entrenamiento y prueba, puesto que al tener tantos problemas para la ejecución del código se tomó la decisión de trabajar únicamente con la mitad de los datos, haciendo uso del conjunto de entrenamiento original y dividiendo este conjunto de entrenamiento en dos subconjuntos, donde un 75% de datos se usó para el entrenamiento y un 25% de datos para la prueba, resultando que aún así los tiempos de ejecución fueron excesivos para la realización de la práctica, sin embargo, ya contábamos con un mismo parámetro de comparación para todos los modelos. Pese a esta idea inicial, en un segundo intento se trabajó con el conjunto completo de datos y pese a haber obtenido tiempos de entrenamiento y prueba excesivos, después de largas horas de trabajo tanto por parte de nosotros como de la computadora, logramos obtener una tabla de resultados que nos permite realizar un análisis interesante de este primer programa.

Referencias

1. IBM. (15 de noviembre del 2022). Algoritmos de clasificación de textos. IBM. Sitio web. Recuperado de: https://www.ibm.com/docs/es/rpa/21.0?topic=classification-text-algorithms. Consultado el 3 de abril del 2022.
2. Kumar, L. (24 de enero del 2021). NLP: Bag of words and TF-IDF explained!. Linkedin. Sitio web. Recuperado de: https://www.linkedin.com/pulse/nlp-bag-words-tf-idf-explained-l-koushik-kumar. Consultado el 3 de abril del 2022.
3. MatLab. (2023). Reducción de palabras a su raíz. MatLab. Sitio web. Recuperado de: https://la.mathworks.com/discovery/stemming.html. Consultado el 3 de abril del 2022.
4. Cifuentes. (2016). Clasificación automática de Tweets utilizando K-NN y K-Means como algoritmos de clasificación automática, aplicando TF-IDF y TF-RFL para las

ponderaciones. Documento web. Pontificia Universidad Católica de Valparaíso. Recuperado de: http://opac.pucv.cl/pucv_txt/txt-8500/UCD8528_01.pdf. Consultado el 6 de abril del 2022.

5. Vuotto, A. (2015). Aplicación del factor TF-IDF en el análisis semántico de una colección documental. Documento web. Universidad Nacional de Mar de Plata. Argentina. Recuperado de: https://www.redalyc.org/pdf/161/16143063001.pdf. Consultado el 6 de abril del 2022.
6. Mosquera, R. (2018). Máquinas de Soporte Vectorial, Clasificador Naïve Bayes y Algoritmos Genéticos para la Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos. Documento web. SciELO. Recuperado de: https://www.scielo.cl/scielo.php?pid=S071807642018000600153&script=sci_arttext. Consultado el 6 de abril del 2022.
7. Chaudhuri, K. (2022). Building Naive Bayes Classifier from Scratch to Perform Sentiment Analysis. Sitio web. Analytics Vidhya. Recuperado de: https://www.analyticsvidhya.com/blog/2022/03/building-naive-bayes-classifier-from-scratch-to-perform-sentiment-analysis/. Consultado el 6 de abril del 2022.
8. Orea, S; Salvador, A; García, M. (2005). Minería de datos: predicción de la deserción escolar mediante el algoritmo de árboles de decisión y el algoritmo de los k vecinos más cercanos. Documento web. Recuperado de: https://d1wqtxts1xzle7.cloudfront.net/34203825/e1-libre.pdf?1405405787=&response-content-disposition=inline%3B+filename%3DMineria_de_datos_prediccion_de_la_deserc.pdf&Expires=1680910229&Signature=PsosFJRB4CTsDnhOruF9jw95bA5xXLoBGYMRKmede~zeuBx66Sd8NMlvzh~HHOXUoICoEWXeauXyuxq0UKm9tTZL~qNXg~elHZ~nmmtaMX-uLDQvqEh3tIg~qlFLexfniADxgs7-aQb9a2Kpgfvj4ftWh2u7K~MYgjTOmEIqkY2PCJavhAk52w7Iu~VTbhAoKn3Q5nzdzJK6-6MW~2wsxhGBUkuTdhU0G4Q4wkjQCNxoCfBf5F8GhgueVEVRoj3RhonzQWUkr5bM9oOEqjBPZXAJyQgrJ8Y4oiEwt9txjSN7toeJqWX0ffXesTyb9qRgQ3rXDn1T-Ci-I3tYXKgd5w__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. Consultado el 7 de abril del 2022.

9. Barrientos, R.; Cruz, N.; Acosta, H. (2009). Árboles de decisión como herramienta en el diagnóstico médico. Documento web. Recuperado de: http://www.soporte.uv.mx/rm/num_anteriores/revmedica_vol9_num2/articulos/arboles.pdf. Consultado el 7 de abril del 2022.
10. Molero, G. (2022). Apuntes de clase de Minería de Datos. Facultad de Ingeniería. UNAM.