PROYECTO 1 – ETAPA 1

Apoyo a la detección de suicidios mediante aprendizaje automático con procesamiento del lenguaje natural

Grupo 7

Kevin Cohen Solano – 202011864

Juan Felipe Castro Vanegas – 201818130

Juan Carlos Marín Morales – 202013973

20 DE OCTUBRE DE 2022

UNIVERSIDAD DE LOS ANDES

Ingeniería de sistemas y computación

Tabla de contenido

Comprensión del negocio y enfoque analítico

Oportunidad/problema Negocio	La empresa busca apoyar la detección de casos de posible suicidio mediante la información histórica de miles de usuarios de la plataforma Reddit, los cuales hacen publicaciones de diferentes tipos. Esto lo hacen con el fin de poder ayudar a esas personas que se encuentran en una situación crítica antes de que sea tare, mediante un análisis automático que sea lo más rápido posible.
Enfoque analítico (Descripción del requerimiento desde el punto de vista de aprendizaje de máquina)	Lo que se busca es que mediante una base de datos que contiene las publicaciones de 195,700 usuarios y su destino final (suicidio o no) se construya un modelo de clasificación (mediante el procesamiento de lenguaje natural) que sea capaz de interpretar las publicaciones de futuros usuarios y determinar con cierto grado de precisión si esa persona tiene o no tendencias suicidas.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Sabemos que la empresa que se beneficia de este modelo no lo hace con ánimo de lucro, su objetivo parece acercarse más a tratar de reducir las tasas de suicidio en una región (por los datos parece ser estados unidos). Es entonces que los verdaderamente beneficiados serían aquellos que puedan ser ayudados de manera oportuna después de haber sido detectados con tendencias suicidas por el modelo.
Técnicas y algoritmos para utilizar	Aprendizaje supervisado para una tarea de clasificación Algoritmos: Naibe Bayes, SVM, Regresión Logística

Entendimiento y preparación de los datos

Tenemos una base de datios que contiene

Dentro de los datos podemos ver que son totalmente completos, debido a que no contiene valores nulos, sin embargo, los textos deben ser procesados pues tienen elementos que deben ser tratados, ya que por ejemplo, al tratarse de textos sacados de foros de internet, cosas como los emojis deben ser omitidos, es por esto que para la preparación de los datos se realiza el siguiente procedimiento:

- **Transformar la columna 'class' a una variable numérica de 1s y 0s**, esto con tal de que los algoritmos propuestos sean capaces de asignar una clase en sus test.
- **Pasar todos los textos a minúsculas**, pues facilita la creación de vocabulario al no tener elementos diferentes que realmente sean los mismos.
- Eliminar los números de los textos, para evitar que sucedan errores en la generación del vocabulario.
- **Tokenizar los textos**: esta operación se encarga de generar un vocabulario de las diferentes palabras que se encuentran en los datos. Esta operación consta de asignar a cada palabra un token y separar cada texto en estos últimos en forma de una lista, teniendo en cuenta una expresión regular que busca eliminar aquellos elementos ajenos al lenguaje inglés (tales como

emojis, caracteres no ASCII, etc). Después de esto se le aplica a cada token una función lemmatizer, la cual se encarga de unificar las diferentes conjugaciones y formas de las palabras en una sola para facilitar el procesamiento posterior de los algoritmos. Después de esto, las listas de listas se vuelven a unificar en un texto ya estando limpias.

- **Se eliminan los artículos, las conjunciones y las preposiciones**, pues estas no aportan nada a los algoritmos. Algunas palabras se eliminan de manera manual, mientras que la mayoría de las StopWords se eliminan con ayuda del módulo que contiene la información de las mismas

Modelado y Evaluación

Primer algoritmo - Regresión Logística

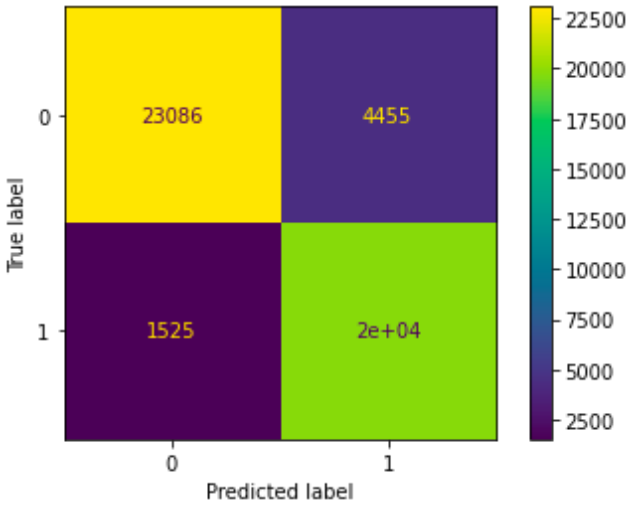
Hecho por: Juan Carlos Marín

Para este algoritmo, se hace uso de CountVectorizer, lo cual propone una matriz binaria, con la aparición de cada token en cada texto, y a partir de esto ejecutamos el algoritmo de regresión logística para la clasificación de los textos. Para esto se hizo uso de otro tokenizer llamado TweetTokenizer, el cual se usa para este tipo de textos (Ya que son posts de reddit).

A partir del uso de este modelo se obtuvieron las siguientes métricas sobre la separación del conjunto de test:

- Accuracy: 87.7%
- Precision 81.68 %
- Recall: 92.86%
- F1 Score: 86.91 %

Y, por último, se tiene la siguiente matriz de confusión:



Esto nos quiere decir que es un buen modelo, para clasificar un texto como un posible y desafortunado suicidio.

Preámbulo

Realizamos la partición de los datos de entrenamiento y prueba, con una proporción de 0.3.

Para los siguientes algoritmos utilizamos el vectorizado de Tfidf, el cual nos permite convertir las palabras a vectores de números, los cuales son interpretables por los modelos a continuación. Estos vectores forman un vocabulario que reemplaza el lenguaje natural por números en las variables X tanto de entrenamiento como de prueba.

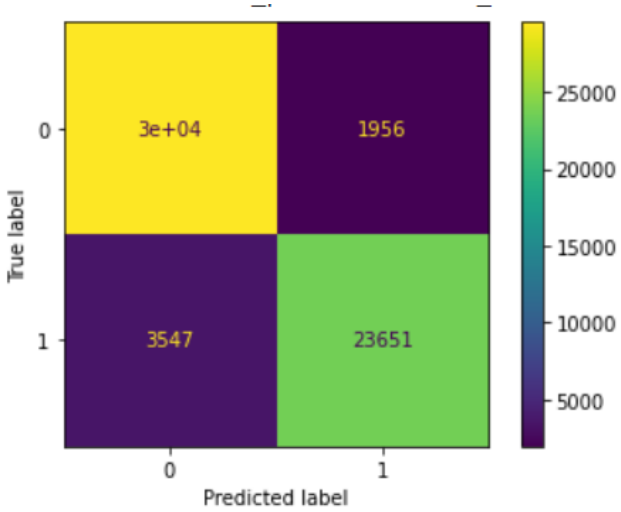
Segundo algoritmo – Naive Bayes

Hecho por: Juan Felipe Castro Vanegas

Este algoritmo consiste en la aplicación del teorema de Bayes y probabilidad condicional, según las diferentes características (palabras vectorizadas) y su influencia en la decisión de si son de una clase u otra. Para este caso se aplica primero un entrenamiento del modelo con la variable Y definida anteriormente (vectorizada) para luego hacer las pruebas definitivas, obteniendo así las métricas del modelo. Estos resultados se pueden ver a continuación:

- Accuracy Score: 90.62
- Precision Score: 92.36
- Recall Score: 86.95
- f1 Score: 89.57

Esta es la matriz de confusión:



Con estas métricas, podemos ver que el modelo fue bastante acertado a la hora de predecir cuando una persona tiene tendencias suicidas o no, pues tiene errores cercanos al 10% o inferiores.

Tercer Algoritmo – SVM (Support Vector Machines)

Hecho por: Kevin Cohen Solano

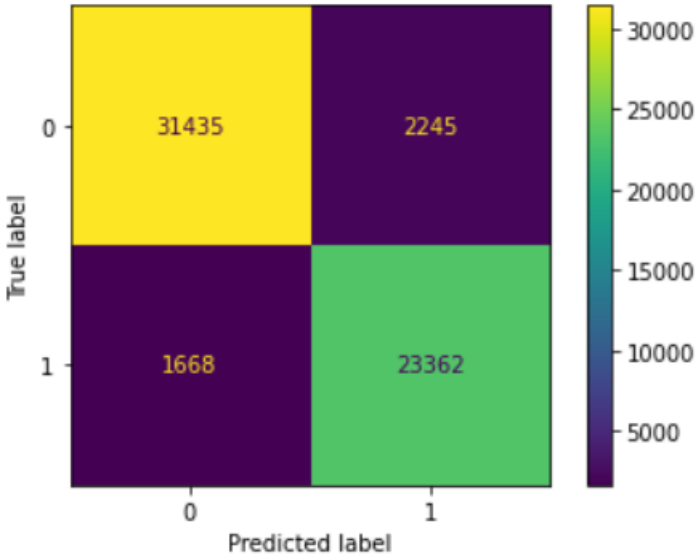
El algoritmo SVM funciona mediante la correlación de los diferentes datos en un espacio de grandes dimensiones, de forma que los puntos de datos se pueden

categorizar, incluso si los datos no se pueden separar linealmente de otro modo. Se detecta un separador entre las categorías y los datos se transforman de forma que el separador se puede extraer como un hiperplano. Tras ello, las características de los nuevos datos se pueden utilizar para predecir el grupo al que pertenece el nuevo registro (IBM, 2021).

Para este algoritmo igualmente entrenamos un modelo con los datos previamente separados y vectorizados, para después aplicar el modelo a los datos de prueba, obteniendo las siguientes métricas:

- Accuracy Score: 93.33
- Precision Score: 91.23
- Recall Score: 93.33
- f1 Score: 92.27

Esta es la matriz de confusión:



Podemos ver que este modelo tiene unas métricas excelentes a la hora de determinar si una persona tiene tendencias suicidas o no según sus publicaciones pues tiene errores menores al 8% y una precisión del 93.33%.

Resultados

Después del proceso realizado con los distintos algoritmos y metodologías, llegamos a la conclusión de que el mejor para cumplir con los objetivos del negocio es el modelo de SVM con un F1 Score de 92.27 %, el cual es el más alto entre todos los modelos probados. Esto es bastante beneficioso, dado que nos permite predecir con bastante certeza posibles casos de suicidio a partir de publicaciones en la plataforma de reddit, lo cual es el objetivo del proyecto. Esto puede ser muy útil para organizaciones como instituciones de salud mental o de ayuda a jóvenes.

Una buena estrategia que podrían implementar las organizaciones, incluso el mismo reddit, seria montar un sistema de alertas cuando se hagan publicaciones en reddit, para estar atentos e incluso llegar a prevenir estos desafortunados eventos.

Trabajo en Equipo

Para el trabajo en equipo se asignaron los siguientes roles:

- **Líder de proyecto:** Kevin Cohen Solano
- **Líder de negocio:** Kevin Cohen Solano
- **Líder de datos:** Juan Felipe Castro Vanegas
- **Líder de analítica:** Juan Carlos Marín Morales

El proyecto fue realizado en un periodo intensivo de aproximadamente 18 horas, donde tuvimos ciertos desafíos asociados a la preparación de los datos sobre todo en la limpieza de los mismos, esto pues no teníamos tantos conocimientos en cuanto a la tokenización y lematización, al igual que había elementos de los textos que no aportaban mucho y tampoco eran reconocidos como stopwords. Tal vez hubo problemas en referencia al entendimiento de los algoritmos, pues al ser todos nuevos para nosotros, tuvimos que investigar como funcionaban para poder tener una buena referencia en nuestras justificaciones. Estos problemas pudieron ser solucionados con trabajo en equipo y buenas habilidades de investigación grupales.

A continuación, las reuniones que realizamos:

Fecha	Razón	Asistentes
Octubre 3	Establecer roles, trabajo futuro y objetivos del proyecto	Kevin, Carlos, Felipe
Octubre 12	Retroalimentación grupal, resolución de dudas y correcciones generales	Kevin, Carlos, Felipe
Octubre 18	Consolidación final, detalles finales	Kevin, Carlos, Felipe

En una escala de 100 puntos proponemos la siguiente repartición:

- Juan Felipe Castro: 33.33
- Kevin Cohen: 33.33
- Juan Carlos Marín: 33.33

Referencias

Bedi, G. (2020, July 13). *A guide to Text Classification(NLP) using SVM and Naive Bayes with Python*. Medium. Retrieved October 20, 2022, from https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34

Cardellino, F. (2021, April 28). *Cómo funcionan los clasificadores Naive Bayes: con ejemplos de código de Python*. freeCodeCamp.org. Retrieved October 20, 2022, from https://www.freecodecamp.org/espanol/news/como-funcionan-los-clasificadores-naive-bayes-con-ejemplos-de-codigo-de-python/

Funcionamiento de SVM. (n.d.). Retrieved October 20, 2022, from https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-how-svm-works

Web-APIs-and-Predicting-Subreddit/Reddit.ipynb at main · tw1270/Web-APIs-and-Predicting-Subreddit. (n.d.). GitHub. Retrieved October 20, 2022, from https://github.com/tw1270/Web-APIs-and-Predicting-Subreddit/blob/main/Reddit.ipynb