Universidad de Los Andes

PROYECTO 1

Etapa 1

Inteligencia de Negocios

María del Pilar Villamil Giraldo

2022

INTEGRANTES

Andres Felipe Arias Russi - 201914996

Diego Granada Martínez - 201922383

Verónica Escobar Aristizábal - 201922107

Proyecto 1 – Etapa 1

Contenido

Introducción

Se decidió trabajar bajo el contexto de salud mental, y se desarrollara el proyecto alrededor del apoyo a la detección de intentos de suicidio a partir de información de Reddit a nivel de comunidades que sufren de depresión o han intentado suicidarse. El proyecto se divide en dos etapas. La primera etapa está relacionada a la construcción de modelos de datos analíticos, y el rol principal del equipo es el de científicos de datos. La segunda etapa está relacionada a el despliegue de estos modelos analíticos y el desarrollo de una aplicación, aquí el rol principal es el de un ingeniero de datos.

Contexto del Caso

El set de datos usado proviene originalmente de una colección de posts realizados a las comunidades de r/SuicideWatch y r/Depression. El autor del dataset menciona que los datos fueron recolectados por medio de PutshiftAPI. Los datos de SuicideWatch toma un recuento de todos los posts hechos a esa comunidad desde diciembre de 2008 a enero de 2021, mientras que los datos de r/Depression fueron recolectados desde las fechas de enero de 2009 a enero de 2021. Por lo tanto, ambos data sets recolectan aproximadamente 12 años de posts en total de cada una de las comunidades. Aunque este es el set de datos original para la investigación el set de datos base que se va a usar en el proyecto es el proporcionado por el equipo de Inteligencia de Negocios en un ZIP que contiene dos archivos uno el cual contiene un set de datos sin clasificar si son suicidal o non-suicidal mientras que otro ya tiene las clasificaciones hechas. Este set de datos contiene una mezcla de posts que pueden ser clasificados como *suicidal* y otros de temáticas no relacionadas y que pueden ser clasificados como *non-suicidal.*

Etapa 1. Analítica de Datos

Comprensión del negocio y enfoque analítico

El objetivo de este proyecto es desarrollar una herramienta para así dar apoyo en la detección de intentos de suicidio a partir de fuentes de texto. Por lo tanto, es sumamente importante para el proyecto desarrollar un modelo que sea capaz de hacer estas predicciones de una forma fiable y precisa. Adicionalmente, para poder hacer accesible las predicciones del modelo esto se requiere desarrollar una herramienta que sea capaz de replicar el proceso de preparación de datos de tal forma que se pueda exponer el modelo a través de una API para su uso más ágil.

El éxito del modelo y el proyecto en general se va a precisión de sus predicciones y su margen de error. En el contexto de suicidio una tasa alta o media de falsos negativos no es aceptable, puesto como es un tema de salud mental delicado que de no ser detectado con rapidez y precisión puede tener consecuencias graves. Adicionalmente, una tasa alta de falsos positivos no es aceptable, pero probablemente se le puede dar un mayor margen de error que a los falsos negativos. Por lo tanto, las métricas que más nos importarán en este caso serán primero el recall y segundo el F1-score. Recall no dice que tan bueno es un modelo en correctamente predecir todas las observaciones positivas de un modelo, por lo tanto, se ajusta más a la sensibilidad del contexto del caso. El F1-score es la media armónica de la precisión y el recall, lo cual nos permite aun tener en mente el balance entre la precisión y el recall que aún tiene importancia puesto si se debiesen optar por obtener un modelo altamente preciso en sus predicciones para asegurar su confiabilidad por parte del usuario.

Para poder lograr el objetivo del proyecto se va a tener que crear un modelo que haga uso de técnicas de analítica de datos aplicadas a textos (NLP) para poder determinar si un texto se puede clasificar como *suicidal* o *non-suicidal* de acuerdo con su contenido.

Dado esto se puede decir que el tipo de tarea es claramente de clasificación binaria, donde un texto dado se tiene que clasificar como *suicidal* o *non-suicidal*. Por lo tanto, todos los algoritmos empleados tienen que estar enfocados en determinar si un texto se puede clasificar como suicida o no. Esto tiene sentido, puesto el requerimiento del negocio es en producir una herramienta que ayude a determinar si un individuo tiene tendencias suicidas para así proveerle la ayuda y apoyo que necesita y evitar un posible intento lo más pronto posible.

Oportunidad/problema Negocio	Apoyo a la detección de intentos de suicidio
Enfoque analítico (Descripción del requerimiento desde el punto de vista de aprendizaje de máquina)	Clasificación Binaria
Organización y rol dentro de ella que se beneficia con la oportunidad definida	Se va a clasificar si un texto de un individuo dado se puede categorizar como suicida o no suicida.
Técnicas y algoritmos a utilizar	Support Vector Machines, Regresión logística, Naïve Bayes

Entendimiento de datos

El entendimiento de datos lo hicimos principalmente antes de hacer la preparación y limpieza de datos. El propósito del hacer el procesamiento en esta etapa es no solo entender la estructura y contenido de los datos, sino que también comenzar a reconocer problemas de calidad para después tener una mejor idea que pasos se tenían que tomar en la etapa de procesamiento. primero entender los datos y sus características, para así familiarizarnos más con el problema y que tipo de características pueden ser relevantes a la hora de hacer los modelos.

Primero se decidió graficar la distribución de frecuencia de palabras más comunes en los datos, excluyendo stop words. Así se puede tener una mejor idea de que palabras resaltaban y si tenían algún significado en el contexto de suicidio o qué tipo de palabras comunes.

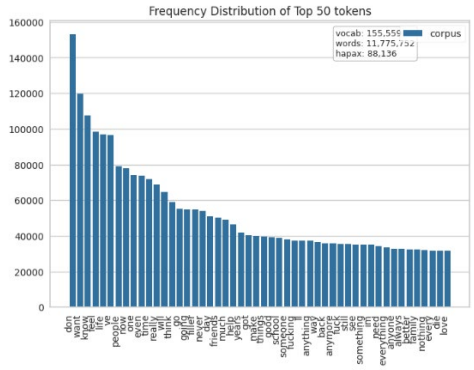


Ilustración 1. Distribución de Frecuencia de Non-stop-words

Segundo se decidió graficar los *stop words* más frecuentes en el set de datos, ya que se sabe que a la hora de procesar este tipo de palabras tienen que ser removidas porque no añaden valor adicional a el modelo para entender el texto.

Ilustración 2. Frecuencia de Stop Words

Tercero se hizo un Word cloud de todas las palabras en general, este deja ver de forma más clara el tipo de palabras que están siendo usadas en general. El Word cloud hace más evidentes palabras con connotaciones negativas que tienen una alta presencia en los textos, algunas de las palabras que resaltan en el contexto de la salud mental siendo del tipo *kill, pain, alone, hurt,* etc. Es con esta nube que también se resaltó la importancia de remover los sufijos (extraer la raíz) y los tiempos de una palabra (pasado, presente, futuro).

Ilustración 3. Word Cloud

La frecuencia de palabras individuales no dijo mucho sobre el contexto de los datos, por lo tanto, saber que frases eran comunes se volvió importante para saber qué tipo de frases eran más frecuentes en discusiones de suicidio y salud mental. Adicionalmente, por si solas es difícil determinar si alguna palabra cobra un sentido positivo o negativo por lo tanto palabras que son usadas juntas frecuentemente pueden dar más contexto. Por lo tanto, se recurrió a usar un n-grama el cual encuentra que oraciones de n palabras son las más frecuentes en el texto. Se probo con n de valor

2,3,4,5 y 6, sin duda n = 3 y n = 4 son los que dieron más información sobre las frases más comunes y las connotaciones negativas que tienen en el contexto del suicidio. Adicionalmente, con n más elevados se pudo comenzar a notar que había una alta frecuencia de palabras repetidas que se deben tratar cuando se procesen los datos puestos no dan valor agregado a los modelos.

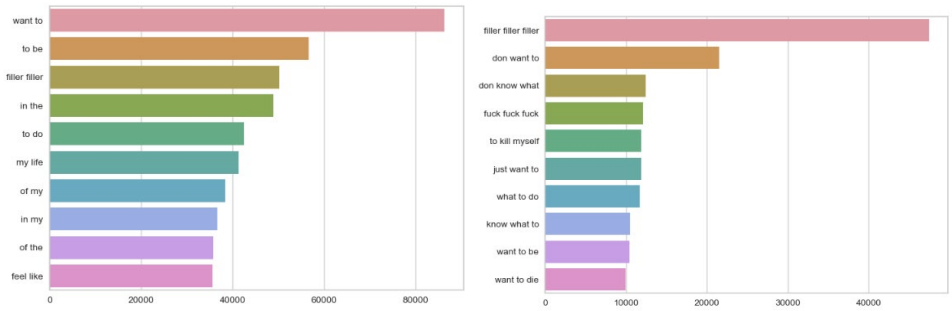


Ilustración 4. Top N-grama, 2 palabras y 3 palabras

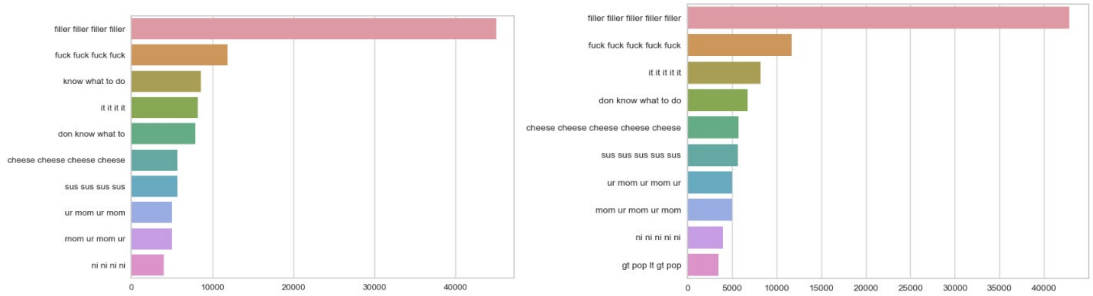


Ilustración 5. Top N-grama, 4 palabras y 5 palabras

En conclusión, el entendimiento de los datos ayudo a entender más el tipo de información que se está tratando y resaltar algunas palabras clave y oraciones de alta importancia en el contexto de discusiones de salud mental. Adicionalmente, se pudieron observar muchos aspectos que se deben tener en cuenta a la hora de procesar los datos tal como las características de la gramática en inglés (tiempos, contracciones, puntuación), *stop-words* y conectores que no son relevantes, lingo del internet (abreviaciones y palabras cuya definición oficial no es la misma que su definición en discusiones en el internet).

Preparación de datos

Para la preparación primero se realizó ajustes y limpieza inicial para así remover elementos innecesarios, arreglar problemas de calidad y preparar un poco los datos para su procesamiento. Adicionalmente, dentro de los mismos modelos se realizó procesamiento adicional de los datos cuando eran consumidos para el modelo para ajustarlos de acuerdo con el modelo.

Para la primera etapa de limpieza de datos se arreglaron problemas con la codificación de los datos dados en el Excel, se expandieron contracciones, se pasaron todos los datos a minúsculas, se removieron los *new lines*, los *stop-words*, se aplicó *stemming*, se eliminaron los valores numéricos, re redujeron secuencias de palabras repetidas a una sola y se quitó toda puntuación. Adicionalmente, se verifico que no existiesen valores duplicados, nulos, inconsistentes o inválidos dentro de los datos. Para preparar los datos para ser consumidos por los modelos se realizaron ajustes a los datos tal como hacer el *encoding* de etiquetas de clase, se tokenizaron los datos y se creó una nueva columna para guardar los datos tokenizados.

Basado en la implementación inicial de la limpieza de datos se creo una única función procesar_texto que generalizaba todo el procesamiento para que así pudiese ser llamada por cualquiera de los modelos antes de procesar un dato. Esto se hizo principalmente por que facilitara pasar los modelos a la implementación en la aplicación posteriormente, ya que la limpieza y preparación de los datos se hace a través de una sola función simple en lugar de fragmentado que es como se hizo originalmente en la preparación de datos.

Modelado y evaluación

Se realizaron tres modelos, cada uno usando un algoritmo diferente, con tal de descubrir cual produciría las mejores predicciones, es este caso, el de F1-Score más alto. Los tres algoritmos utilizaron como representación de las palabras el modelo TF-IDF de trigramas, pues permite que las palabras adquieran un determinado peso y relevancia que ayudará a que los modelos puedan entender mejor el contexto de las palabras y, por ende, una mejor predicción. Específicamente, se escogieron como máximo 10000 palabras y trigramas puesto que se obtenía un rendimiento más alto y es recomendable usar bigramas o trigramas.

1. Supervised Vector Machines (SVM)

Autor: Verónica Escobar Aristizábal

Support Vector Machine (SVM) es un algoritmo de machine learning supervisado que se puede usar tanto para tareas de clasificación como de regresión. En este algoritmo se traza un punto en un espacio N-dimensional, con N siendo el número de *features* que se tienen. Después se procede a identificar los hiperplanos que diferencian las diferentes clases de datos de la mejor manera. En un problema de clasificación binaria, como es el nuestro, solo se necesita encontrar un hiperplano que sea capaz de diferenciar ambas clases, *suicidal* o *non-suicidal,* lo mejor posible. Como se pueden encontrar varios hiperplanos que diferencien las clases el mejor se considera el que maximice la distancia (margen) entre el dato más cercano de cada clase. Se escogió el modelo debido a que SVM tiene una buena reputación en el análisis de texto natural, por lo cual era un buen punto de inicio.

La forma en la que se adapta un texto para ser procesado por *SVM* es que se trata al texto como una bolsa de palabras, para cada palabra que aparezca en la bolsa se tendrá un *feature*. Así que se transforma el texto en un vector numérico donde cada campo representa una palabra y el numero representa su frecuencia en relación con el resto de las palabras del texto. Por lo tanto, cada texto en el set de datos se representa como un vector de miles de dimensiones cada una representando la frecuencia de una palabra en el texto. Lo único que resta aquí es encontrar que hiperplano permite particionar este espacio N-dimensional para diferenciar las clases.

Para procesar los datos para el modelo se creó una función auxiliar que ayudaba a facilitar la limpieza y preparación de datos para el modelo, esta función es llamada dentro de una función usada para predecir la clasificación de un dato usando un modelo. Una vez se tenía el dato limpio se aplicaba *Term Frequency Inverse Document Frequency* (TFIDF) lo cual califica con más valor los términos que aparecen frecuentemente, pero en una cantidad de documentos más pequeños, al mismo tiempo convierte a el texto en un vector. Una vez hecho esto se llamaba a LSVC para crear el modelo y después se aplica una función para ayudar a seleccionar las mejores *features* con mayor peso para resaltarlas en los datos a procesar. La razón por la que se usó TFIDF fue explicada al inicio de la sección; no obstante, cabe resaltar que, para este modelo, se utilizó un método SelectFromModel *()* que permitió reducir las entradas de entrenamiento extrayendo las variables más relevantes y con más peso. Ahora, el modelo se optimizó con base a lo anterior y, además, se probaron distintos modelos de manera manual (debido a deficiencia de recursos), y se utilizó un parámetro de regularización de 1000, norma L1 y máximo 1000 iteraciones.

La matriz de confusión resultado del modelo muestra que el modelo es capaz de acertar la mayoría de las veces en sus predicciones sobre si un texto es *suicidal* o *non-suicidal*. Los falsos negativos y falsos positivos son bajos en comparación al set de datos y se tienen una mayor proporción de falsos negativos (3:1).

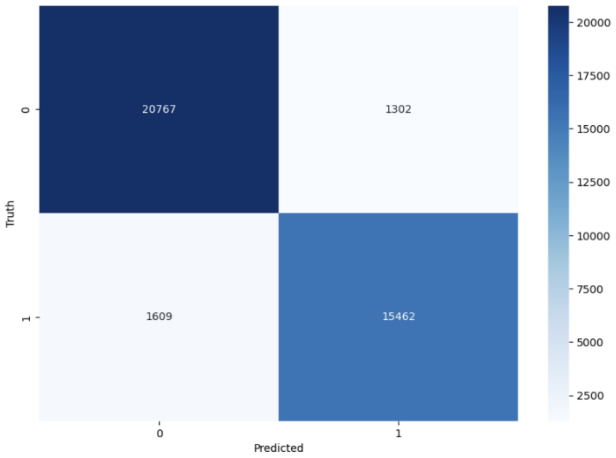


Ilustración 6. Matriz de Confusión SVM

En este caso todas las métricas presentan valores superiores a 0.9. Donde el F-1 score y la exactitud están entre los más altos.

```
Exactitud: 0.93
Recall: 0.9057465877804464
Precisión: 0.9223335719398712
Puntuación F1: 0.9139648293187528
```

	precision	recall	f1-score	support
0	0.93	0.94	0.93	22069
1	0.92	0.91	0.91	17071
accuracy			0.93	39140
macro avg	0.93	0.92	0.92	39140
weighted avg	0.93	0.93	0.93	39140

```
Accuracy:  0.9256259580991313
F1 score:  [0.93450332 0.91396483]
```

Ilustración 7. Métricas SVM

Dado las métricas de mayor importancia para modelos con el objetivo de negocio dado son el F-1 y el recall se puede considerar que este es en generar un buen modelo, a pesar de que entre todas las métricas el recall sea el más bajo pero su valor en sí es relativamente alto y cercano al del resto de las métricas.

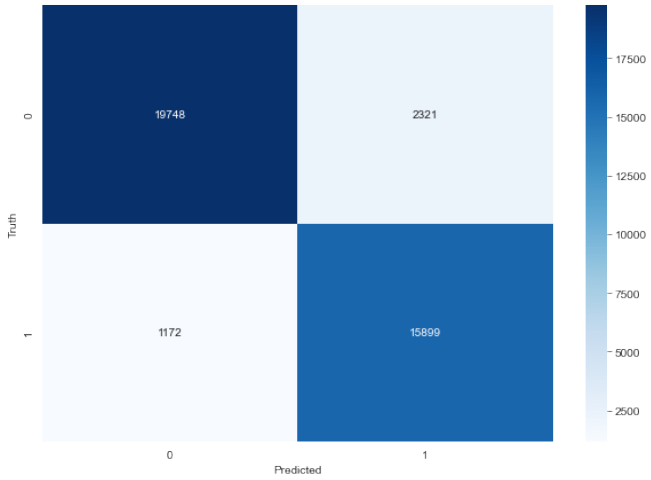
2. Naïve Bayes

Autor: Diego Granada Martínez

Naive Bayes es un algoritmo de machine learning supervisado que se usa frecuentemente para problemas de clasificación. Este algoritmo es un clasificador probabilístico que usa como fundamento de este algoritmo está en el teorema de Bayes para hacer sus predicciones. El teorema de bayes se puede encontrar cual es la probabilidad de un evento A ocurriendo dado que un evento B ocurrió. De esta forma se asume que los feature son independientes, la presencia de un set particular de feature no afecta el otro, por lo tanto, es llamado *naive*.

El procesamiento de los datos para el algoritmo fue hecho en una función de procesamiento, donde para cada ejemplo de texto pasado se aplican todos los pasos de procesamiento ya dichos (desde tokenizacion a remover puntuación) justo antes de alimentar el dato al modelo. Para el entrenamiento lo que se hizo es que se pasó todo el set de datos de entrenamiento por la función, creando un nuevo set de datos y con esto se entrenó el modelo, pero a la hora de consumir un dato nuevo con los datos de prueba es que se llama una función especial que primero llama la función de procesamiento y después llama la función de *fit*.

Es menester resaltar que el Naïve Bayes utilizado fue el multinomial y que es un constructor de frecuencias, es decir, en esencia no tiene parámetros de entrada. Aun así, se consiguieron las siguientes métricas:




```
[[19748  2321]
 [ 1172 15899]]
              precision    recall  f1-score   support

           0       0.94      0.89      0.92     22069
           1       0.87      0.93      0.90     17071

    accuracy                           0.91     39140
   macro avg       0.91      0.91      0.91     39140
weighted avg       0.91      0.91      0.91     39140
```

Además de lo anterior, algo destacable de Naïve Bayes es que es un algoritmo bastante rápido y que consigue resultados buenos.

3. Regresión Logística

Autor: Andres Felipe Arias Russi

Regresión logística es un algoritmo de machine learning supervisado que se usa para tanto problemas de clasificación como de regresión. Este algoritmo busca estimar la probabilidad de que un evento ocurra, en este caso que un dato dado pertenezca a una clase dependiendo de sus features. En problemas de clasificación binaria una probabilidad menor a .5 se predecirá como 0 mientras que una superior se predecirá como 1. Una regresión logística es una función, de una forma similar a la de una regresión lineal, por lo tanto, usa una combinación con pesos de los features dados y los pasa por una función sigmoide. Se escogió también debido al alto rendimiento del modelo entrenado de RL, y debido a que no es muy propenso al overfitting.

El proceso para adaptar un texto para ser procesado por Regresión Logística es similar al ya mencionado en ambos algoritmos. Con la única nota siendo que los pesos de la función regresión logísticas son calculados en base de las

features/palabras del set de datos dado. El algoritmo se escogió debido a que es un algoritmo simple y muy eficiente que nos permite evaluar rápidamente predicciones, algo que podía salir útil al desplegar la aplicación. Además, tolera muy bien las columnas irrelevantes, lo cual es muy útil en NLP debido a la cantidad de ngramas que no serán relevantes pero no se lograron eliminar en el preprocesamiento.

Para este algoritmo, se utilizó el optimizador L-BFGS y de clase multinomial. Después del entrenamiento, se consiguieron las siguientes métricas:

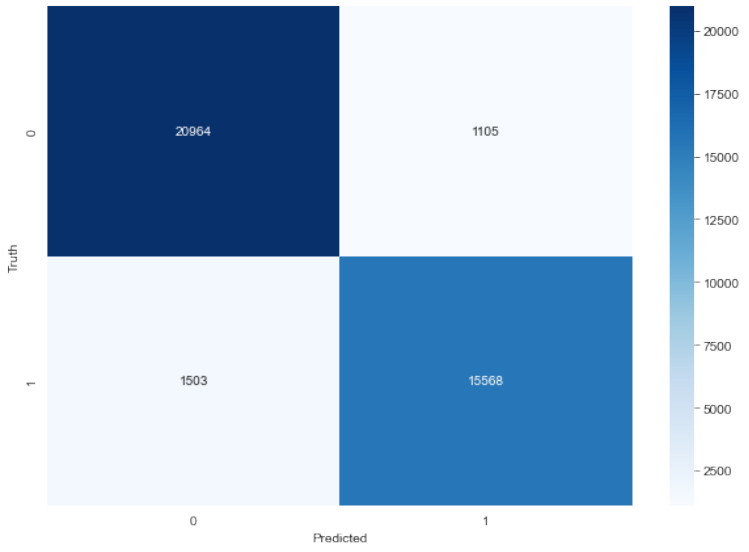
```
              precision    recall  f1-score   support

           0       0.93      0.95      0.94     22069
           1       0.93      0.91      0.92     17071

    accuracy                           0.93     39140
   macro avg       0.93      0.93      0.93     39140
weighted avg       0.93      0.93      0.93     39140
```

Por lo tanto, se concluye que Regresión Logística fue nuestro algoritmo con mejor puntaje F1, con aproximadamente 0.94.

Resultados



	SVM	Naive Bayes	Regresión logística
F1-Score	0.93	0.92	0.94
Recall	0.94	0.89	0.95
Exactitud	0.93	0.91	0.93
Precisión	0.93	0.94	0.93

Debido a las reglas del proyecto, la métrica más importante a considerar es la F1; sin embargo, como fue explicado, en la métrica de Recall recae una gran importancia en la detección de textos depresivos. Así las cosas, Regresión Logística termina siendo mejor; por lo tanto, es el que será utilizado. Unos valores de 0.94 y 0.93 en el F1 y la precisión respectivamente demuestran que el modelo logró acercarse al rendimiento máximo posible utilizando modelos simples de aprendizaje automático. Consideramos que los stakeholders deberían estar altamente satisfechos con el modelo de Regresión Logística que les vamos a entregar.

Trabajo en Equipo

Los roles fueron distribuidos de la siguiente manera. Verónica Escobar fue líder de proyecto y líder de datos, Diego Granada fue Líder de negocio y Andres Felipe Arias Russi fue líder de analítica. Veronica Escobar fue responsable del modelo que usa SVM, fue la encargada de hacer el entendimiento, procesamiento de datos y el video, además de gestionar la etapa 1 del proyecto. Diego Granada fue encargado con el modelo que usa Naive Bayes, aporto en partes del procesamiento de datos y en discusiones sobre el negocio relacionadas al proyecto. Andres Felipe Arias Russi fue quien implemento el algoritmo de regresión logística, fue el encargado de analítica, escogió los algoritmos a probar, reviso los resultados de estos y evaluó cual sería el mejor a usar para el proyecto.

Todos dedicaron alrededor de 6 horas de trabajo de trabajo individual para esta etapa del proyecto. El reto mayor que nos encontramos con esta etapa fue el de gestionar los algoritmos usados y como procesar los textos de tal forma que dieran la mayor información al algoritmo. Hicimos la planeación muy linear por lo tanto no se usó el tiempo tan efectivamente, a futuro sería bueno buscar una mejor forma para que todos pudiésemos trabajar en paralelo. Adicionalmente esto produjo un poco de discusión en términos de cómo se habían procesado los datos y llevo a tener que hacer ajustes para que fueran más digeribles por los modelos y en la implementación futura. Próximamente se pueden explorar masa medidas de optimización para algoritmos con los que no estamos tan familiares como SVM y regresión logística.

Video

Este es el link al video donde se presenta la etapa uno del proyecto

https://youtu.be/Ag80zc_j7YM

Datos, Resultados y Repositorio

En la wiki se encuentra el notebook donde esta el entendimiento y el procesamiento. Adicionalmente esta el notebook con todos los modelos.