Etapa 1 – Analítica de Textos

Inteligencia de Negocios – 2022_20 [Sección 1]

María Valentina García [201813544] y

Pedro Vallejo [201625720]

Tabla de contenido

Comprensión de negocio y enfoque analítico

Oportunidad / Problema del Negocio	Poder automatizar la detección de intentos de suicidio a partir de información recolectada de Reddit a nivel de comunidades que sufren de depresión o han intentado suicidarse. El criterio de éxito desde el punto de vista del negocio se basa en que la automatización pueda predecir en su mayoría (aprox. 60% de precisión) si un texto proviene de una persona en riesgo de suicidio.
Enfoque Analítico	Es notable que dentro de los datos recibidos hay una clara etiqueta la cual identifica ya sea si los datos son de suicidio o no. Por lo anterior la tarea de aprendizaje que se debe de usar es una de **aprendizaje supervisado,** utilizando **algoritmos de clasificación**.
Organización y rol dentro de ella que se beneficia con la oportunidad definida	La organización se verá beneficiada por la oportunidad definida ya que teniendo un sistema automatizado se podrán evaluar y concluir sobre cantidades masivas de datos de una manera rápida y eficiente. Dado esto proceso, se abre la posibilidad del negocio de responder de manera ágil sobre los casos de emergencia que se puedan presentar dentro de su plataforma.
Técnicas y algoritmos a utilizar	Se utilizarán algoritmos de aprendizaje supervisado, con objetivo de realizar una clasificación de textos en suicidio o no suicidio. Se seleccionaron 5 algoritmos para llevar esto a cabo: - Árboles de decisión - Bosques aleatorios - Bernoulli – Naive Bayes - Multinomial– Naive Bayes - Máquina de Soporte Vectorial con kernel lineal – LinearSVC El criterio de selección se dio respecto a dos factores importantes: desempeño en clasificación, según experiencia de laboratorios pasados y según la literatura, y tiempo de ejecución.

Entendimiento y preparación de los datos

Entendimiento

Para el entendimiento de los datos nos encontramos con un conjunto de datos el cual es conformado por: un índice de una publicación de Reddit, el texto de dicha publicación y una clasificación de si esa publicación viene de una persona con ideación suicida o no.

Preparación

Esta etapa se puede clasificar en limpieza, normalización y vectorización. Un paso extra se realizó para el algoritmo de Árboles de Decisión, que es el balanceo de clases.

- **Limpieza:**

En esta etapa se realizó to_lowercase (pasar todo el texto a minúscula), replace_numbers (convertir los números a palabras), remove_punctuation (quitar la puntuación), remove_non_ascii (eliminar los caracteres no ASCII), y remove_stopwords (eliminar las palabras “genéricas”, es decir, artículos, pronombres). Adicionalmente, se eliminaron los comentarios que no aportaban información y que eran solo números como, por ejemplo, el comentario 57369, el cual consistía en mencionar los dígitos de π.

- **Normalización:**
  En esta etapa se aplicó stem_words y lemmatize_verbs con los cuales se dejan los, adjetivos y verbos en su palabra raíz (sin prefijos, sufijos ni conjugaciones). Un ejemplo de stem es pasar de suicidal a suicide y un ejemplo de lemmatize es pasar de learning a learn.

- **Vectorización:**
  Se aplicó la vectorización siguiendo TfidfVectorizer. Tf-idf es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. El valor tf-idf aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son generalmente más comunes que otras. Siendo así, genera un vector (o lista) para cada comentario (o documento), en donde se tiene un número de Tf-idf para cada palabra. El resultado de esto es una matriz en donde las filas son los comentarios y las columnas las palabras de los comentarios.

- **Balanceo:**
  Un preprocesamiento adicional se realizó para el algoritmo de árboles de decisión y bosques aleatorios en el cual se realizó un balanceo de clases para mejorar los resultados de la clasificación. El dataset original contaba con 110141 datos de “non suicide” y 85527 datos de “suicide”. Se realizó un balanceo utilizando SMOTE dejando ambas clases con 110141 datos (0 corresponde a “non suicide” y 1 a “suicide”):

Modelado y evaluación

Se implementaron 5 algoritmos de clasificación:

- Árboles de decisión
- Bosques aleatorios
- Bernoulli – Naive Bayes
- Multinomial– Naive Bayes
- Máquina de Soporte Vectorial con kernel lineal – LinearSVC

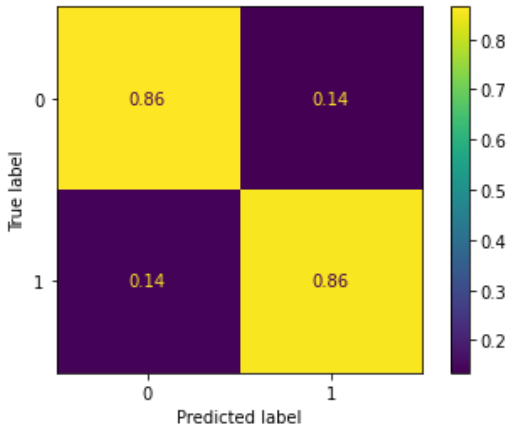
Los dos primeros algoritmos se escogieron basados en los buenos resultados obtenidos en el laboratorio de clasificación. Los tres últimos se escogieron en base a una revisión de los mejores algoritmos para clasificación de textos en la literatura.

Es importante resaltar que se intentó implementar otros algoritmos de clasificación como KNN y BaggingClassifier, pero estos tomaron un tiempo de ejecución demasiado grande en comparación a los 5 seleccionados, es por esto que se descartaron. Cada uno toma horas solamente en el entrenamiento.

Árboles de decisión

Desarrollado por: Pedro

El método de árboles de decisión es un método de aprendizaje supervisado principalmente utilizado para problemas de clasificación. Este es una estructura en forma de árbol donde cada nodo representa un atributo del dataset, las ramas representan las reglas de decisión y cada nodo hoja representa la respuesta al problema de clasificación. Para la escogencia de los mejores atributos para el nodo raíz y los primeros nodos en el árbol, este método usa métodos de selección de atributos coeficiente de Gini y la entropía.

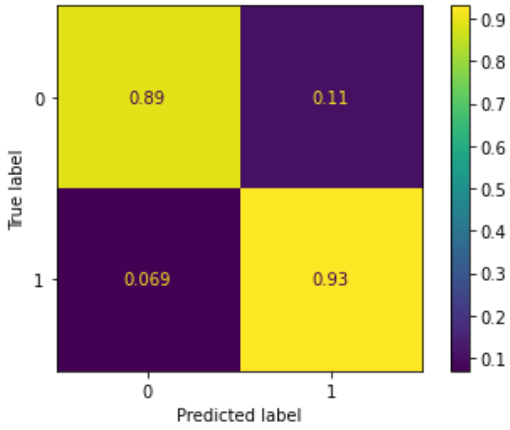


Exactitud sobre entrenamiento: 1.00
Exactitud sobre test: 0.86
Recall: 0.859002566923359
Precisión: 0.8637536873156342
Puntuación F1: 0.8613715756572898

Bosques aleatorios

Desarrollado por: Pedro

El método de bosques aleatorios el cual combina el resultado de múltiples arboles de decisión y resuelve el problema de overfitting que se presenta en los árboles de decisión. Los bosques de decisión tienen como principales hiperparámetros: el tamaño de los nodos, el número de árboles y el número de atributos.



Exactitud: 0.91
Recall: 0.9311972406281201
Precisión: 0.8960607913354878
Puntuación F1: 0.9132911955844387

Naive-Bayes

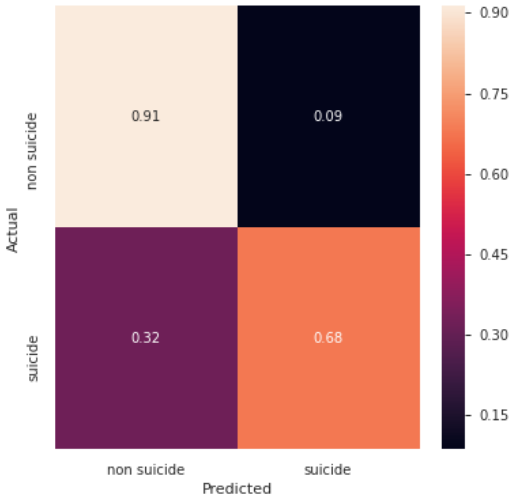
Desarrollado por: Valentina

El método Naive Bayes es un algoritmo de aprendizaje supervisado basado en la aplicación del teorema de Bayes con el supuesto “ingenuo” (naive) de independencia condicional entre cada par de features dado el valor de la variable de clase. Para cada comentario se determina la probabilidad de que sea de tipo “suicide” o “non suicide” dados los features de ese comentario son las palabras que contiene (o los puntajes dados por Tf-idf para ser más concretos). Para decidir a qué categoría pertenece, se comparan ambas probabilidades y se toma la categoría que tenga mayor probabilidad. El concepto de probabilidad condicional corresponde al teorema de Bayes y se supone que las variables (features) son independientes. La diferencia entre los distintos tipos de algoritmos de Naive Bayes radica en la distribución utilizada para calcular las probabilidades. En el caso de BernoulliNB, se utiliza la distribución de Bernoulli, mientras que en MultinomialNB se utiliza la distribución multinomial (que es una generalización de la distribución binomial). También existen otros paquetes de sklearn con otras distribuciones que no fueron utilizadas en este caso, que son CategoricalNB, ComplementNB y GaussianNB.

Los algoritmos de Naive Bayes reciben como parámetro un “alpha” que es un parámetro de suavizado, donde 0 implica sin suavizado. En nuestros casos, encontramos que los mejores valores son alpha=0.11112 para MultinomialNB y alpha=1e-05 para BernoulliNB.

Resultados:

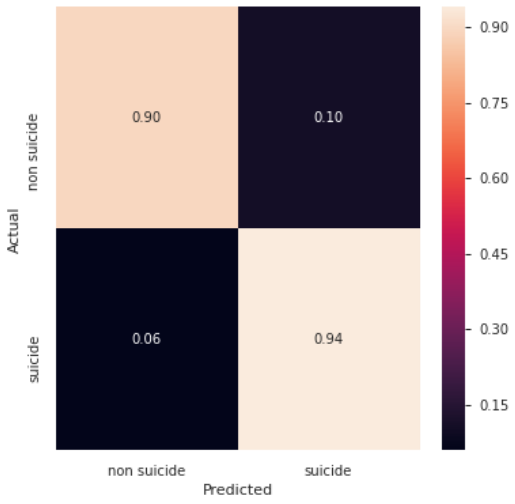
Bernoulli



Recall: 0.6779670872826655
Precisión: 0.8598082595870207
Puntuación F1: 0.7581363592027831

Multinomial

Desarrollado por: Valentina



Recall: 0.9405128801535152
Precisión: 0.875974870017331
Puntuación F1: 0.9070973893048427

Aunque ambos algoritmos siguen el mismo procedimiento, MultinomialNB tiene un mejor desempeño en todos los scores. Esto nos dice que los datos se describen mejor por una distribución multinomial que por una Bernoulli.

Máquina de Soporte Vectorial – LinearSVC

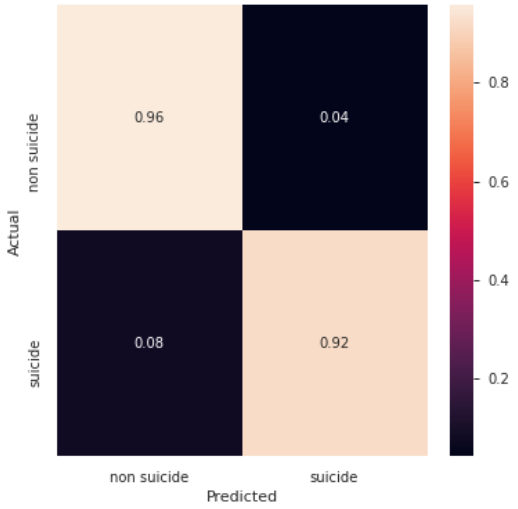
Desarrollado por: Valentina

La máquina de soporte vectorial es un algoritmo simple para tareas de clasificación y regresión. Puede proporcionar alta precisión con menos potencia de cálculo muy

rápido. El objetivo del algoritmo de la máquina de vectores de soporte es encontrar un hiperplano en un espacio N-dimensional (N: el número de características) que clasifique claramente los puntos de datos. Para separar las dos clases, se pueden elegir muchos hiperplanos posibles. Nuestro objetivo es encontrar un plano que tenga el margen máximo, es decir, la distancia máxima entre puntos de datos de ambas clases. Maximizar la distancia del margen proporciona cierta confianza para que los puntos de datos futuros se puedan clasificar con más seguridad. Los hiperplanos son límites de decisión que ayudan a clasificar los puntos de datos. Los puntos de datos que caen a cada lado del hiperplano se pueden atribuir a diferentes clases. Los vectores de soporte son puntos de datos que están más cerca del hiperplano e influyen en la posición y orientación del hiperplano. Usando estos vectores de soporte, maximizamos el margen del clasificador. Eliminar los vectores de soporte cambiará la posición del hiperplano. Estos son los puntos que nos ayudan a construir nuestra SVM.

El método Clasificador de vectores de soporte lineal (LinearSVC) aplica una función de kernel lineal para realizar la clasificación y funciona bien con una gran cantidad de muestras. El algoritmo recibe como parámetro “C” el cual es un parámetro de regularización. La fuerza de la regularización es inversamente proporcional a C. En nuestro caso, escogimos un óptimo de C=1.12.

Resultados:



Recall: 0.9184741524684538
Precisión: 0.9438866977411259
Puntuación F1: 0.9310070437063452

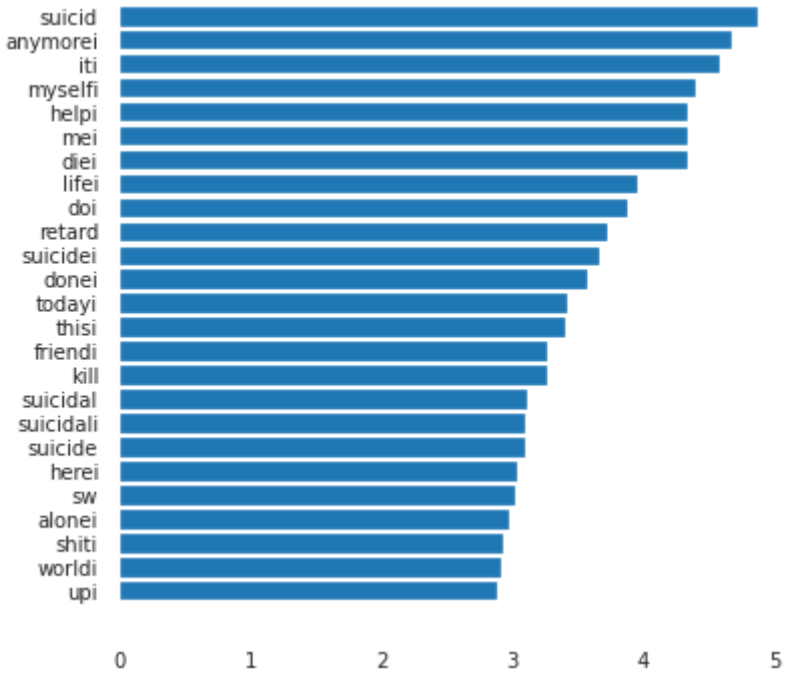
Resultados

A primera vista se puede notar como mayoría de los modelos utilizados cumplen con el criterio de éxito principal desde el punto de vista del negocio que era obtener un modelo con una precisión alta (precisión alta definida como una mayor al 60%). Incluyendo el modelo que obtuvo los peores resultados, BernoulliNB, se cumple este criterio de éxito establecido desde el punto de vista del negocio.

Sin embargo, el modelo que tuvo un mejor puntaje F1 score (que representa tanto el recall y la precisión), fue el de máquina de soporte vectorial LinearSVC, el cual fue de F1:0.93. Además, este modelo no tuvo overfitting pues el Recall también es alto, de maś del 90%. Una ventaja adicional y que es útil para el tipo de plataformas como Reddit, que manejan grandes cantidades de datos, es que el tiempo de ejecución de LinearSVC es mucho menor que el segundo algoritmo con mejores resultados, que es el de Bosques aletorios, teniendo tiempo de ejecución del orden de segundos y minutos respectivamente.

Sin embargo, si pensamos como objetivo el poder distinguir los comentarios con comportamiento suicida, deberíamos fijarnos en el algoritmo que tuvo un mayor éxito en la identificación de los comentarios clasificados como “suicide”. El mejor algoritmo para esto fue el de MutlinomialNB, clasificando erroneamente solo el 6% de los comentarios que debían ser de tipo “suicide”.

Aun considerando esto, decidimos escoger el que tiene mejor puntaje general de F1: LinearSVM. Además, encontramos que las palabras que tienen más importancia al momento de la clasificación son las relacionadas con suicidio (suicide), ayuda (help), morir (die), matar (kill), como se muestra en el siguiente gráfico:



La selección de estas palabras como las más importantes tiene sentido en el contexto de usuarios de comunidades que sufren de depresión o han intentado suicidarse, y con respecto a nuestro objetivo que es identificar potenciales intentos de suicidio.

Siendo así, sugerimos a la empresa utilizar el algoritmo de LinearSVC con un parámetro C=1.12, pues tiene muy buenos resultados al momento de clasificar comentarios entre “suicide” y “non suicide”, no hace overfitting y tiene un tiempo de

ejecución extremadamente corto. Teniendo en cuenta la gran cantidad de datos manejados, consideramos que esta es la mejor opción. Recomendamos tener especial cuidado con los comentarios que contienen las palabras “suicide”, “anymore”, “myself”, “help”, “die”, y las otras que se muestran en el gráfico de barras anterior. Sugerimos tener un identificador automático de estas palabras para hacerle seguimiento a estos usuarios desde una etapa temprana y tener así más chances de detectar los intentos de suicidio.

Trabajo en equipo

Roles

María Valentina García: Líder de datos y líder de analítica
Pedro Vallejo: Líder de proyecto y líder de negocio

Tareas realizadas

Tarea	Comprensión de negocio y enfoque analítico
Integrante	María Valentina García y Pedro Vallejo
Tiempo	45 minutos

Tarea	Entendimiento y preparación de los datos
Integrante	María Valentina García y Pedro Vallejo
Tiempo	435 minutos

Tarea	Árbol de decisión
Integrante	Pedro Vallejo
Tiempo	180 minutos

Tarea	Bosques aleatorios
Integrante	Pedro Vallejo
Tiempo	120 minutos

Tarea	Naive-Bayes
Integrante	Maria Valentina Garcia
Tiempo	60 minutos

Tarea	Multinomial
Integrante	Maria Valentina Garcia
Tiempo	20 minutos

Tarea	Máquina de Soporte Vectorial – LinearSVC
Integrante	Maria Valentina Garcia
Tiempo	60 minutos

Tarea	Resultado
Integrante	María Valentina García y Pedro Vallejo

Tiempo	60 minutos

Repartición de puntos

María Valentina García: 50 puntos

Pedro Vallejo: 50 puntos

Reflexión

Se tuvo un buen trabajo en equipo donde nos apoyamos mutuamente para el inicio del proyecto e intentamos sacar los mejores modelos después. El único punto para mejorar creemos que es comenzar con la preparación de datos mucho antes ya que esta compuso el paso que más nos costó y de haber empezado con más antelación, hubiéramos tenido más tiempo para entrenar otro tipo de algoritmos, que toman más tiempo de entrenamiento, pero que podrían resultar en mejores clasificaciones.