Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas
Maestría en Ciencia de Datos

Tarea 3

-ANÁLISIS DE CLASIFICACIÓN DE DATOS- REDDIT SUICIDIOS

Procesamiento y Clasificación de Datos
MET. Mayra Cristina Berrones Reyes

Alumna:
Abril Grisel Guevara Cedillo

Matrícula:
1419239

San Nicolás de los Garza a 2 de junio del 2022

0.1. Objetivo

Realizar un análisis de clasificación de datos comparando diferentes modelos explicando ventajas y desventajas de cada así como utilizar diferentes métricas para evaluar los resultados del modelo.

0.2. Introducción

Continuaré con la misma base de datos que utilicé para la tarea 1 y 2 la cual se encuentra en Kaggle es una colección de publicaciones de los subreddits "SuicideWatchz "Depresión"de la plataforma Reddit, las publicaciones se recopilan utilizando la API pushshift. La base originalmente tiene una columna la cual contiene la palabra Suicidio si el comentario tiene ideación suicida y No Suicidio para los que no, cuando realicé la tarea 1 y 2 ignore esta columna pero para el objetivo de esta tarea si es de utilidad para poder aplicar un modelo de aprendizaje supervisado. Corté el set a sólo 2000 comentarios.

0.3. Procesamiento de los datos

Para poder aplicar los modelos de clasificación primero definí la variable features como los comentarios y la variables labels como la clasificación de Suicidio o No Suicidio. Después apliqué varios pasos para obtener la variable features limpia:

- Quitar características especiales
- Sustituir espacios múltiples por un sólo espacio
- Convertir a minúsculas

  Después, usando la librería NLTK se quitaron los stopwords y se vectorizaron por medio de la función TfidfVectorizer y después vectorizer.fit transform

0.4. Análisis de Clasificación de Texto

El análisis que aplicaré será de clasificación binaria, los sistemas de aprendizaje de máquina aprenden por medio de ejemplos. En clasificación, para que el sistema aprenda, necesita saber al menos dos cosas de los ejemplos:

- Características: Las propiedades o atributos de los ejemplos. Es decir lo que distingue, diferencia o caracteriza a los ejemplos.
- Etiquetas: Identificador de cuál es la clase a la que pertenece el ejemplo.

El modelo de aprendizaje de máquina necesita varios ejemplos para aprender; a los ejemplos se les conoce también como instancias.

Para que el modelo aprenda se debe entrenar por lo que se divide en un set de entrenamiento y uno de prueba para evaluar la precisión del modelo. El set de entrenamiento es un un 80 % y el set de prueba es del 20 % restante.

Se aplicaron 5 modelos de clasificación para comparar los resultados:

- Clasificación de texto con Random Forest

  El clasificador Random Forest es un algoritmo de aprendizaje supervisado que se puede utilizar para problemas de regresión y clasificación. Es uno de los algoritmos de aprendizaje automático más populares debido a su alta flexibilidad y facilidad de implementación. Utiliza la aleatoriedad para mejorar su precisión y combatir el sobreajuste, lo que puede ser un gran problema para un algoritmo tan sofisticado. Estos algoritmos hacen árboles de decisión basados en una selección aleatoria de muestras de datos y obtienen predicciones de cada árbol. Después de eso, seleccionan la mejor solución viable a través de votos. Suponiendo que un conjunto de datos tenga características "m", el random forest elegirá aleatoriamente las características "k"donde k ¡ m. Ahora, el algoritmo calculará el nodo raíz entre las características k eligiendo un nodo que tenga la mayor ganancia de información. Después de eso, el algoritmo divide el nodo en nodos secundarios y repite este proceso "n"veces. Ahora se tiene un bosque con n árboles. Por último, se combinan los resultados de todos los árboles de decisión presentes en el bosque.

  Técnicamente, es un algoritmo de conjunto. El algoritmo genera los árboles de decisión individuales a través de una indicación de selección de atributos. Cada árbol se basa en una muestra aleatoria independiente. En un problema de clasificación, cada árbol vota y la clase más popular es el resultado final. Por otro lado, en un problema de regresión, calculará el promedio de todas las salidas del árbol y ese sería su resultado final.

  Ventajas:

  - El algoritmo de Random Forest es significativamente más preciso que la mayoría de los clasificadores no lineales.
  - Este algoritmo también es muy robusto porque utiliza múltiples árboles de decisión para llegar a su resultado.
  - El clasificador de Random Forest no se enfrenta al problema del sobreajuste porque toma el promedio de todas las predicciones, cancelando los sesgos y, por lo tanto, solucionando el problema del sobreajuste.
  - Puede utilizar este algoritmo tanto para problemas de regresión como de clasificación, lo que lo convierte en un algoritmo muy versátil.

  Desventajas

- Este algoritmo es sustancialmente más lento que otros algoritmos de clasificación porque utiliza múltiples árboles de decisión para hacer predicciones. Cuando un clasificador de random forest hace una predicción, cada árbol en el bosque tiene que hacer una predicción para la misma entrada y votar sobre la misma. Este proceso puede llevar mucho tiempo.
- Debido a su ritmo lento, los clasificadores de random forest pueden no ser adecuados para predicciones en tiempo real.
- El modelo puede ser bastante difícil de interpretar en comparación con un árbol de decisión, ya que puede hacer una selección siguiendo el camino del árbol. Sin embargo, eso no es posible en un random forest, ya que tiene múltiples árboles de decisión.

A continuación se muestran los resultados de este modelo

```
Matriz de Confusion RF
[[169  31]
 [ 21 179]]
Clasificación RF
              precision    recall  f1-score   support

 No Suicidio       0.89      0.84      0.87       200
    Suicidio       0.85      0.90      0.87       200

    accuracy                           0.87       400
   macro avg       0.87      0.87      0.87       400
weighted avg       0.87      0.87      0.87       400

Score RF
0.87
```

Tabla 1: Resultados Random Forest

Los resultados muestran un porcentaje alto de exactitud 87 %

- Clasificación de texto con KNeighbors

  Es un método que simplemente busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean. Es un algoritmo:

  - Supervisado: esto -brevemente- quiere decir que tenemos etiquetado nuestro conjunto de datos de entrenamiento, con la clase o resultado esperado dada “una fila” de datos.
  - Basado en Instancia: Esto quiere decir que nuestro algoritmo no aprende explícitamente un modelo (como por ejemplo en Regresión Logística o árboles de decisión). En cambio memoriza las instancias de entrenamiento que son usadas como “base de conocimiento” para la fase de predicción.

Este algoritmo funciona de la siguiente forma:

- Calcula la distancia entre el item a clasificar y el resto de items del dataset de entrenamiento.
- Selecciona los "k" elementos más cercanos (con menor distancia, según la función que se use)
- Realiza una "votación de mayoría" entre los k puntos: los de una clase/etiqueta que ¡¡dominen¿¿ decidirán su clasificación final.

Teniendo en cuenta el punto 3, para decidir la clase de un punto es muy importante el valor de k, pues este terminará casi por definir a qué grupo pertenecerán los puntos, sobre todo en las "fronteras" entre grupos. Para este análisis elegí una k de 10.

Ventajas:

- Es sencillo de aprender e implementar

Desventajas

- Utiliza todo el dataset para entrenar "cada punto" y por eso requiere de uso de mucha memoria y recursos de procesamiento (CPU). Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).

A continuación se muestran los resultados de este modelo

```
Matriz de Confusion KN
[[183  17]
 [ 59 141]]
Clasificación KN
              precision    recall  f1-score   support

 No Suicidio       0.76      0.92      0.83       200
    Suicidio       0.89      0.70      0.79       200

    accuracy                           0.81       400
   macro avg       0.82      0.81      0.81       400
weighted avg       0.82      0.81      0.81       400

Score KN
0.81
```

Tabla 2: Resultados K Neighbors

Los resultados muestran un porcentaje de exactitud 81 %

- Clasificación de texto con Naive Bayes

  Los clasificadores Naive Bayes (NBC por su siglas en inglés) son algoritmos de aprendizaje automático simples pero potentes. Se basan en la probabilidad condicional y el teorema de Bayes.

  Ventajas:

- Son algoritmos muy sencillos que nos permiten conseguir buenos resultados en algunos problemas de clasificación sin necesidad de muchos recursos. También escalan muy bien, lo que significa que podemos agregar muchas más funciones y el algoritmo seguirá siendo rápido y confiable.
- Incluso en el caso de que los NBC no fueran adecuados para el problema que estábamos tratando de resolver, podrían ser muy útiles como referencia.Este proceso puede ahorrarnos mucho tiempo y nos da una retroalimentación inmediata sobre si los algoritmos complejos realmente valen la pena para nuestra tarea.

A continuación se muestran los resultados de este modelo, utilicé un alfa de 0.01

```
Matriz de Confusion NB
[[145  55]
 [ 10 190]]
Clasificación NB
              precision    recall  f1-score   support

 No Suicidio       0.94      0.72      0.82       200
    Suicidio       0.78      0.95      0.85       200

    accuracy                           0.84       400
   macro avg       0.86      0.84      0.84       400
weighted avg       0.86      0.84      0.84       400

Score NB
0.81
```

Tabla 3: Resultados Naive Bayes

Los resultados muestran un porcentaje de exactitud 81 %

- Clasificación de texto con Support Vector Machine

  Support Vector Machine es un algoritmo de clasificación supervisado donde se traza una línea entre dos categorías diferentes para diferenciarlas. SVM también se conoce como la red de vectores de soporte. Hay muchos casos en los que la diferenciación no es tan simple como se muestra arriba. En ese caso, la dimensión del hiperplano debe cambiarse de 1 dimensión a la enésima dimensión. Esto se llama Kernel. Para ser más simple, es la relación funcional entre las dos observaciones. Agregará más dimensiones a los datos para que podamos diferenciar fácilmente entre ellos.

  Podemos tener tres tipos de núcleos.

  - Núcleos lineales úcleos polinómicos úcleo de la función de base radial

  Ventajas:

- Eficaz en espacios de grandes dimensiones.
- Todavía eficaz en casos donde el número de dimensiones es mayor que el número de muestras.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamada vectores de soporte), por lo que también es eficiente en memoria.

Desventajas

- Los SVMs no proporcionan directamente estimaciones de probabilidad, éstas se calculan utilizando una validación cruzada quíntuple.

A continuación se muestran los resultados de este modelo

```
Matriz de Confusion SVC
[[183  17]
 [ 30 170]]
Clasificación SVC
              precision    recall  f1-score   support

 No Suicidio       0.86      0.92      0.89       200
    Suicidio       0.91      0.85      0.88       200

    accuracy                           0.88       400
   macro avg       0.88      0.88      0.88       400
weighted avg       0.88      0.88      0.88       400

Score SVC
0.8825
```

Tabla 4: Resultados Support Vector Machine

Los resultados muestran un porcentaje alto de exactitud de 88.25 %

- Clasificación de texto con Stochastic Gradient Descent

  El Descenso de Gradientes Estocástico (SGD) es un enfoque simple pero muy eficiente para el aprendizaje discriminatorio de clasificadores lineales bajo funciones de pérdida convexa tales como Máquinas Vectoriales de Soporte (lineal) y Regresión Logística. A pesar de que SGD ha estado presente en la comunidad de aprendizaje automático durante mucho tiempo, ha recibido una considerable cantidad de atención recientemente en el contexto del aprendizaje a gran escala. SGD se ha aplicado con éxito a los problemas de aprendizaje a gran escala y escasa máquina de aprendizaje a menudo encontrados en la clasificación de textos y el procesamiento del lenguaje natural.

  Un gradiente es la pendiente de la función, el grado de cambio de un parámetro con la cantidad de cambio en otro parámetro.Matemáticamente, se puede describir como la derivada parcial de un conjunto de parámetros relativos a sus entradas. Cuanto mayor sea la pendiente, más pronunciada será la pendiente.El descenso de gradiente es una función convexa.

Gradient Descent se puede describir como un método iterativo que se utiliza para encontrar los valores de los parámetros de una función que minimiza la función de costo tanto como sea posible.Los parámetros se definen inicialmente con un valor particular y a partir de eso, gradiente descent se ejecuta de una manera alternativa para encontrar los valores óptimos de los parámetros, utilizando el cálculo, para encontrar el valor mínimo posible de la función de costo dada.

Ventajas:

- Para conjuntos de datos más grandes, puede converger más rápido, ya que provoca actualizaciones de los parámetros con mayor frecuencia.
- Debido a las actualizaciones frecuentes, los pasos dados hacia los mínimos de la función de pérdida tienen oscilaciones que pueden ayudar a salir de los mínimos locales de la función de pérdida (en caso de que la posición calculada resulte ser el mínimo local).
- Es computacionalmente rápido ya que solo se procesa una muestra a la vez.

Desventajas

- Debido a las frecuentes actualizaciones, los pasos dados hacia los mínimos son muy ruidosos. Esto a menudo puede inclinar el descenso del gradiente hacia otras direcciones.
- Además, debido a los pasos ruidosos, puede llevar más tiempo lograr la convergencia a los mínimos de la función de pérdida.
- Las actualizaciones frecuentes son computacionalmente costosas debido al uso de todos los recursos para procesar una muestra de entrenamiento a la vez.

A continuación se muestran los resultados de este modelo

```
Matriz de Confusion SGD
[[178  22]
 [ 31 169]]
Clasificación SGD
              precision    recall  f1-score   support

 No Suicidio       0.85      0.89      0.87       200
    Suicidio       0.88      0.84      0.86       200

    accuracy                           0.87       400
   macro avg       0.87      0.87      0.87       400
weighted avg       0.87      0.87      0.87       400

Score SGD
0.8675
```

Tabla 5: Resultados Stochastic Gradient Descent

Los resultados muestran un porcentaje alto de exactitud de 86.75 %

0.5. Resultados y Conclusiones

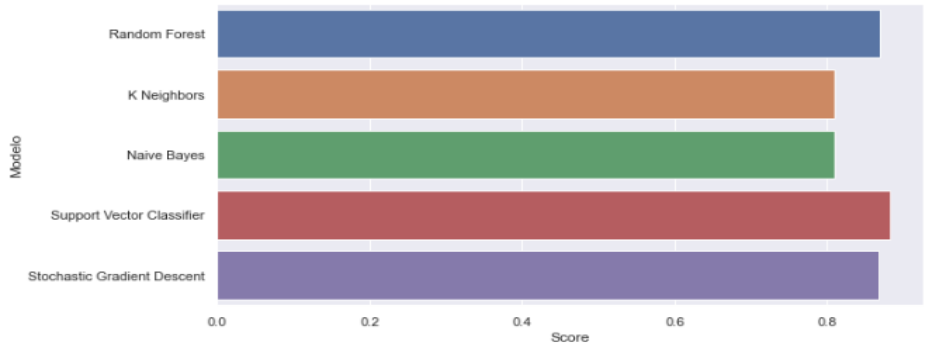
En la gráfica comparativa de la métrica de exactitud de clasificación en los resultados de las 5 modelos aplicados, puedo observar que los modelos con una exactitud más alta son el Support Vector Classifier y el Random Forest las cuales son mayores a 87 %. El hecho de que el Random Forest utiliza la aleatoriedad para mejorar su precisión y de esta manera combate el sobreajuste, puede ser una de las ventajas más grandes versus los otros modelos comparados. En el contexto de esta base en particular es importante calcular la exactitud del modelo en su totalidad debido a que si el modelo categoriza algún comentario como ideación suicida cuando en realidad no lo es (falsos positivos) puede causar algunos inconvenientes para las personas que utilicen esta herramienta, por ejemplo un psicólogo en una escuela que este interesado en saber si los estudiantes presentan este padecimiento psicológico pueden perder el tiempo enfocándose en estudiantes que en realidad no presentan este padecimiento psicológico en lugar de enfocarse en las estudiantes que realmente lo necesitan. Y en caso contrario, si el modelo categoriza algún comentario como normal cuando en realidad si tiene ideación suicida (falsos negativos) sería aún más grave debido a que no estaría dándole la atención al estudiante que lo necesita.

Debido a lo anterior otra métrica que podría ser utilizada para evaluar el modelo después de la exactitud, podría ser la de 1- sensibilidad, sensibilidad se describe como la proporción de verdaderos positivos, por lo que al restar 1 - menos la tasa de verdaderos positivos, obtengo la tasa de falsos negativos. En las tabla de resultados mostradas por cada modelo la sensibilidad la podemos ver en la métrica llamada recall en la categoría positiva. Por lo tanto el que tenga este valor más pequeño sería el mejor bajo este criterio de comparación.

A continuación se muestran el comparativo de este modelo en cuanto a la métrica de exactitud que es la más adecuada debido a que lo que se espera es que ninguno de los dos casos comentados en el párrafo anterior suceda.

Modelo	Score
Random Forest	0.8700
K Neighbors	0.8100
Naive Bayes	0.8100
Support Vector Classifier	0.8825
Stochastic Gradient Descent	0.8675

Tabla 6: Comparativo entre la exactitud de los modelos



Gráfica 1: Comparativo entre la exactitud de los modelos

0.6. Referencias

Github `https://github.com/AbrilGuevara/Procesamiento-de-Datos/tree/main/Tarea%203`

Base de datos `https://www.kaggle.com/datasets/nikhileswarkomati/suicide-watch?resource=download`