Práctica no. 3: K-Nearest Neighbors

Jair Antonio Bautista Loranca
a01365850@itesm.mx
Tecnológico de Monterrey
Monterrey, N.L., México

Maximiliano Zambada Camacho
a01570146@itesm.mx
Tecnológico de Monterrey
Monterrey, N.L., México

RESUMEN

Dentro de los diferentes tipos de modelos que existen para Machine Learning, éstos se pueden categorizar entre modelos que requieren entrenamiento y modelos que no lo requieren. La diferencia radica en que los modelos que no requieren entrenamiento realizan el cálculo en el momento de realizar una petición. Esto implica que aunque se ahorra el tiempo de entrenamiento habrá un tiempo de espera por cada petición.

ACM Reference Format:
Jair Antonio Bautista Loranca and Maximiliano Zambada Camacho. 2021. Práctica no. 3: K-Nearest Neighbors. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1. INTRODUCCIÓN

Continuando con el trabajo que se ha realizado en prácticas anteriores para profundizar en las diferentes técnicas de aprendizaje automático. Para esta práctica se trabajo con el modelo de K-Nearest Neighbors, el cual forma parte de un tipo de modelos que no requieren entrenamiento.

. Cabe mencionar que además de implementar el modelo de K-Nearest Neighbors, también se realizará una comparación contra el modelo de regresión logística. Esto con la intensión de medir y comparar el perfomance que cada modelo tiene con el mismo conjunto de datos.

2. CONCEPTOS PREVIOS

El modelo de K-Nearest Neighbors consiste en medir las distancias a partir del punto que se desea clasificar, y dados los k puntos más cercanos entonces se contará cuál es el atributo dominante para así asignarlo a punto que se desea clasificar.

Algorithm 1 Descenso de gradiente

1: $N \leftarrow k$ vecinos más cercanos a $\vec{x}$
2: $c \leftarrow$ la clase más frecuente dentro de N
3: $\vec{x} \leftarrow c$

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

. Como podemos ver el algoritmo es bastante sencillo sin embargo queda bastante abierto en cuanto nos referimos a "vecinos más cercanos". El concepto de medida lo podemos describir de la siguiente manera.

$$f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$$

. Como podemos ver esta definición nos permite tener cualquier función de medida que nos sea conveniente de acuerdo al contexto. Para la práctica utilizaremos un caso especial de la distancia de Minkwoski que resulta en la distancia Euclideana.

$$L_p(\vec{x},\ \vec{y}) = \|\vec{x} - \vec{y}\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

$$p = 2 \Rightarrow L_p(\vec{x},\ \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3. METODOLOGÍA

Para el desarrollo de esta práctica se trabajo de manera colaborativa, para el desarrollo de los modelos solicitados. Específicamente trabajamos a la par uno consultado la documentación de las bibliotecas y los conceptos a utilizar mientras que el otro desarrollaba el script en Python. De esta manera logramos acabar en una cantidad de tiempo razonable.

4. RESULTADOS

4.1. Performance de K-Nearest Neighbors

Después de generar diferentes modelos para los datos dados, graficamos su precisión contra el k número de vecinos que se utilizan por modelo. De esta manera podemos determinar de cuál es la k más apropiada a utilizar.

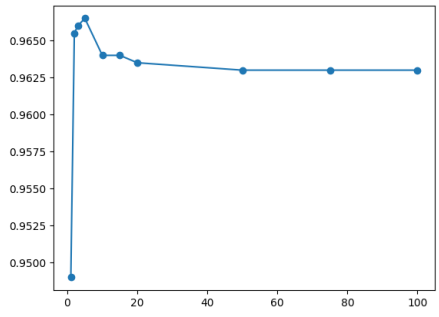

Figura 1: Dataset `Default.txt`

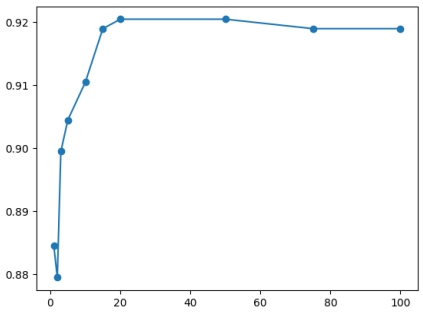

Figura 2: Dataset `genero.txt`

Como podemos ver para el dataset `Default.txt` obtiene la mejor precisión cuando $k = 5$. Mientras que en el dataset `genero.txt` hubo un empalme entre $k = 25$ y $k = 50$, por lo que se decidió utilizar el promedio $k = \frac{25+50}{2} \approx 37$.

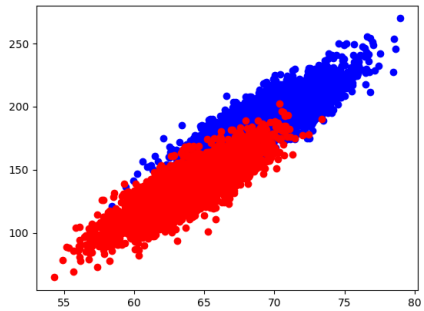

Figura 3: Dataset genero.txt

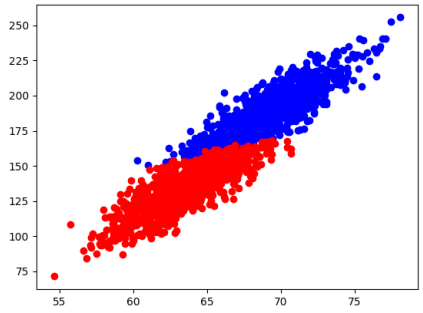

Figura 4: Clasificación por K-Nearest Neighbors

Como podemos ver la clasificación del modelo para el dataset de género es bastante cercano, dado que podemos ver que se conserva la forma general de los datos. Sin embargo se puede ver que en vez de que las clasificaciones se empalmen un poco como viene en el dataset, el clasificador segmenta dichas clasificaciones de manera más agresiva. Esto implica que puede haber casos donde falle el clasificador, que es de esperar con los modelos de predicción.

4.2. Comparación de clasificadores

Para esta sección haremos una comparación contra el modelo de regresión logística utilizado en la práctica previa. Primero compararemos las matrices de confusión para conocer cómo se está desempeñando los clasificadores con las entradas dadas.

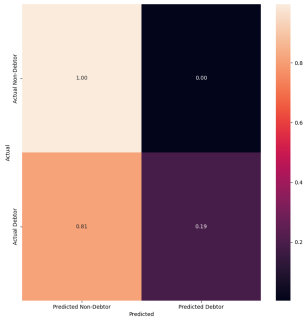

Figura 5: `Default.txt` por K-Nearest Neighbors

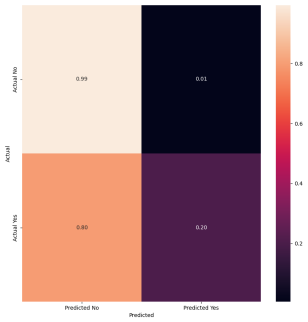

Figura 6: `Default.txt` por regresión logística

Como podemos ver obtenemos resultados bastante similares, aunque es posible ver que a ambos clasificadores les cuesta mucho trabajo clasificar cuando se trata de deudores. Sin embargo como el porcentaje de éstos es muy bajo entonces no le afecta mucho a la precisión del modelo.

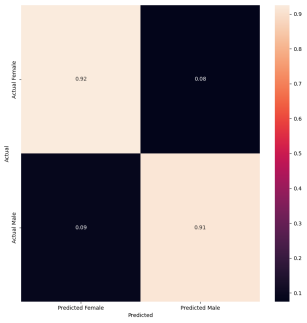

Figura 7: `genero.txt` por K-Nearest Neighbors

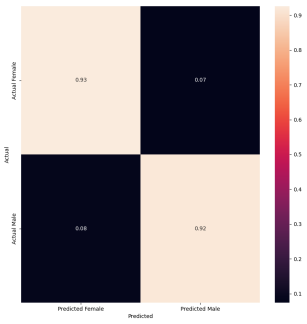

Figura 8: `genero.txt` por regresión logística

De manera similar podemos observar que ambos clasificadores logran trabajar con el dataset de manera muy buena, ya que tienen suficiente precisión y además el hecho de tener el mismo resultado por dos formas diferentes confirma en que estamos implementando modelos adecuados para describir el dataset.

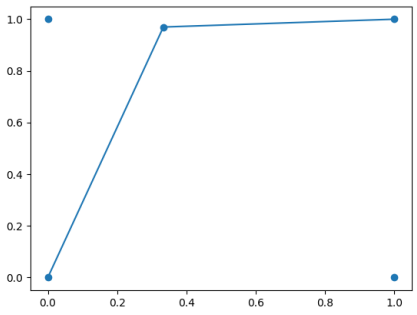

Figura 9: `Default.txt` por K-Nearest Neighbors

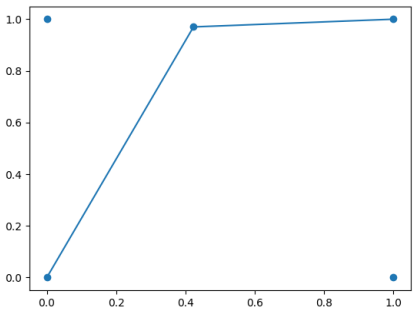

Figura 10: `genero.txt` por regresión logística

Podemos ver que en el espacio ROC el modelo por K-Nearest Neighbors resulta más ligeramente más efectivo en describir el dataset que el modelo de regresión logística.

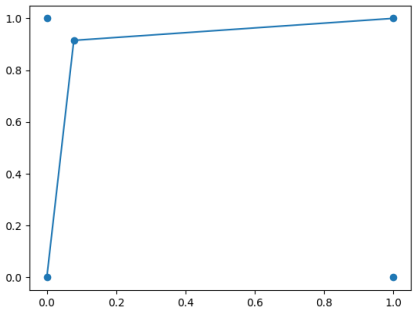

Figura 11: `genero.txt` por K-Nearest Neighbors

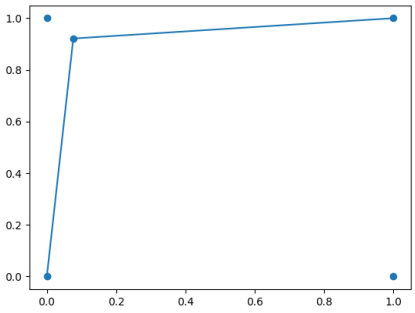

Figura 12: `Default.txt` por regresión logística

Mientras tanto el para dataset de `genero.txt` por una cantidad despreciable se puede decir que la regresión logísitca supera el modelo de K-Nearest Neighbors. Sin embargo esto es tan ligero que bien podría ser error de medición. Por lo que podemos ver que ambos modelos son adecuados para describir el dataset.

5. CONCLUSIONES Y REFLEXIONES

Jair Antonio. Como podemos ver el performance que tiene el modelo de K-Nearest Neighbors es bastante bueno, ignorando el tiempo que tarda en ejecutarse por consulta. Sin embargo dado que es muy cercano al modelo de regresión logística es posible que algunas aplicaciones donde sea crítico el performance sea más conveniente utilizar el segundo.

Maximiliano Zambada. Después de haber implementado el algoritmo de regresión logística, y haber comparado los resultados de esta y del de K - Nearest Neighbor, pude ver cómo ambos pueden obtener el resultado correcto, pero también se puede ver la diferencia de eficiencia, al ser mejor el de regresión logística, más que nada por el tiempo que se consume al utilizarlos.