Clustering usando modelo KNN

Alejandra Verónica López Chiquito[a]

[a]*Universidad Autónoma de Querétaro, Querétaro, México*

Abstract

El dataset que contiene una lista sobre los pasajeros que viajaban en el barco Titanic se ha convertido en una de las más populares en la rama de machine learning. En este ejercicio se utilizará dicho dataset para evaluar un modelo de clustering llamado k-nearest neighbors algorithm (KNN) donde se quiere crear grupos sobre los pasajeros que sobrevivieron o no, además se hará un analisis de los datos para poder encontrar relaciones entre los datos.

Keywords: Python, Clustering, KNN, Machine learning, Aprendizaje no supervisado,

1. Introducción

Una de las prácticas más comunes en machine learning o ciencia de datos es el clustering. Esta práctica es bastante útil para la visualización de los patrones que tienen los datos.

El clustering tiene bastantes ventajas, por ejemplo bajo costo computational, fácil implementación, mejora la visualizacióny comprensión de los datos, entre otros.

Existen muchas técnicas de agrupación o clusterización dentro de machine learning, podemos encontrar de aprendizaje supervisado o no supervisado. En este documento se explicará e implementará una técnica de aprendizaje supervisado que tiene por nombre K-Nearest Neighbors o KNN.

La implementación de un modelo KNN es usada para problemas de regresión o imputación de datos sin embargo su mayor utilidad es dentro de la clasificación ya que es muy fácil del implementar y contiene pocos hiperparámetros.

2. Definiciones

2.1. Voto mayoritario

Este criterio se basa es asignar al punto elegido en cada iteración al cluster con del 50 por ciento de votos de los vecinos elegidos para comparar.

2.2. Voto Plural

Este criterio se basa es asignar al punto elegido en cada iteración al cluster con mayor cantidad de votos de los vecinos elegidos para comparar.

2.3. Precisión (Acurracy)

Es una métrica bastante popular dentro del machine learning y utilizada para evaluar los modelos implementados. Es fácil de implementar en los modelos de aprendizaje supervisado ya que se basa en encontrar el porcentaje de aciertos contra el total de etiquetas, es decir, que entre mayor sea el accuracy obtenido, el modelo trabaja mejor. El parámetro que regresa normalmente es entre 0 y 1.

3. Marco teórico

El método de aprendizaje supervisado de clustering KNN es un modelo de clasificación basado en la proximidad de los vecinos más cercanos. Es decir que se logrará obtener una aproximación local y aunque es un método muy antigüo aún se pueden encontrar muy útil al aplicarlo a muchos problemas actuales.

Este método de agrupamiento o clustering es conocido por ser un método perezoso (Lazy learning), esto es porque se almacena solo un grupo de datos sin tener que pasar por un proceso de entrenamiento.

Los pasos que sigue el algoritmo de KNN es primeramente seleccionar el número de vecinos que serán todmados en cuenta para obtener el voto mayoritario. Después se calculará la distancia entre el punto que se quiere agrupar a los otros puntos. Como ya se mencionó se asignará un número k de vecinos cercanos y después de ordenar las distancias obtenidas, se tomaran los k vecinos para calcular el voto mayoritario y asignarselo al punto que se está tratando. Se realizara el mismo proceso para todos los puntos que se requiera saber su grupo o cluster correspondiente.

Para encontrar el número de vecinos cercanos más óptimo, el método más común es el método del codo (elbow method). Al final del ejercicio se utilizará una métrica para evaluar el modelo y poder definir el número de vecinos óptimo para este ejercicio.

4. Preporcesamiento de datos

Antes de implementar cualquier algoritmo de machine learning se analizan los datos y se 'arreglan' para poder mejorar el proceso de entrenamiento del modelo. A este proceso se le conoce como preprocesamiento de los datos. Las librerías que comúnmente se utilizan en este proceso son Pandas y Numpy, es por esto que en este ejercicio se utilizarán.

El dataset elegido para esta implementación, es una base de datos con información de los pasajeros que embarcaron en el

famoso barco Titanic. Se cuenta con 1309 instancias, 12 atributos (’PassengerId’, ’Survived’, ’Pclass’, ’Name’, ’Sex’, ’Age’, ’SibSp’, ’Parch’, ’Ticket’, ’Fare’, ’Cabin’, ’Embarked’) y la etiqueta será la columna ’Survived’. En esta columna únicamente se tienen 2 valores posibles, 0 para no sobrevivientes y 1 para sobrevivientes.

Además de esto se pueden tener más observaciones sobre el dataset. Al analizar las columnas se puede notar que se tuvo un error humano en la captura de los datos, esto se infiere debido a la descripción del data set en Kaggle. Los valores de la columna ’Embarked’ son 3, los cuales son S, C y Q. Posterior a analizar las columnas, se pude ver que en las columnas ’Fare’ y ’Cabin’ se encuentran los valores faltantes de la columna ’Embarked’.

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	S	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	S	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05	S	NaN
5	6	0	3	Moran, Mr. James	male	0.0	0	330877	8.4583	Q	NaN	NaN
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.075	S	NaN
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	S	NaN
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	C	NaN

Figure 1: Se indican ejemplos de datos que requieren asignarse a la columna correcta.

Antes de cualquier preprocesamiento se crea una copia del dataset para trabajar con ella y así evitar modificar el conjunto de datos original. Como primer paso se identifican los valores que pertenecen a la columna Embarked en las 2 columnas previas, para después borrarlas de ahí y colocarlas en la correcta.

Al terminar con estas acciones, se puede ver que las columnas ’Fare’ y ’Cabin’ terminan con muchos valores faltantes, sin embargo estas columnas de tarifa y cabina se pueden omitir en la implementción del modelo KNN ya que el precio que pagó un pasajero se puede ver relacionado con la clase en la que viaja, es decir que en su lugar se puede tomar la columna de ’Pclass’ para ver reflejado el valor monetario que pagó un pasajero para abordar. Y por otro lado, adicionalmente a que se infiere que la cabina del pasajero también está relacionada con el precio que pagó el pasajero se infiere que esta columna tiene poca relación con la etiqueta del pasajero. En las gráficas que se mostrarán más adelante podremos ver las relaciones que tienen las variables o columnas con la etiqueta de cada pasajero y basandose en estas gráficas se tomarán unicamente las columnas con mayor correlación.

Link del conjunto de datos: https://www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset

5. Metodología

Con ayuda de Matplotlib y seaborn, librerías del lenguaje python que se utilizan para crear gráficos apartir de listas o arrays, se graficarán las intersecciones entre las variables con el objetivo de encontrar las mejores relaciones entre ellas y de esta manera elegir las variables independentes para este modelo.

Utilizando esta tabla de correlación de Pearson, se puede notar que únicamente se tiene un valor grande de correlación en la variable sexo. sim embargo no se puede notar algún patron sobresaliente con otra variable, es por esta razon que se

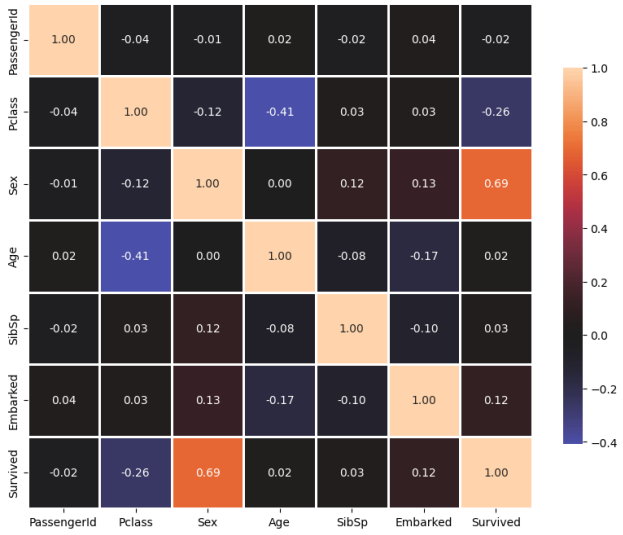

Figure 2: Correlación de Pearson entre las variables

toma la decisión de no descartar ninguna variable indpendiente. Las únicas columnas que se descartaron fueron ’Name’, ’Ticket’, ’Fare’ y ’Cabin’ por tener datos no numéricos o no tener relación lógica o influyente con la etiqueta de sobreviviente.

Una vez con toda información estudiada y limpia se puede iniciar la implmentación del modelo.

6. Implementación

Anteriormente se mencionó agrosomodo el funcionamiento del modelo KNN. Ahora se mencionarán más detalles sobre esta implmentación. Como primero punto es im portante mencionar que como se buscan aproximaciones locales en base a los vecinos más cercanos, se tiene que tener un parámetro con el cual saber si algún punto es un vecino cercano y este es la distancia entre 2 puntos. Para obtener este valor, se va a utilizar la formula del método Manhattan. Como se ve a continuación es la raíz de la suma del cuadrado de la resta de los puntos. De manera más fácil se puede apreciar visualmente en la figura 3.

$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Figure 3: Formula para calcular distancia con método Manhattan

Una vez que se tiene esta fómula, se aplicará iterativamente, es decir que se seleccionará un fila como punto y se calculará la distancia con el resto de los puntos para posteriormente guardar en una lista adicional el número k de vecinos más cercanos.

Obteniendo los vecinos, se obtendrá la etiqueta que tiene cada uno para aplicar la técnica de voto mayoritario y generar la nueva etiqueta.

Para evaluar si el redimiento de este algoritmo se obtendrá el valor de la predicción obtenido en un paso anterior. Para evaluar el modelo se aplicará la métrica mencionada anteriormente,

accuracy. Es decir que se toma el array que generó el modelo con las nuevas etiquetas y cuentan las coincidencias con las etiquetas reales, se dividen entre el número total de pasajeros y se obtiene el porcentaje de aciertos que tuvo el algoritmo de KNN.

Este proceso se implementó de manera manual, es decir programando el algoritmo desde cero y también usando librerías como apoyo, en este caso sklearn con el modulo KNeighborsClassifier. En esta última implementación se graficó el error cuadrático medio con cada número k de vecinos.

7. Conclusiones

Como se mencionó anteriormente, el algoritmo de K-Nearest Neighbors es muy útil como implementación de clustering, en este ejercicio se puede comprobar que este algoritmo genera resultados constanstes. Algunos modelos generan resultados esporadicos muy buenos pero no son estables, en esta ocasión no fue el caso, ya que al ser un modelo donde no se cuentan con tantos hiperparámetros por lo que los resultados son estables y confiables.

Con pruebas manuales si logró encontar que el número 4 de vecinos se obtiene el accuracy más alto que es de 0.8678 y posteriormente a esto se logró validar cuando se graficó el accuracy de todos los k números de vecinos.

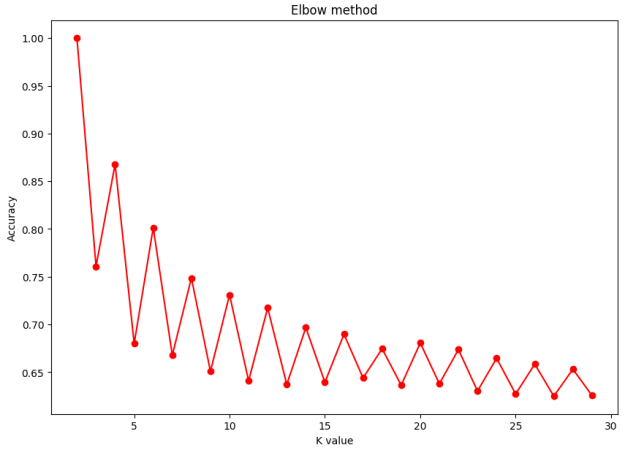

Figure 4: Formula para calcular distancia con método Manhattan

Las expectativas sobre este modelo eran bajas pero al realizar la investigación y sobretodo al ver el resultado de la implementación se puede concluir que es un método muy bueno y fácil de implmentar para problemas sin tanta complejidad y con pocos datos. Posteriormente se podrá probrar este algoritmo para problemas de regresión.

References

[1] IBM (2023) https://www.ibm.com/mx-es/topics/knn
[2] Joos Korstanje. (2012 - 2023) https://realpython.com/knn-python/
[3] Müller Andreas, Guido Sarah.(2016) Introduction to Machine Learning with Python. A Guide for Data Scientists. First Edition. O'Reilly
[4] McKinney, Wes. (2017) Python for Data Analysis. Second Edition. O'Reilly
[5] Sampaio, Cássia. (2023) https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/
[6] Scikit learn (2023) https://scikit-learn.org/stable/modules/generated/sklearn.neighbors
[7] Pandas Documentation (2023) https://pandas.pydata.org/pandas-docs/stable/reference/index.html