Alumno: Gerardo Hernández Nava
Correo electrónico: ghernandez100@alumnos.uaq.mx

K vecinos más cercanos (KNN) para la clasificación de sobrevivientes del Titanic.

Examen

I. Introducción.

El Titanic es uno de los barcos más famosos de la historia. Su construcción tardó 3 años, iniciando en marzo de 1909, pero bastaron 2 horas con 30 minutos para que el barco se hundiera a 600 km al sur de Terranova, Canadá, durante el cuarto día de su primer viaje. Donde murieron 1517 personas de los 2223 pasajeros y tripulantes de la nave [1], [2].

Dicho evento incentiva la selección de la base de datos *Titanic - Machine Learning from Disaster* [3] para este proyecto, el cual consiste en el diseño e implementación de un algoritmo de K vecinos más cercanos (KNN, por sus siglas en inglés) para la clasificación de las personas que iban abordo este trasatlántico y saber si son sobrevivientes o no de dicho naufragio. Además se realizó un análisis de componentes principales para saber cuáles eran los atributos más significativos para realizar la clasificación con éxito. También se utilizaron dos técnicas de validación, la lineal y la cruzada, para saber cúal brindaba un mejor entrenamiento.

Por otro lado, la clasificación es un proceso de categorización de un conjunto de datos entre un número limitado de clases, en este caso sería sobreviviente o no sobreviviente.

Cabe mencionar que el algoritmo de K vecinos más cercanos forma parte de los algoritmos de aprendizaje supervisado de aprendizaje máquina.

La base de datos ya mencionada es el registro 891 pasajeros, con 12 atributos, uno de ellos es el decisivo - que nos indica si el pasajero fue sobreviviente o no - los otros once

dan información como el nombre del pasajero, cabina, número de boleto, lugar donde abordó, edad, género, clase a la que pertenecía, entre otros.

II. Objetivo

El objetivo principal de esta práctica es implementar el algoritmo de KNN, el cual será probado con la base de datos de los sobrevivientes del Titanic, haciendo uso de la validación de k-fold y la validación en línea.

III. Marco teórico.

A. Algoritmo de K vecinos más cercanos

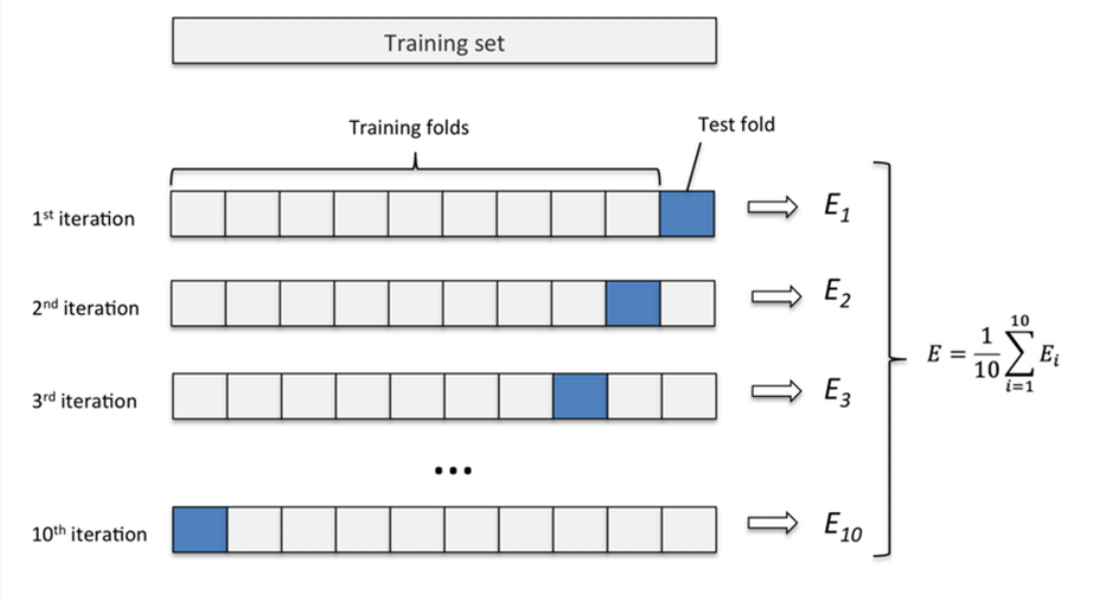
El algoritmo de KNN es un algoritmo que pertenece a los algoritmos de aprendizaje supervisado. Como primera etapa se debe seleccionar un número k de vecinos, posteriormente se calcula la distancia de cada uno de los puntos hacia todos los demás. Se toman los k vecinos màs cercanos y se le atribuye al punto que se está analizando la clase más frecuente en los vecinos que se analizan. Finalmente se selecciona el mejor número de vecinos.

B. Análisis de componentes principales

El análisis de componentes principales es una técnica estadística con la cual se busca la reducción de dimensionalidad y la minimización de la pérdida de la información. Donde cada uno de los atributos que se mantienen toman el nombre de componentes principales.

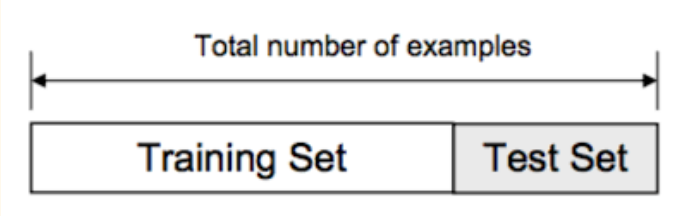
C. Validación cruzada.

Existen situaciones en las que no se cuenta con un gran número de datos para entrenamiento y prueba. La validación cruzada permite dividir el conjunto de datos, de tal forma que, los datos se pueden utilizar en el entrenamiento y en prueba. Específicamente en la validación cruzada de k iteraciones (k-fold cross validation) se divide el conjunto de datos original en subconjuntos, de modo que, durante el entrenamiento se toma cada k subconjunto como un conjunto de prueba total, mientras que los demás subconjuntos se toman como entrenamiento, esto se repetirá k veces, y cada vez el conjunto de prueba será diferente y el resto de los datos se utilizará para entrenamiento [4]. En cada iteración se busca el mayor valor de precisión y se elige la iteración que haya tenido el mejor desempeño.



D. Validación en línea

Este método consiste en la creación de subconjuntos, en los cuales el 80% de los datos se considera para el entrenamiento, mientras que un 20% es para las pruebas del modelo de clasificación. Sin embargo, estos porcentajes pueden variar según las necesidades de la base de datos y el proyecto.



IV. Metodología

1. Se importaron las librerías necesarias para poder implementar el algoritmo, como numpy, matplot, pandas, math, seaborn collections y tqdm.
2. Se comienzan a declarar las funciones en las cuales se va a basar nuestro algoritmo. Las primeras dos funciones son requeridas para realizar nuestro análisis de componentes principales.

   La primera de ellas es para obtener la covarianza, fue denominada de la misma manera y recibe como parámetros dos vectores, donde

cada uno corresponde a un atributo de nuestra base de datos. Se calcula la covarianza entre ellos, mediante la siguiente fórmula, y se regresa el valor de covarianza entre ellos.

$$cov_{xy} = \frac{\Sigma_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

La segunda función denominada *matriz_covarianza* hace uso de la función descrita anteriormente, mandando poco a poco los atributos para que de esta forma quede como resultado una matriz cuadrada que nos indica la covarianza que existe entre cada uno de los atributos.

Posteriormente se realizaron las funciones que nos ayudaron a dividir la base de datos en conjuntos de entrenamiento y de prueba.

Se inició con la función para la validación lineal. La función se denominó *split_80_20*. Como lo dice su nombre ayuda a dividir el conjunto de datos en dos, la primera parte corresponde a un 80% de los datos de que se tienen y serán utilizados para el entrenamiento del modelo, el restante 20% se asigna para la prueba. Pero antes de dividir el conjunto de datos se crea una vector con los índices de los datos y se aleatoriza, posteriormente se divide este arreglo y para finalizar se extraen los datos de X y Y según los indicen que se tengan en cada arreglo. La función recibe los datos X y sus etiquetas Y.

Posteriormente se hizo una función denominada *split_k_fold*. Esta función recibe como parámetro el conjunto de datos X y el número de folders en los que se quiere dividir el data set. Ya que se recibieron los parámetros se hace un vector con los índices de X y se aleatorizan. Se calcula el número de elementos que se deben tener por folder y se realizan dos matrices de dimensiones m*n donde m es el número de folders y n el número de elementos ya sea en el conjunto de entrenamiento o en el conjunto de prueba. Se van llenando dichas matrices según los índices que le corresponden a cada conjunto.

Se continuó con las funciones para calcular las cifras de mérito (exactitud, precisión, especificidad, sensitividad, puntaje F1 y error cuadrático medio). Todas las funciones reciben las etiquetas asignadas por el algoritmo y las etiquetas reales.

Para finalizar se realizaron las funciones para el algoritmo KNN.

Se inició con la función para calcular la distancia euclidiana, la cual es utilizada para conocer la distancia de un punto dado hacia todos sus vecinos y hacer un posterior análisis con la función *pertenencia*.

La función *pertenencia* indica a qué clase se asocia el punto analizado, esta decisión se toma con la clase a cual pertenece la mayoría de los vecinos más cercanos según la distancia euclidiana.

Posteriormente se realizó la función con la cual se entrenó el modelo. Primero se calcula el número de clases existentes en el dataset, el número de instancias y se crea una matriz donde se guardaran las exactitudes dado un número n de vecinos. La matriz se va llenando mediante dos ciclos for anidados, el primero de ellos recorre el número de vecinos que se desean tomar en cuenta y el segundo cada unos de los puntos. Ya que se terminó de llenar la matriz se calcula máxima exactitud para poder determinar el mejor número de vecinos a tomar en cuenta. Está función regresa todas las exactitudes calculadas, la mejor y el número de vecinos que se tomaron en cuenta.

Para finalizar con las funciones se realizó una función para clasificar los datos de prueba. Donde se calculan las distancias de cada uno de los puntos y se le asigna una etiqueta según los vecinos determinados como los mejores en el entrenamiento.

3. Se importó la base de datos, se calcularon las dimensiones, se imprimieron las primeras 5 líneas y se suman el número de datos faltan por atributo, para conocer un poco más de la base de datos.
4. Se eliminan los atributos que no proporcionan gran información como el nombre, el boleto entre otros o que tenga un alto número de datos faltantes como la cabina.

5. Se modifican los atributos con valores alfabéticos a valores numéricos, como el sexo y el lugar de embarcación.
6. Se grafica la distribución de los atributos para que al momento de la imputación verificar que no se haya modificado significativamente la distribución de los atributos.
7. Se realiza una imputación por media para los atributos que presentan datos faltantes.
8. Se graficaron los atributos donde se realizó la imputación para verificar lo antes mencionado.
9. Se normaliza la base de datos, para que las observaciones se encuentren entre cero y uno.
10. Se ejecuta la función matriz_covarianza ingresando como parámetro la matriz X menos el la media de cada atributo.
11. Obtenemos los eigenvalores de la matriz de covarianza y calculamos el porcentaje de cada unos atributos y así determinar cual de los atributos proporcionan mayor información y cuántos deben ser tomados en cuenta.
12. Se eliminan todos los atributos que no se tomarán en cuenta para la clasificación de la matriz de atributos X.
13. Se dividen los datos en conjuntos de entrenamiento (80%) y conjuntos de prueba(20%).
14. Se llama la función de entrenamiento y se obtienen el mejor número de vecinos
15. Para finalizar se manda el conjunto de prueba a la función clasificar y se calculan las cifras de mérito.
16. Se realiza el entrenamiento, prueba y la medición de las cifras de mérito con la validación cruzada con 5 y 10 folders o subconjuntos.

V. Resultados

A continuación, se muestran los resultados obtenidos a lo largo de la implementación del algoritmo KNN.

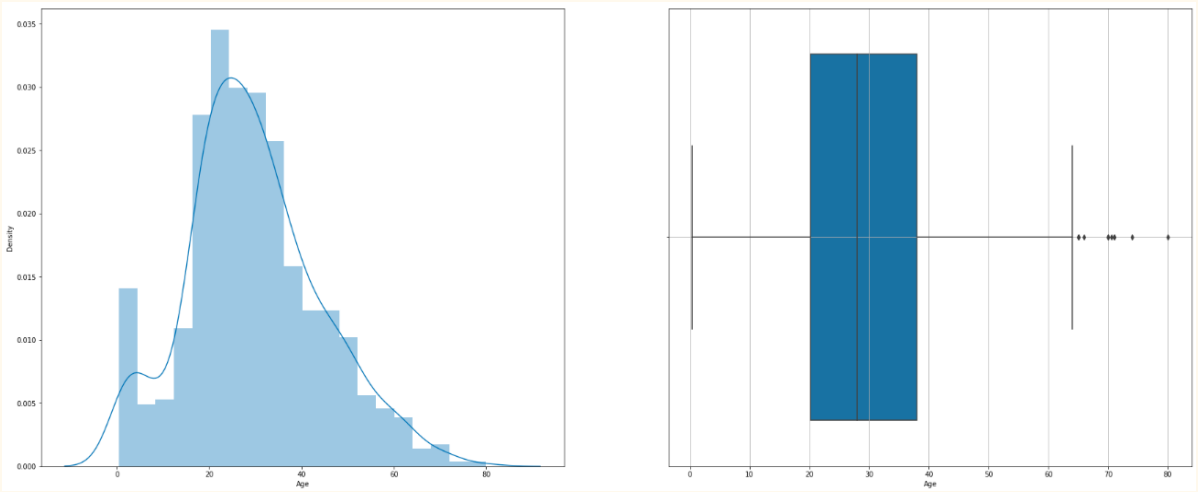
V.I PCA

En la Tabla 1 se muestran los atributos que contiene la base de datos así como el número de datos nulos que se encuentran.

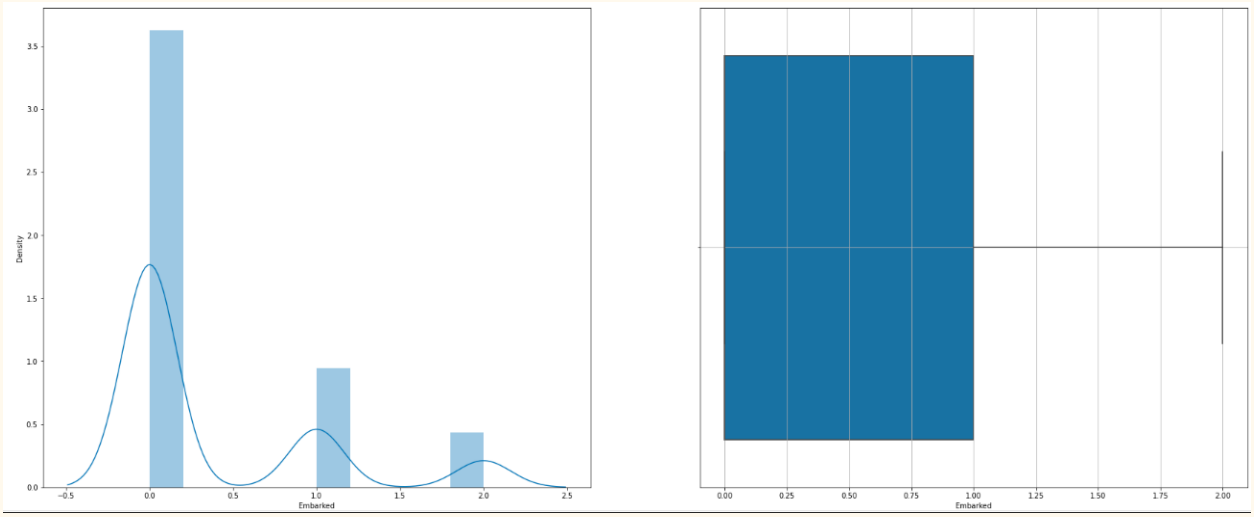
Tabla 1. Atributos y datos nulos en la base de datos.

Atributo	Nulos
PassengerId	**0**
Survived	**0**
Pclass	**0**
Name	**0**
Sex	**0**
Age	**177**
SibSp	**0**
Parch	**0**
Ticket	**0**
Fare	**0**
Cabin	**687**
Embarked	**2**

En la Ilustración 1, se muestran los atributos de Edad y de lugar de embarcación antes de realizar la imputación por medias.



a)



b)

Ilustración 1. La ilustración a) corresponde a la distribución del atributo edad, mientras que la ilustración b) corresponde a la distribución de los lugares de embrcación.

La Ilustración 2 corresponde a la distribución de estos mismos atributos pero una vez realizada la imputación por medias.

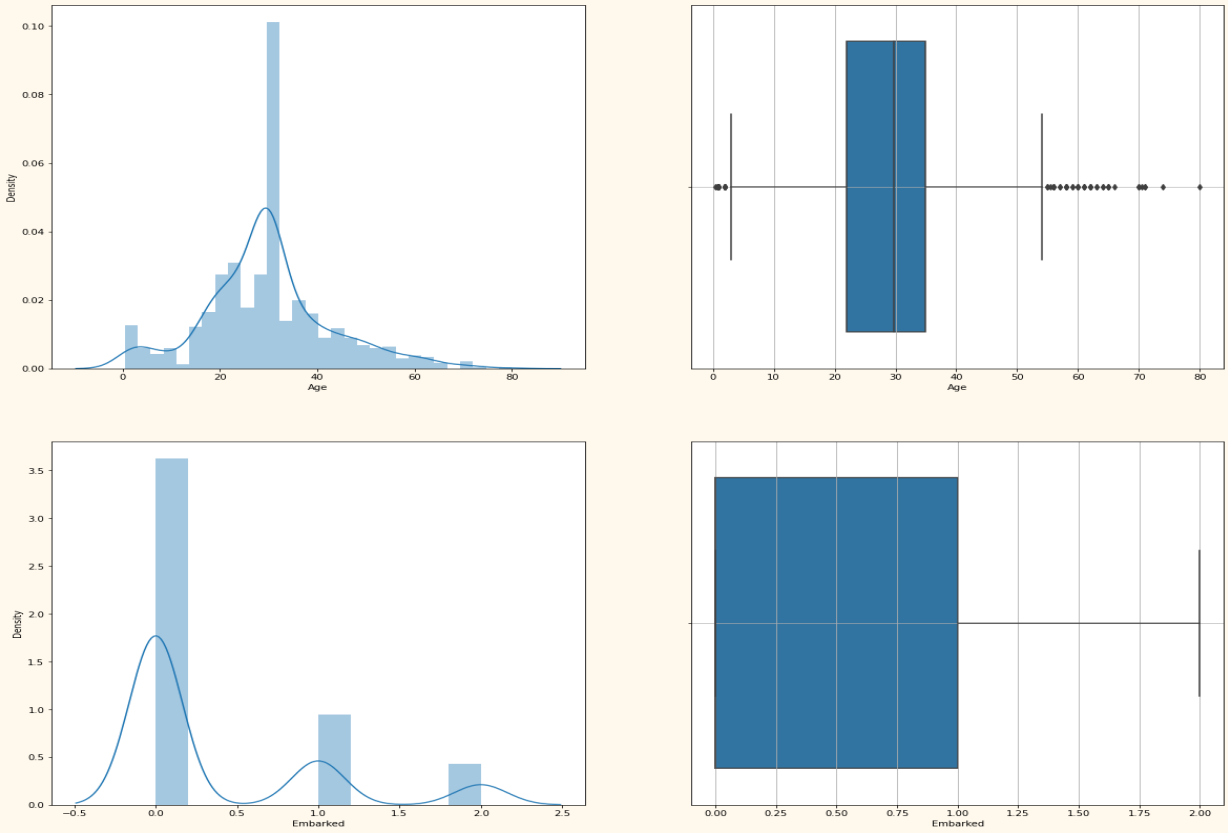


Ilustración 2. Distribución de los atributos después de la imputación por medias.

Se realizó una matriz de correlación como parte del proceso de PCA, en la Ilustración 3. se puede observar la matriz de correlación con un mapa de calor.

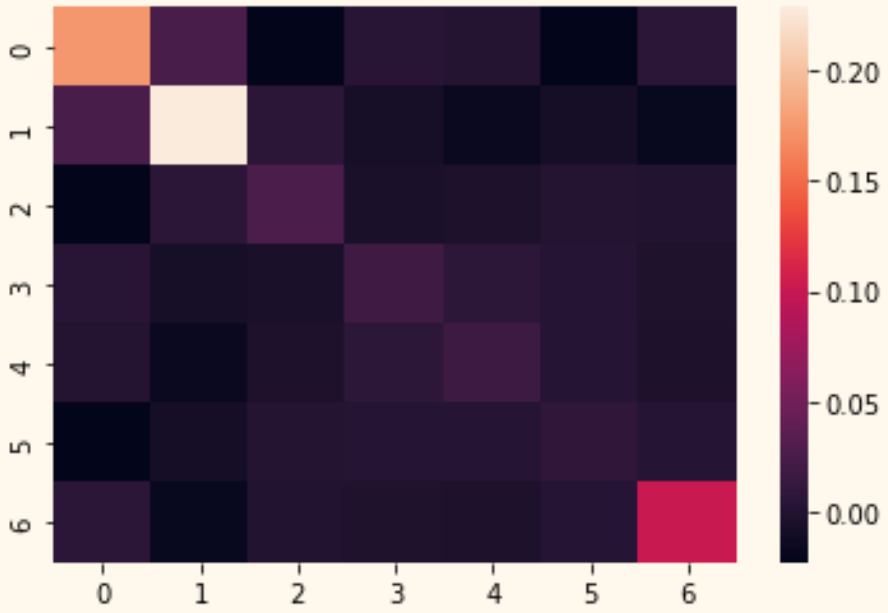


Ilustración 3. Mapa de calor de la matriz de correlaciones.

Calculando los eigenvalores de cada uno de los atributo se obtienen los siguiente valores: clase del pasajero = 42.0589233, género = 29.85370112, edad = 17.04686433, = 5.00491112, = 3.24547827, = 0.95680078, lugar de embarcación = 1.83332108.

Ya que se deseaba trabajar con un porcentaje mayor al 90% de la información se tomaron en cuenta los primeros cuatro atributos, con los cuales se conservó el 93.96% de la información.

Ya que se decidió sobre los atributos que se iban a conservar se eliminaron del conjunto de información y se realizó un análisis del comportamiento de cada uno de los atributos en pares. Dicho análisis se puede observar en la Ilustración 4.

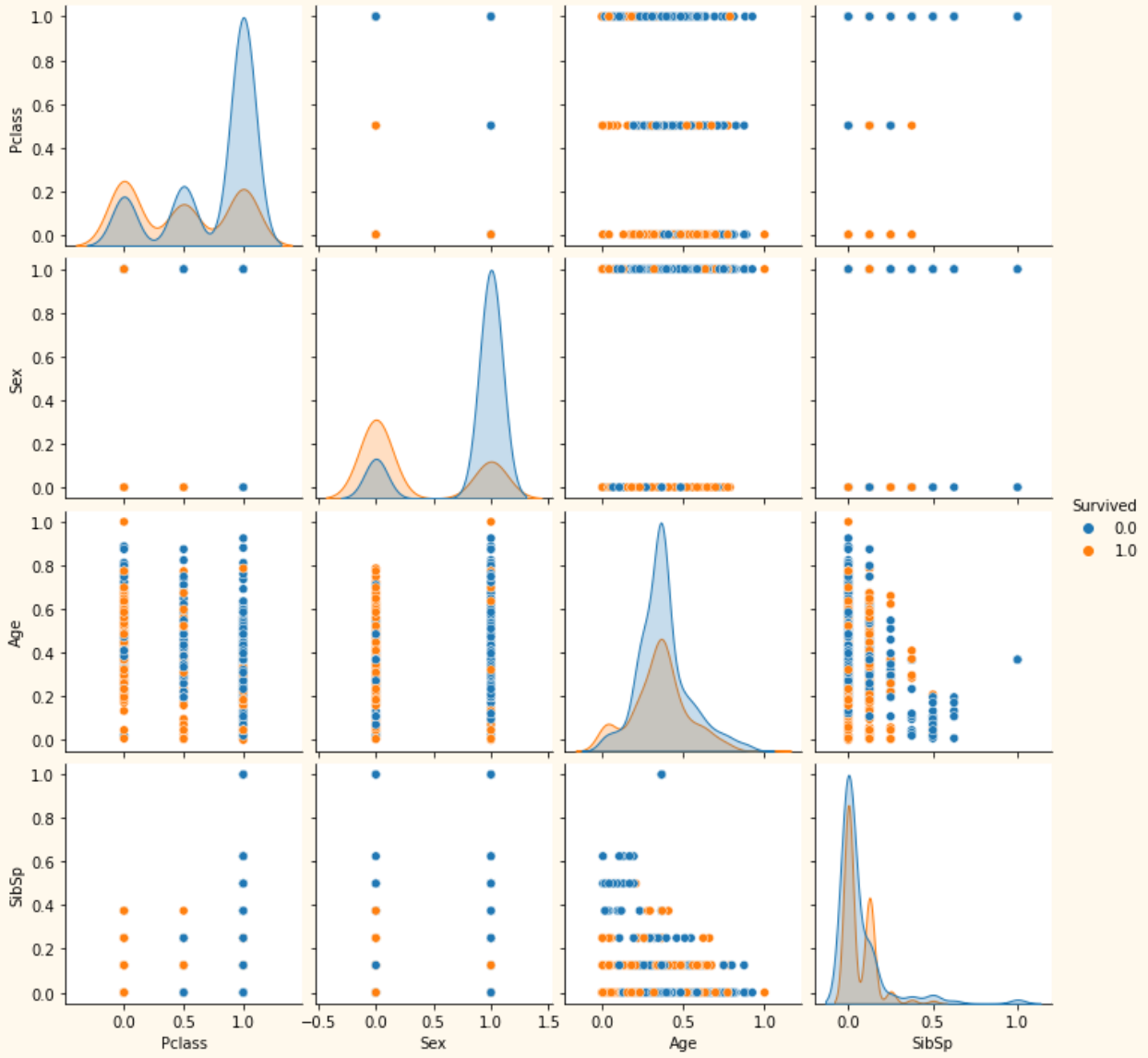
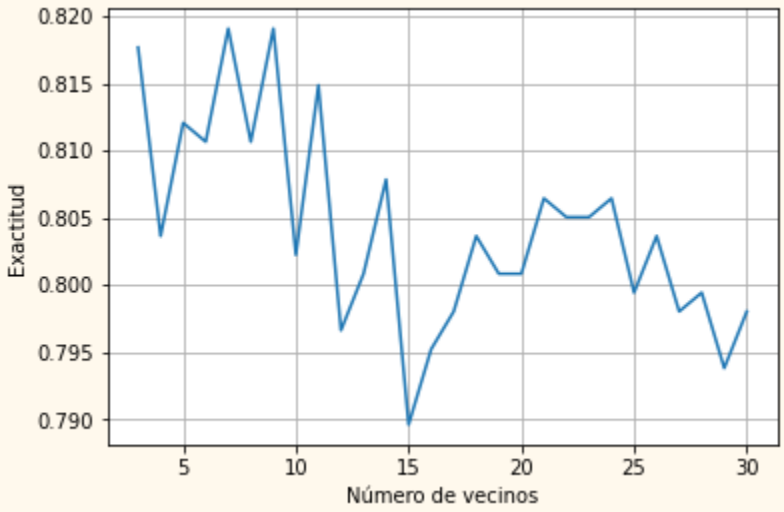


Ilustración 4. Distribución de los atributos que se tomaron en cuenta para la clasificación de sobrevivientes del Titanic.

V.II Validación lineal.

Una vez realizado el entrenamiento mediante la validación lineal con el 80% de las instancias se obtiene que el mejor número de vecinos es de 7, esto se puede corroborar en la Gráfica 1 donde se muestran el número de vecinos que se ocupan contra la exactitud que se obtiene con dicho número de vecinos.



Gráfica 1. Número de vecinos vs exactitud de clasificación de sobrevivientes del Titanic.

La evaluación del modelo diseñado nos indica que se tienen las siguientes cifras de mérito para la validación lineal:

Exactitud: 0.8146067415730337
Precisión: 0.8909090909090909
Sensitividad: 0.6447368421052632
Especificidad: 0.9411764705882353
Puntaje F1: 0.7480916030534351
Error cuadrático medio: 0.09269662921348315

V.III 5-fold

La resultados arrojados por la validación de 5-fold, determinaron que se obtuvieron las siguientes cifras de mérito:

Exactitud: 0.805586592178771
Precisión: 0.7841872739841911
Sensitividad: 0.6980685980685981
Especificidad: 0.8769034305327933
Puntaje F1: 0.7327781995796088
Error cuadrático medio: 0.09720670391061452

Los mejores vecinos obtenidos en cada una de las iteraciones con los siguientes: 4, 4, 3, 3 y 5 respectivamente.

V.IV 10-fold

En el caso del entrenamiento de la validación 10-fold nos indica que el mejor número de vecinos a considerar son los siguientes: 4, 5, 8, 3, 3, 3, 8, 3, 4 y 4.

De esta forma se obtienen las siguientes cifras de mérito:

Exactitud: 0.8099999999999999
Precisión: 0.7993171612975725
Sensitividad: 0.6843202888791124
Especificidad: 0.8883495038170544
Puntaje F1: 0.7298240369257198
Error cuadrático medio: 0.095

VI. Conclusiones

A partir del preprocesamiento de datos realizado: reducción de dimensiones del conjunto de datos eliminando características que no atribuían gran información en la toma de la decisión (clasificación), codificación de la información alfabética a numérica y normalización de los datos para evaluar el algoritmo KNN con tres distintos métodos de validación se puede concluir que en esta ocasión el tipo de validación o de entrenamiento no influyó en gran medida el rendimiento del algoritmo, ya que las diferencias entre las cifras de mérito son mínimas y no existe una donde un mètodo predomine.

Referencias.

[1] “El Titanic, la historia completa del barco insumergible.” https://historia.nationalgeographic.com.es/a/historia-titanic-tragedia-barco-insumergible_16344 (accessed May 06, 2022).

[2] “40 Fotos de Titanic.” https://www.nationalgeographic.com.es/temas/titanic/fotos (accessed May 06, 2022).

[3] “Titanic - Machine Learning from Disaster | Kaggle.” https://www.kaggle.com/c/titanic (accessed May 06, 2022).

[4] L. Laura-Ochoa, “Evaluation of Classification Algorithms using Cross Validation,” Ind. Innov. Infrastruct. Sustain. Cities Communities, pp. 24–26, 2019, doi: 10.18687/LACCEI2019.1.1.471.].