Inteligencia Artificial para la Ciencia de Datos

Reporte final de “Titanic: Machine Learning from Disaster”

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Integrantes:

Yolanda Elizondo Chapa - A01137848

Daniel Salvador Cázares García - A01197517

Angel Corrales Sotelo - A01562052

Izael Manuel Rascón Durán - A01562240

Yoceline Aralí Mata Ledezma - A01562116

Índice

Introducción

El 15 de abril de 1912, durante su primer viaje, el RMS Titanic se hundió tras chocar con un iceberg. Desgraciadamente, no había suficientes botes salvavidas para todos, por lo que murieron 1,502 de los 2,224 pasajeros. Aunque se podría pensar que hubo cierto grado de suerte en la supervivencia, parece que algunos grupos de personas tuvieron más probabilidades de sobrevivir que otros.

Por lo que, a través de este proyecto se creó un modelo de Machine Learning que pudiera predecir ¿Qué personas tenían más probabilidades de sobrevivir? usando datos de los pasajeros como nombre, edad, género, ticket, clase, etc. proporcionados por el reto de Kaggle “Titanic - Machine Learning from Disaster”.

Análisis de los resultados

El proyecto se dividió en 4 fases principales: Análisis de los datos, preparación de los datos, entrenamiento del modelo y evaluación del modelo. Los cuales fueron muy útiles para lograr un buen desempeño del modelo, ya que sin la comprensión de las variables proporcionadas y la manipulación de los datos para atacar problemas como datos atípicos o datos nulos, así como el ajuste de los hiperparámetros el desempeño del modelo hubiera sido gravemente afectado.

Análisis de los datos

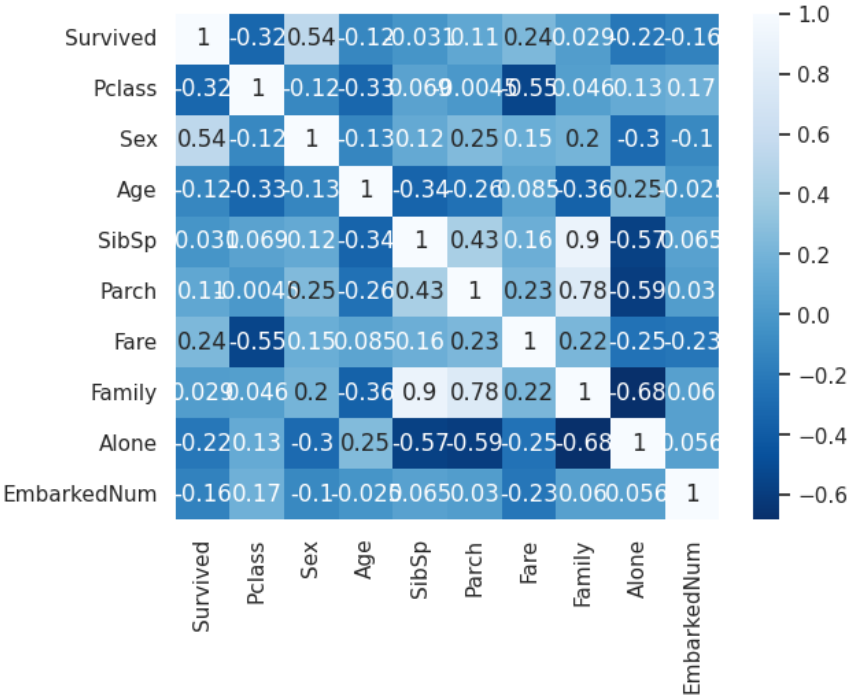
Esta fase se centró en conocer y sacar el mejor provecho de la base de datos la cuál se dividió en tres conjuntos de datos: Entrenamiento con 712 registros, validación con 179 registros y prueba con 418 registros. Cabe resaltar que los tres conjuntos de datos contienen la misma información de variables y fueron limpiados de igual manera.

Las variables fueron las siguientes:

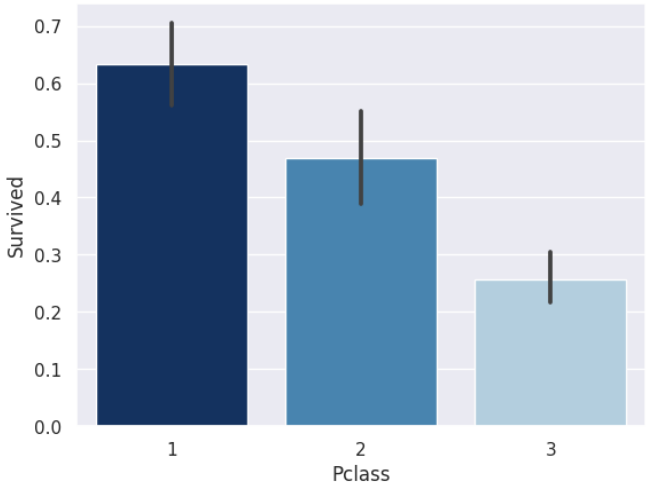
PassengerId	Categórico	Integer
Survived	Categórico	Integer
Pclass	Categórico	Integer
Name	Categórico	String
Sex	Categórico	Integer
Age	Numérico	Float
SibSp	Numérico	Integer
Parch	Numérico	Integer

Ticket	Categórico	String
Cabin	Categórico	String
Fare	Numérico	Float
Embarked	Categórico	String

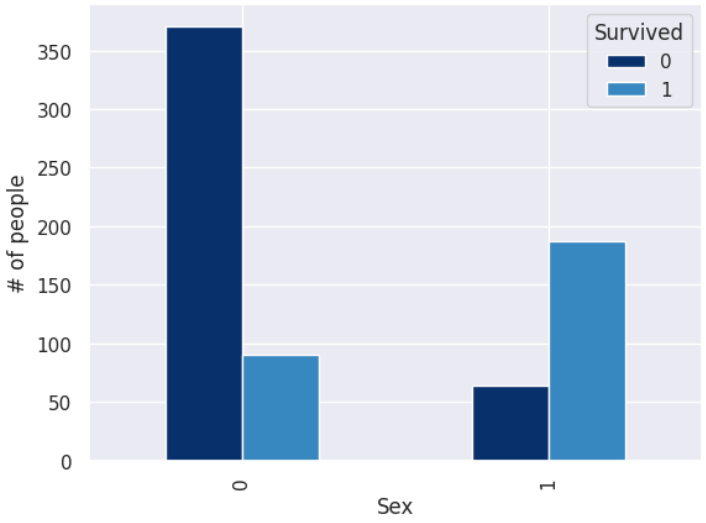
* A la variable Age le faltan 148 datos, a Cabin 556 y Embarked 2.



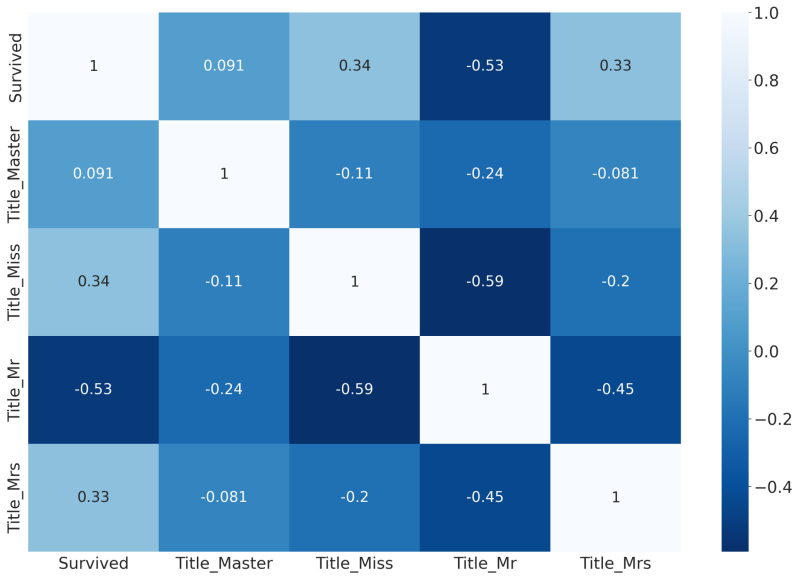
- Se observó que la variable Pclass tiene una correlación negativa moderada con la variable objetivo (Survived). También se encontró una clara diferencia entre la cantidad de sobrevivientes y la clase a la que pertenece:
    - 1ra clase: Mayor porcentaje de sobrevivientes con 0.629
    - 2da clase: Segundo mayor porcentaje de sobrevivientes 0.472
    - 3ra clase: Menor porcentaje de sobrevivientes con 0.242



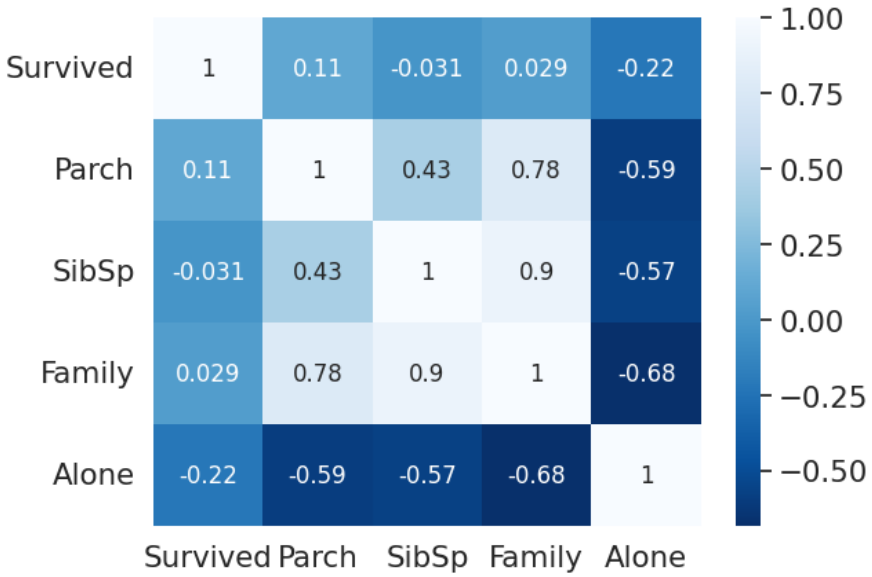
- Respecto a la variable sex se observó una correlación con la variable objetivo ya que el 81% de los hombres y el 25% de las mujeres fallecieron. Lo que indica que el sexo fue un factor importante en la supervivencia, pues si eras mujer era más probable que sobrevivieras.



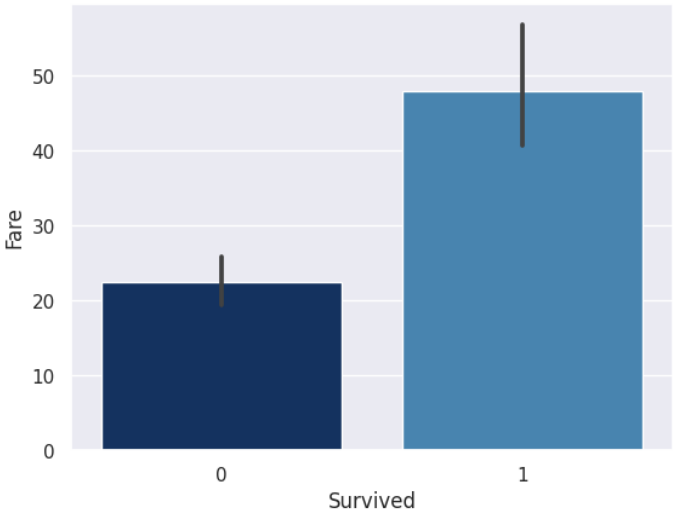
- Debido a que era muy difícil obtener información respecto a la variable name, se decidió extraer los títulos y a partir de esos se descubrió que tienen una correlación relevante con la supervivencia (excepto "Master"), pues la correlación de estos con supervivencia fue mayor a 0.34. Lo cual tiene sentido, pues los títulos eran puestos a partir del sexo y la edad.



- Con la variable SubSp y Parch no se observó una correlación fuerte por lo que se intentó tener una mayor correlación creando dos variables Family y Alone. Family especificaba el número de familiares a bordo que tenía el pasajero (es decir, SibSp + Parch) y la variable Alone simplemente decía si el pasajero tenía o no un familiar abordo (SibSp + Parch == 0 | SibSp + Parch > 0). Finalmente se encontró que Alone tenía una mayor correlación.



- Cabin tiene 556 datos faltantes por lo que no se encontró que esta variable fuera importante y fue excluida en el modelo.
- Respecto a Fare se encontró que quienes sobrevivieron más, pagaron una tarifa mayor en comparación a aquellos que no sobrevivieron.



-

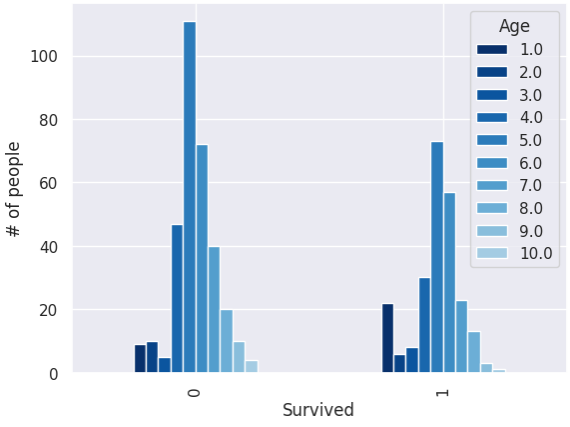
-

Las variables que finalmente se seleccionaron fueron: sex, fare, Pclass, alone, title, age y embarked.

Preparación de los datos

Se realizaron las siguiente modificaciones a la base de datos (conjunto de entrenamiento, validación y prueba) para lograr un buen desempeño en el modelo:

- Se obtuvieron variables dummies de título.
- Se creó la variable “Alone”, utilizando los datos de Parch y Sibsp, y se transformó a binaria.
- La variable sexo se transformó a binaria.
- Se separaron edades por intervalos observando la distribución de la edad en cuanto a la supervivencia.



- La variable Embarked se cambió a numérica.

Para el manejo de valores faltantes se eliminaron reglones donde Embarked = NaN y se rellenaron edades faltantes con base al sexo, título y clase. Por otra parte, se realizó un escalamiento a todas las variables con StandardScaler(), para así tener todas las variables en una misma escala y evitar de alguna manera el sesgo que el modelo pudiera tener al dar preferencia a una variable numérica que tiene mayor escala. Por último, se eliminaron las columnas no utilizadas (Cabina, Ticket, SibSp, Parch, PassengerId y Name).

Elección de modelo

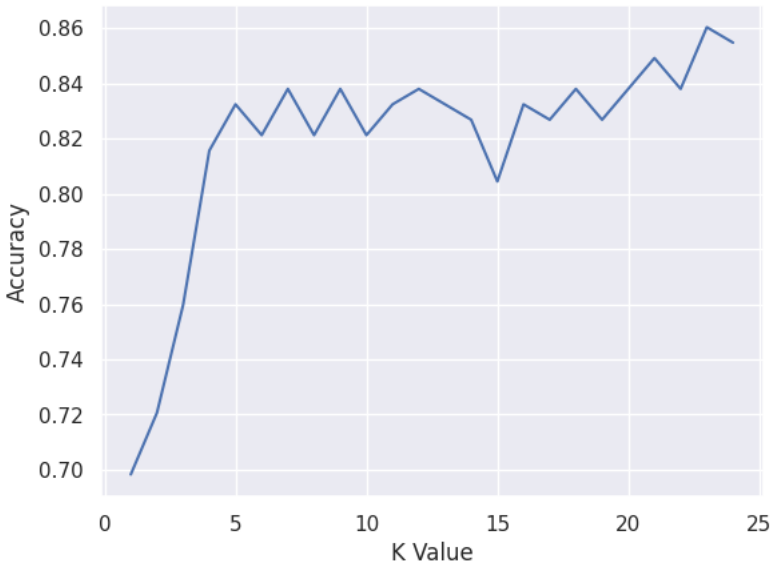
El reto presentado es un problema de clasificación por lo tanto se realizaron pruebas con los siguientes modelos, ya que suelen tener mejor desempeño en problemas de clasificación, para así seleccionar el mejor de acuerdo al accuracy. Las siguientes pruebas y ajuste de hiperparámetros fueron realizadas con el conjunto de validación.

- Regresión logística, con el cual se logró una exactitud o accuracy de 0.83
- Random Forest Regressor, con el cual se logró una exactitud o accuracy de 0.78
- Random Forest Classifier, con el cual se logró una exactitud o accuracy de 0.79
- Support Vector Machine, con el cual se logró una exactitud o accuracy de 0.82
- KNN (K-nearest Neighbors), con el cual se logró una exactitud o accuracy de 0.86

Tras comparar los 5 modelos, se eligió KNN (K-nearest neighbors) debido a que tuvo una mayor exactitud. Para este modelo se probaron diferentes hiperparámetros como n_neighbors, p y leaf_size con GridSearchCV, finalmente se obtuvo que n_neighbors = 23 tuvo el mejor desempeño.

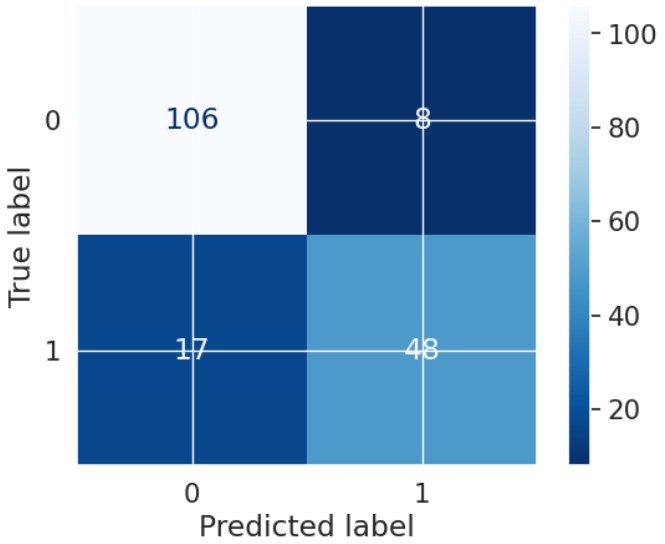
Regularización y ajuste de hiperparámetros

Debido a que el algoritmo K-nearest neighbors es no paramétrico, no es posible regularizarlo. Sin embargo, se realizó el ajuste de hiperparámetros utilizando la técnica GridSearchCV en donde se prueban distintas combinaciones de hiperparámetros, en este caso, los hiperparámetros probados fueron n_neighbors, lead_size y p. Finalmente, se encontró que el modelo que funcionaba mejor fue el que tenía n_neigbors = 23 y los demás hiperparámetros con el valor predeterminado.



Evaluación del modelo

El modelo KNN con una K óptima de 23, tuvo la siguiente matriz de confusión donde se observa que se obtuvieron 8 falsos positivos, 17 falsos negativos, 106 verdaderos positivos y 48 verdaderos negativos.



Dentro de la competencia Kaggle con el conjunto de prueba se obtuvo un puntaje final de .7799.

Referencias bibliográficas

Gong, D. (2022, July 12). *Top 6 machine learning algorithms for classification*. Medium. Retrieved September 17, 2022, from https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501

Anexos

Repositorio del proyecto
https://github.com/dscazares/Solucion-Titanic-Kaggle