1. Dataset Description

The titanic dataset is a dataset that contains information of some passengers on the titanic, a ship that sank in 1912 after hitting an iceberg. The dataset can be used to predict whether a passenger survived or not based on variables such as age, class, gender, etc. The description of each column are as follows:

1. Pclass: Represents the socio-economic class of each passenger on the ship
2. Survived: Represents whether the passenger has survived or not (1 for survived and 0 for not survived)
3. Name: The name of the passenger
4. Sex
5. Age
6. SibSp: The number of siblings and spouses of the passenger
7. Parch: The number of parents and children of the passenger
8. Ticket: The ticket id
9. Fare: The amount paid for the ticket
10. Embarked: The port from which the passenger boarded the titanic
11. Cabin: The cabin occupied by the passenger
12. Dataset preprocessing

A screen shot of a computer program

Description automatically generated

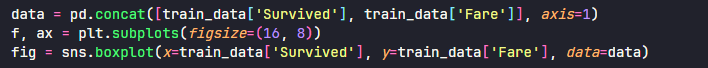
Here we imported some visualization libraries that will be used later on, and imported pandas to get the csv as dictionary in python.



Then used pd.read\_csv to read the csv file and store it in train\_data variable, and then we invoked train\_data.info to gain some insights about the data (the number of null rows)

A screenshot of a computer

Description automatically generated

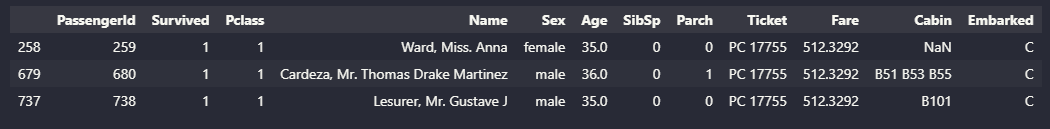


Then we started our data cleaning by gathering out numerical features and searching for the outliers, and we found only one extreme at a fare that is almost equal to 500, as shown below, note that some values we did not consider as outliers as there were too many rows denoted by the box plot, and removing them would reduce out training samples, so we decided to remove only the most extreme point as it was only one

A graph with different colored bars

Description automatically generated with medium confidence





Here we get all passengers who paid fares greater than 500, we get three rows with fares greater than 500, and we replace those fares with the mean of those fares (calculated without the outliers) as shown below

A computer code with text

Description automatically generated

Now we will check if there is still null rows

A screenshot of a computer

Description automatically generated

As we see there are 177 rows in age column are empty, so we are going to replace the NaN values with the mean age values, and as for the cabin we are going to drop this column, and also drop any other irrelevant feature

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

Now we are going to use MinMaxScaler to normalize our columns to be in the range [1,5] since there are many columns with varying ranges so we need to normalize them first

A computer code with colorful text

Description automatically generated with medium confidence

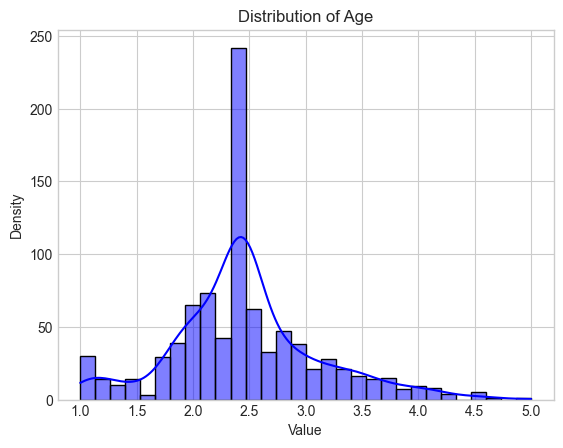
A screenshot of a computer screen

Description automatically generated

Now we will check the distribution of the data, since some models may work best (or not even affected) when the data is normally distributed.

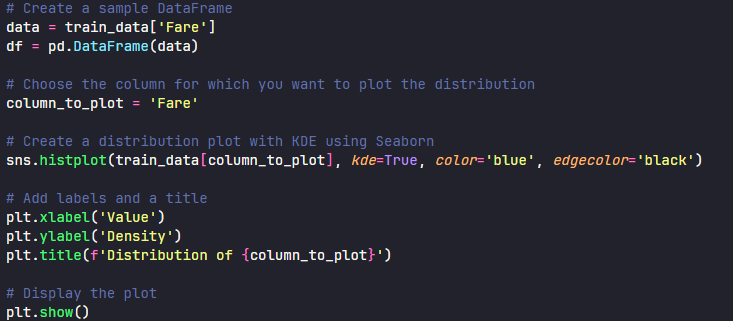
A screen shot of a computer program

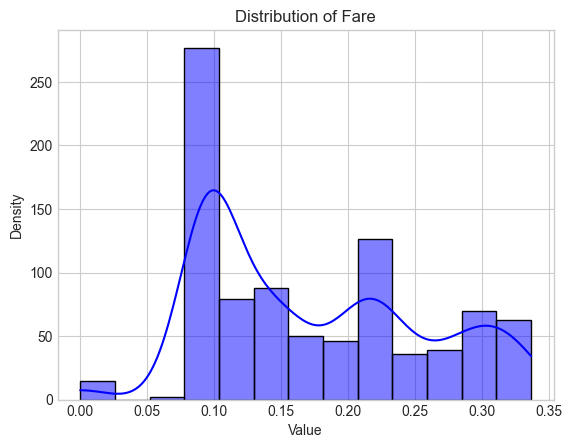
Description automatically generated



The Fare was not normally distributed, so we did apply box cox transformation to try to somehow make the data normally distributed, this the result after applying box cox transformation.







Finally, we applied PCA to the numerical features (Parch, Age, Fare, SibSp) to reduce the dimensions while retaining the variance as much as possible (95% variance retained as shown below)

A screen shot of a computer code

Description automatically generated



We then save the updated data in an updated\_train.csv