**THE TITANIC SURVIVAL SURVEY**

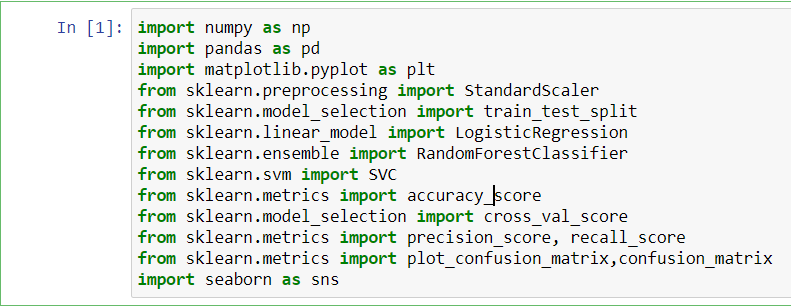
**Submitted by: Drishti Agarwal**

**DESCRIPTION**

*“The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from Southampton to New York City. There were an estimated 2,224 passengers and crew aboard the ship, and more than 1,500 died, making it one of the deadliest commercial peacetime maritime disasters in modern history."* The dataset that is analyzed holds the details of 899 passengers on board who either survived the disaster or unfortunately died, along with their details. Here we will first study the raw data and convert it into much more usable form through data wrangling, then we will scale the values and encode the character type values of some columns for data visualization and model selection/application.

**APPROACH FOLLOWED**

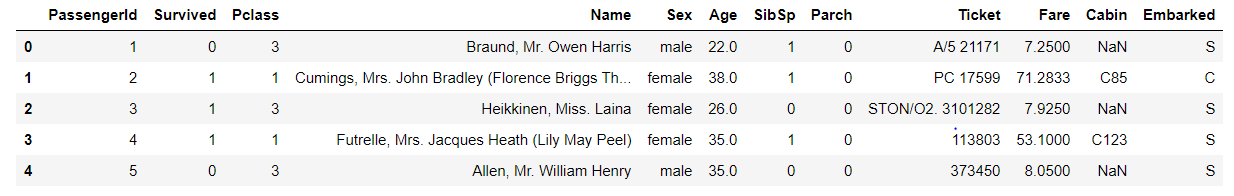
1. **Importing the libraries required:** the following libraries were used to build this project.



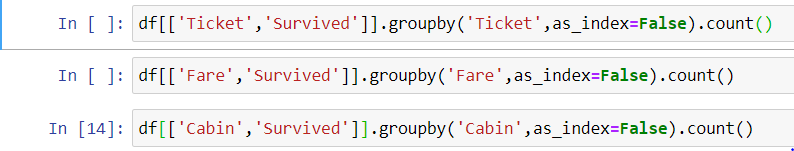
1. **Loading the Dataset:** the train.csv file was loaded to program using pandas function

read\_csv().

1. **Data Wrangling:** Before we can dive deeply, we must better understand what is in your data, which will inform how you want to analyze it.

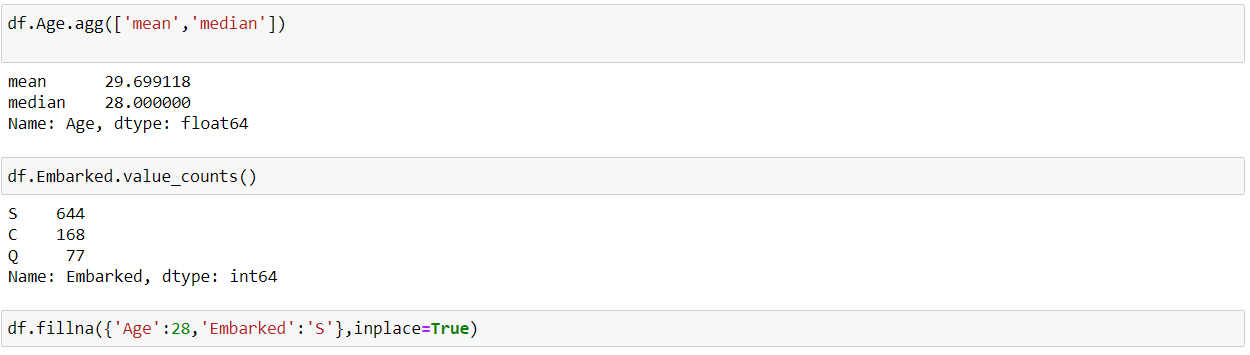


Here, columns like PassengerId and name have unique values and hence they will not contribute in classifying the survival of passengers. Also, we can observe that columns like ticket and cabin have alphanumeric value and should be checked by groupby function to determine groping in terms of survived column as follows:



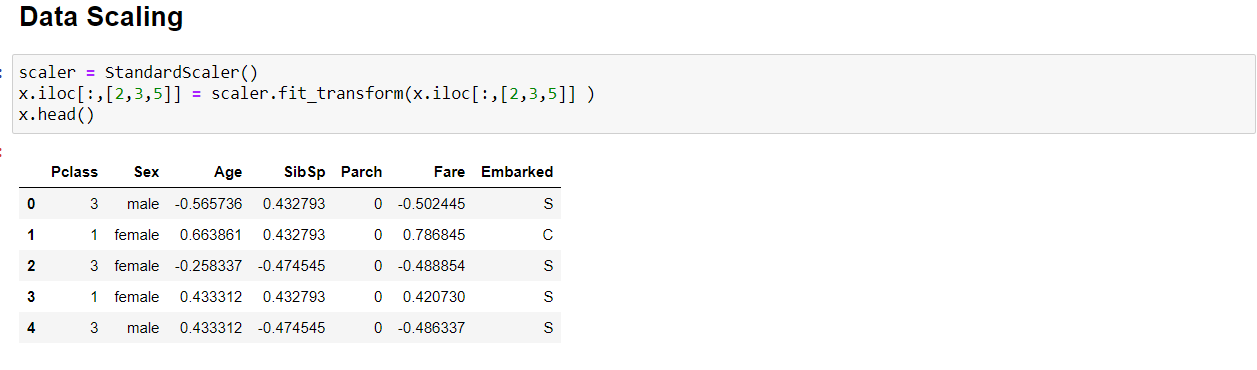
Thus, using this function on every column, we can specifically determine columns having a set of repeated values to be encoded. After doing this we observed that columns like ticket and cabin have way too many unique values that are alphanumeric and pose no significance over the survived columns value in terms of mean survival value. Also, age and fare have multiple numeric value that needs to be scaled. Furthermore, we can observe that parch, pclass, sex and embarked can be encoded into different columns for all grouped categories.

Next step is to determine the null values in the dataset and replace them with fillna as follows:

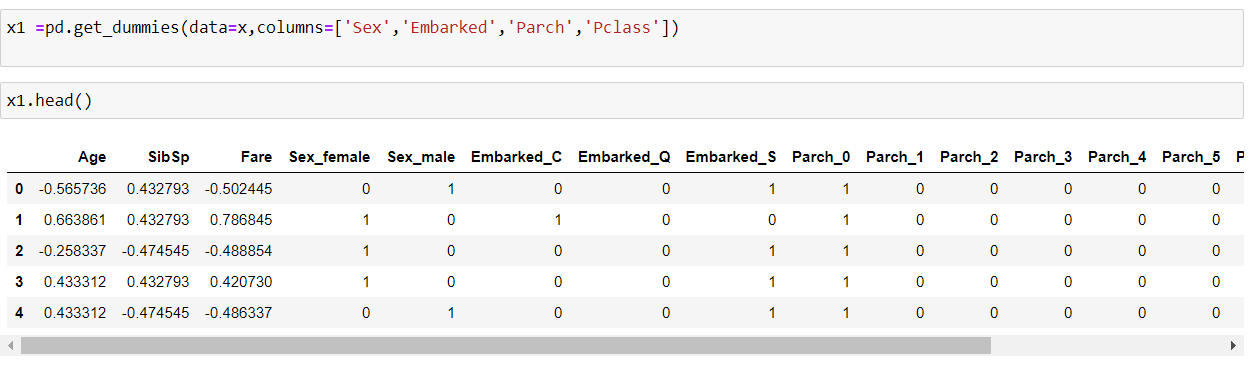


Finally, for easy handling of data we will divide it into x and y where x holds the features and y will hold the output column Survived.

1. **Data Scaling: The columns to be scaled can be done as follows:**



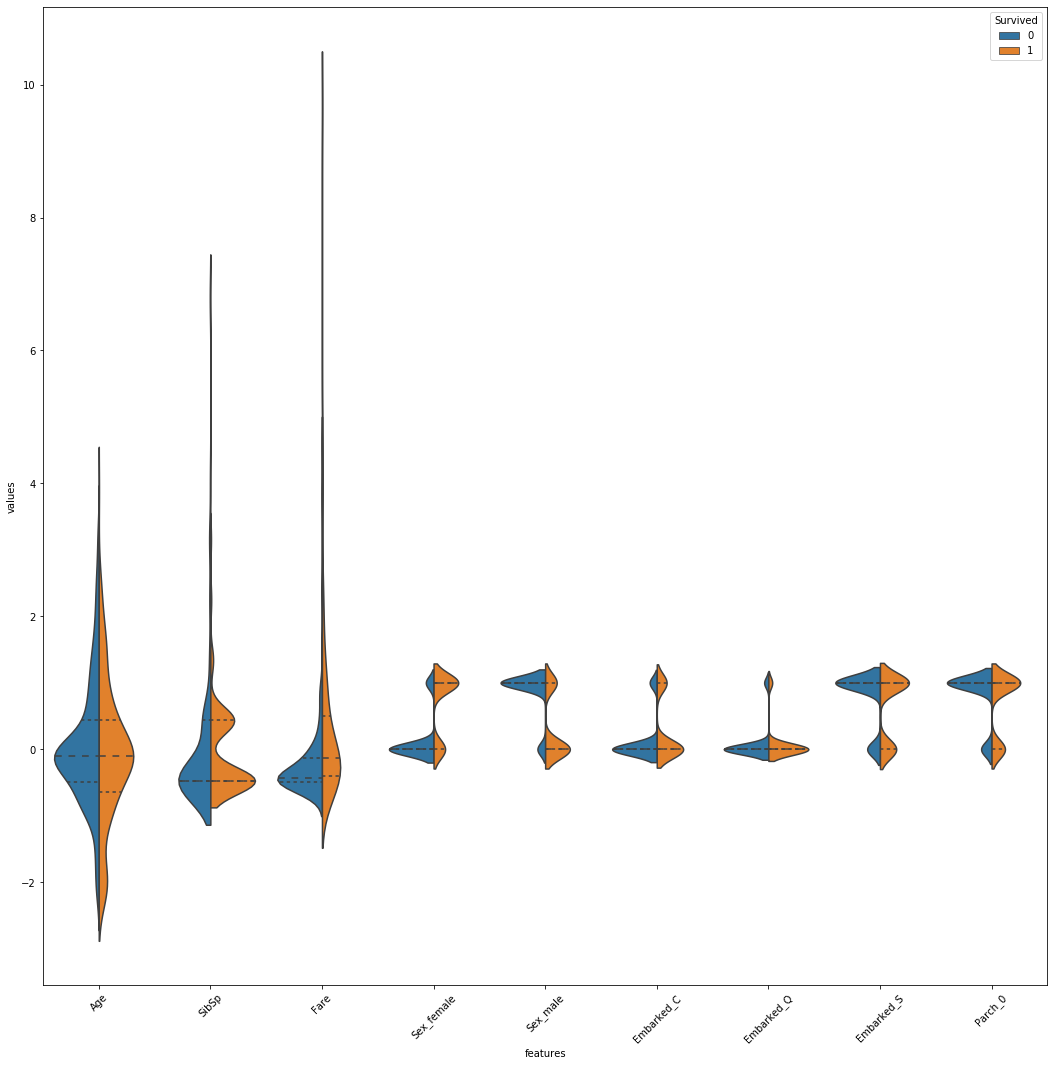
1. **Data encoding: The columns to be encoded can be done as follows:**

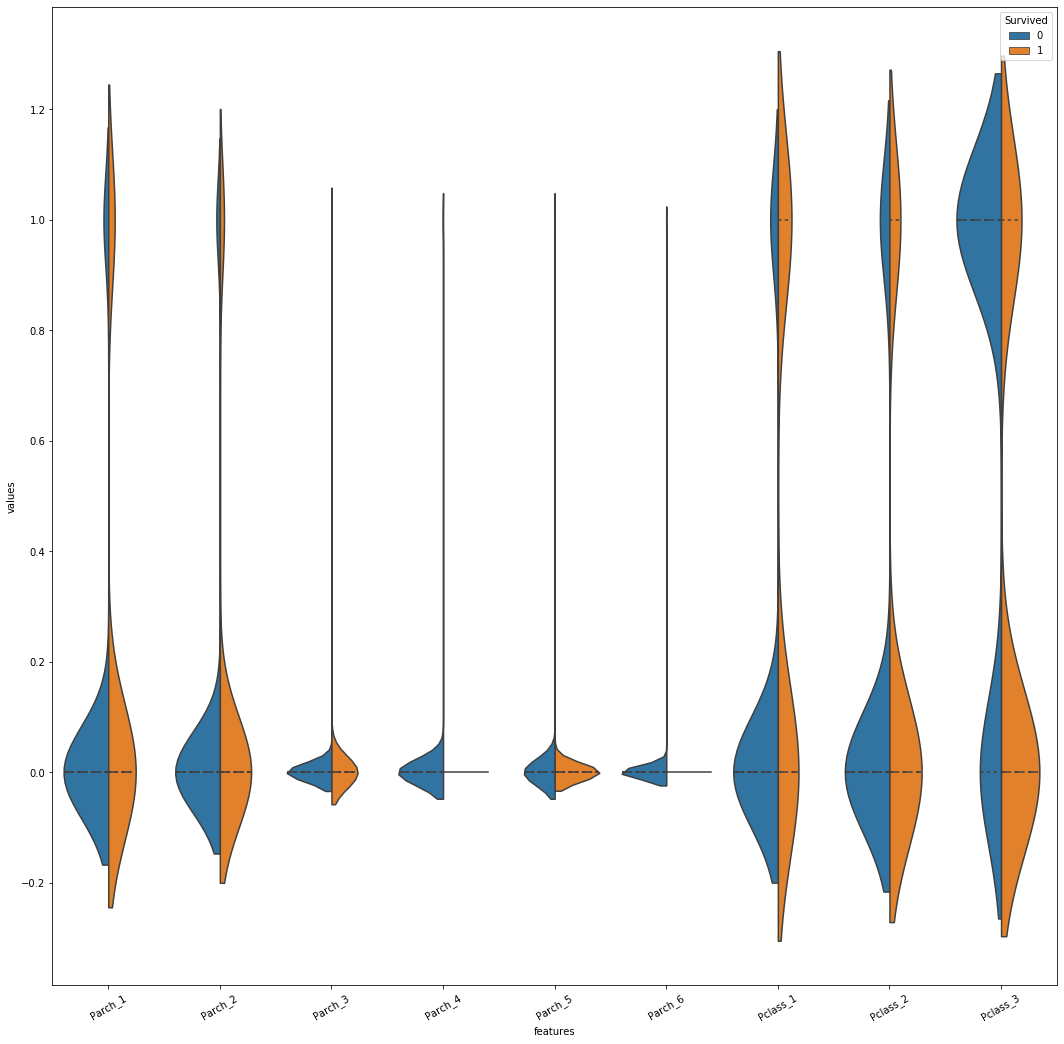


1. **Data Visualization:** This is done with the help of seaborn library that creates interactive graphs which helps us in determining the suitable features to be analysed by the model for final prediction. First, we will create a countplot to determine class imbalance.



Then to analyze the features we created violin plots, A Violin Plot is used to visualise the distribution of the data and its density. This chart is a combination of a [Box Plot](https://datavizcatalogue.com/methods/box_plot.html) and a [Density Plot](https://datavizcatalogue.com/methods/density_plot.html) that is rotated and placed on each side, to show the [distribution shape](https://en.wikipedia.org/wiki/Shape_of_the_distribution) of the data

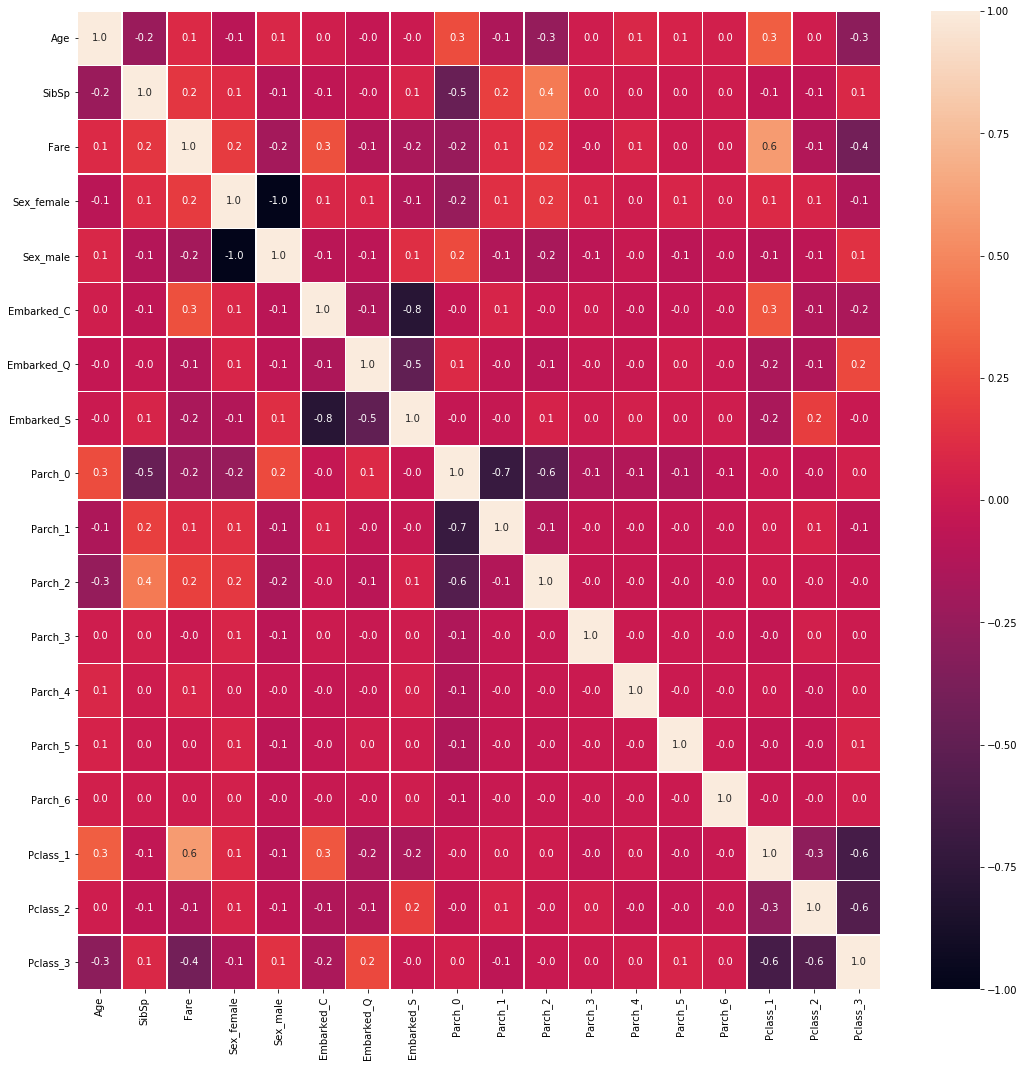




These plots indicate that out of all columns the embarked\_Q has equal probability density distribution for survived and not survived and it can be dropped.

Further we can create a heatmap to determine the correlation between the features.

In the figure below we can see that the diagonals columns are 1.0 as the columns will be correlated to themselves on x and y and no other cell having this value indicate no correlation between two different features.



1. **Model Selection:** first, we will split the model into train and test sets.



Its, feasible to try multiple algorithms and select the best one. Here, I tried to fit the model for Logistic Regression, Random forest and SVM and observed SVM classifier to give the best accuracy and cross value score after hyperparameter tuning.



**RESULTS:**

The model achieved the highest accuracy of approximately 84% and the cross-value score is 82%. Here we opted for an SVM classifier model as we observed quite a lot of outlier values that can efficiently be handled by this model.

