**Problem definition**

My job is to predict who is going to survive The Titanic Disaster and who is not based on some features that are known about the passengers. I have received two datasets: One dataset titled train.csv and the other is titled test.csv. Train.csv contains the details of a subset of the passengers on board (891) reveals the “ground truth” - did the passenger survive the accident. The test.csv dataset contains similar information but does not disclose the “ground truth” for each passenger. It’s my job to predict the outcomes. My job is to predict whether the other 418 passengers on board (found in test.csv) survived using the patterns found in the train.csv data, if there are any.

**Data collection**

I have tried to combine the two datasets train.csv and test.csv to work with them in R, however, I have more experience with Python and its libraries used for data analysis and because of that I decided to use pythonic libraries to read the datasets into their respective data frames. The data is provided to me in a .csv format which makes my work easier as I do not have to work with different data formats. On top of that, the datasets are relatively small in volume and thanks to that I do not need to clean the data before preparing it for analysis.

**Exploratory Data Analysis**

I started my analysis in Jupyter Notebook by looking at the train and test csv datasets and comparing their variables. After I gained a better understanding of these datasets, I can conclude that the train dataset has 891 rows and 12 columns. It has 177 missing values at the age column, 687 missing values at the Cabin variable, and 2 missing values at the Embarked variable. After, I plotted a bar chart with the Sex variable using count which showed that there are 577 male and 314 female passengers. There are 491 passengers traveling as class 3, 184 traveling in class 2, and 216 traveling in class 1.  Plotting some more charts with other variables showed me some more details about the train dataset. One of the interesting findings during this part of my analysis was that female passengers have approximately 80% probability of survival while their male counterparts survived only 20% of the time. Another insightful finding was that higher price implies higher survival rate - it seems that people who had more expensive tickets were more likely to survive in my train dataset.

**# Feature Engineering**

After I had a decent understanding of the dataset using visualizations and simple computations I moved onto the feature engineering part of my project. The first thing was to delete the columns that provided little to no value for my work. I decided to drop the columns Ticket, Cabin, PassengerId, and Died (was created to better visualize the relationship within sex and survival rate). Next, I split the Name column and extracted the titles of different passengers. This proved to be really useful and relatively simple to do. I split the column Name, found out that there are 17 different titles for representing various social status of the passengers on-board. After that I created a dictionary which could be improved upon as I am no expert in social titles from the 19.th century. I have concluded that there were 6 "social classes": 1. Officer, 2. Royalty, 3. Miss, 4. Mrs., 5. Mr., 6. Master and I have matched the previous titles passengers provided (17) to these new titles (6). After I had the Title column ready, I dropped the name column which provided little value (most likely). After this I decided to convert Sex, Embarked, and Title columns into numerical with thinking that it is going to make my logistic regression model more accurate. My first analysis showed me that there are missing values in age column. I have decided to compute a median a median for both, male and female age variables and then impute these values to the respective missing values of my train dataset. I also saw that there are 2 rows with missing Embarked values but because they were just two I decided to drop the whole row. There might be some space for improvement. After these adjustments were made, there are no missing values in any of my variables. Another technique to improve my models accuracy is to normalize or rescale the Age and Fare columns for the values to fall between 0 and 1. This was done using a formula which df1.Age = (df1.Age-min(df1.Age))/(max(df1.Age)-min(df1.Age)). After all these steps were done, I had a clean dataset with only numerical values, and I was ready to fit an ML model to it.

**# Model Selection**

I am trying to put the dependent variable into 1 of two classes - true or false (Survived == 1 || 0). This is a problem to try solving with Logistic Regression - binary classification.

I have tried different algorithms such as decision tree, or random forest.

**## Decision Tree**

I have tried to use decision tree for the same problem and datasets. My score was 0.77511 which is lower than the one I received using logistic regression. It might have been because the dataset I used only contained numerical values or simply because this algorithm did not perform as well on the given problem. I have not yet tried to change the dataset to be better working on decision-tree algorithm.

**## Random Forest**

When implementing Random Forest, I did a couple changes to my dataset. These changes included

I spent the most time on random forest implementation and I think that the data preparation was really thorough and helped me achieve a score of

**## Logistic Regression**

My highest score using logistic regression was 0.78708. It is classification problem and also there are only two classes which makes is suitable to try logistic regression (binary classification).

**# Model Training**

**# Model Evaluation**

**## The accuracy of The Logistic Regression model was 0.8371 and confusion matrix provided the following results: [[96, 14], [15, 53]].**

- 98 - True Positives: The number of instances correctly classified as positive (Survived = 1).

- 12 - False Positives: The number of instances that were incorrectly classified as positive (Survived = 1) when in fact they were negative (Survived = 0).

- 16 - False Negatives: The number of instances that were incorrectly classified as negative (Survived = 0) when in fact they were positive (Survived = 1).

- 52 - True Negatives: The number of instances correctly classified as negative (Survived = 0).

**## The accuracy of The Decision Tree model was 0.8483 and confusion matrix provided the following results: [[98, 12], [15, 53]].**

**## The accuracy of The Random Forest model was  and confusion matrix provided the following results: .**

**# Preparing the test dataset**

**# Results Interpretation**

My best score was achieved using Logistic Regression and my prediction was accurate with an official score 0.78708 which ranks me at 1053 out of 13953 players. In this paper, I will describe my how I achieved this score and what I might try to improve my score further. The score of 78.7% means that I am 78.7% confident that my model can predict the future of who is going to survive and who is not 78.7% of the time. My accuracy score from sklearn.metrics was 84.27%.

**# Space for improvement**