Table of Contents

Mini Project 1. Titanic-Machine Learning from Disaster1

Introduction and Objective3

Model Selection ……………………………………………………………………………………………………………………………3

Preprocessing the data……………………………………………………………………………………………………………….…3

Handling Missing values4

Feature Extraction5

Handling categorical features6

Converting to numerical values …………………………………………………………………………………………. 7

Standardization8

Important Observations8

Predictive Models……………………………………………………………………………………………………………………….10

Logistic Regression …………………………………………………………………………………………………………….10

Naïve Bayes ..………………………………………………………………………………………………………………………10

KNN ………....……………………………………………………………………………………………………………………….10

SVM ……………………………………………………………………………………………………………………………………11

Random Forest …………………………………………………………………………………………………………………..11

Bagging ………………………………………………………………………………………………………………………………11

Neural Network …………………………………………………………………..……………………………………………..11

Decision Tree ……………………………………………………………………………………………………………………..11

Further Tested versions of Pre-processing ………………………………………………………………………………….11

Handling missing data ……………………………………………………………………………………………………….11

Extracting new features …………………………………………………………………………………………………….12

Converting categorical features to numerical values………………………………………………………….14

Standardization …………………………………………………………………………………………………………………14

Saving to csv …………………………………………………………………………………………………………………….14

Conclusion……………………………………………………………………………………………………………………………………15

References…………………………………………………………………………………………………………………………………..15

**MINI PROJECT 1. TITANIC-MACHINE LEARNING FROM DISASTER**

* 1. **Introduction and Objective:**

The Titanic was a British [passenger liner](https://en.wikipedia.org/wiki/Superliner_(passenger_ship)) that [sank in the North Atlantic Ocean](https://en.wikipedia.org/wiki/Sinking_of_the_RMS_Titanic) in 1912 after striking an [iceberg](https://en.wikipedia.org/wiki/Iceberg) during her [maiden voyage](https://en.wikipedia.org/wiki/Maiden_voyage) from [Southampton](https://en.wikipedia.org/wiki/Southampton) to [New York City](https://en.wikipedia.org/wiki/New_York_City). Of the [estimated 2,224 passengers and crew](https://en.wikipedia.org/wiki/Sinking_of_the_RMS_Titanic#Casualties_and_survivors) aboard, more than 1,502 died, making the sinking one of modern history's deadliest peacetime commercial [marine disasters](https://en.wikipedia.org/wiki/List_of_maritime_disasters_in_the_20th_century#Peacetime). Titanic has inspired countless books, articles and films, and her story has entered the public consciousness as a cautionary tale about the perils of human hubris.

The objective is to use machine learning algorithms to estimate which passengers survived the Titanic disaster. This project is part of a Kaggle competition, the main goal is focused on answering the question, what type of people were more likely to survive this accident.

* 1. **Model Selection:**

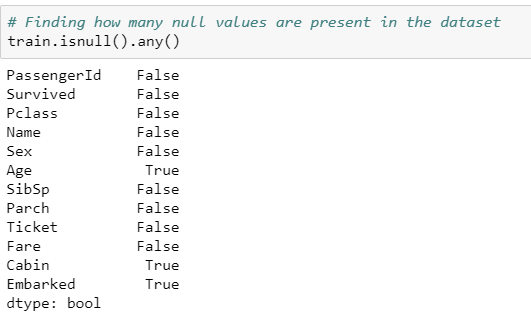
For the purpose of this project we have selected the following models to predict the who survived from the accident. Each of these models are discussed more in length, in the later parts of this report.

1. Logistic Regression
2. K -NN
3. Naïve Bayes
4. Neural Networks
5. Decision Tree
6. Bagging Classifiers
7. Random Forest
8. SVM
   1. **Preprocessing the data:**

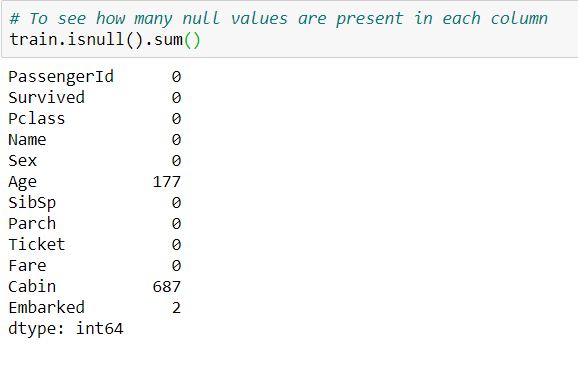
Pre-processing the data is one of the most critical and the first steps before we can proceed creating our model. When dealing with real world data, it is very likely that the data can be inconsistent, incomplete, lacking in certain behaviors and it may contain many errors. It is almost impossible to get a desired accuracy without cleaning the data and proceeding towards modelling a predictive model. We typically utilized two separate files viz., train.csv and test.csv available on the Kaggle to train and test our models respectively.

In order to pre-process our data, the following steps were employed:

1. From the train.csv file, we first identified that there were NULL values in 3 features of the data. We used the following code to identify the NULL values in train.csv

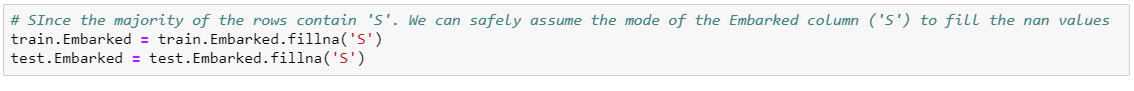


1. Checking the number of null values in each feature.

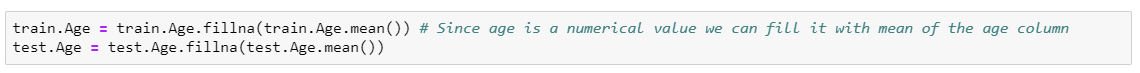


**Handling missing data:**

1. With ‘Embarked’ having the least number of NULL values, we began processing our data with this feature. The train.csv typically contains 3 values for the ‘Embarked’ feature viz., ‘S’ , ‘C’ and ‘Q’. Since both the train.csv and test.csv files contain majority of their values as ‘S’ we replaced the two rows of NULL values for ‘Embarked’ with ‘S’.



1. We next moved on to cleaning the data for the ‘Age’ feature. Since ‘Age’ has a numerical value, we replaced NULL values in the ‘Age’ column with the mean of the age.



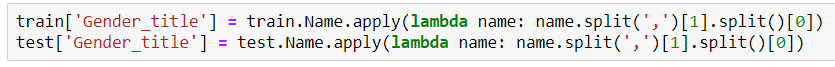
1. The third feature ‘Cabin’ contains the most number of NULL values in it. We extracted the 1st letter of it to know the cabin each person belongs to. If a person doesn’t have any cabin then it will saved as ‘N’ (since Nan). There is a sample in Cabin\_name with ‘T’. But the remaining Cabin names are in alphabetical order which end at ‘G ‘. So ‘T’ might be incorrectly entered. So I replaced it with ‘A’

**Feature Extraction:**

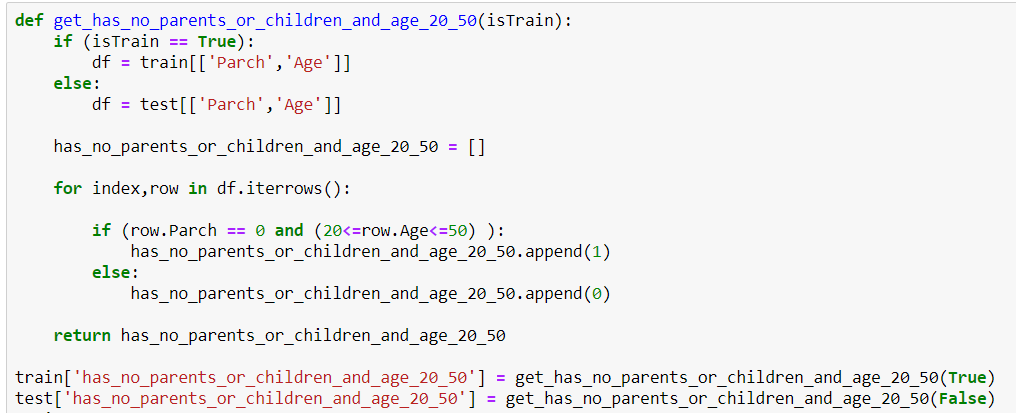
1. Since Fare decides where a person is placed in the ship which is dependent on the survival rate of that person. We used Fare categories to bin different people into different Fare categories.



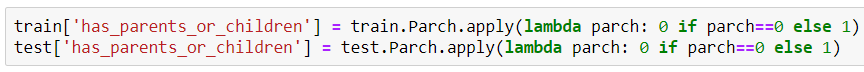
1. We extracted Gender\_title from the person’s name so as to use it as a feature. Since gender of a person is highly correlated with the survival of that person. We can use Gender\_title as an estimator if a person had chances for survival or not.



1. Since Parch is dependent on Age, and since both are correlated with the survival rate of person. We created Age categories to see how the correlation exists. We observed that a person with no parents or children and having an age between 20 to 50 had high survival chances. So we extracted this feature named **‘has\_no\_parents\_or\_children\_and\_age\_20\_50**’.

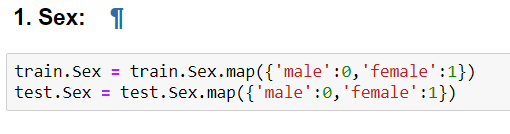


1. Since having no parents or children is also a good indicator for survival rate, which can be seen from the visualizations in the preprocessing.ipynb file, we extracted another feature named ‘**has\_parents\_or\_children’** which had 14.7% correlation with the survival rate.

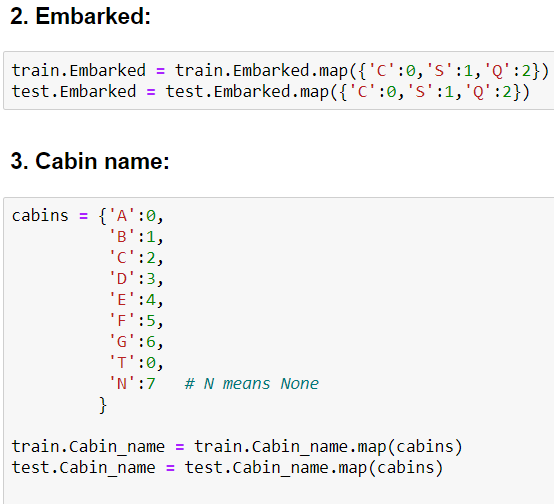


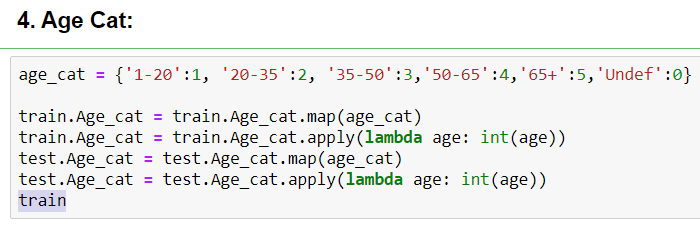
**Handling Categorical features:**

1. Since Sex can be Male or Female. We mapped it to 0 and 1 respectively.



1. Since the number of features would increase with the number of unique values in the column when creating dummies and also because we cant predict if there are any other unique values which may show up in test data. We mapped all the values in columns ‘Embarked‘ & ‘ Cabin\_name‘,’FareCat’,’Gender\_title’ and ‘Age\_cat’ to discrete numerical values for the model to be able to fit.

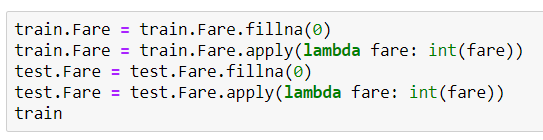






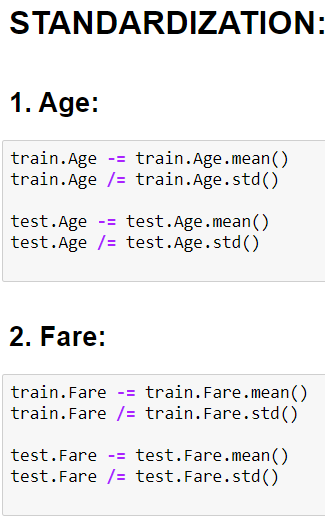
**Converting to numerical values:**

1. Since the Fare is a continuous value and for the model to fit, data should be discrete. So we converted Fare to discrete values.



**Standardization:**

1. Since the values of Age and Fare are on a different scale than the other columns. We standardized ‘Age’ and ‘Fare’ columns.



**Important Observations:**

1. 38% people survived in 891 samples, and no null values.
2. 50% - max% Quartiles are with samples of Pclass-3, and no null values.
3. No frequent names, and no null values.

4. More samples are Male (577/891)\*100 = 64.75% are male

5. There are 714 no null values in Age. Mean of age is 30. 75% people are below 38 age.

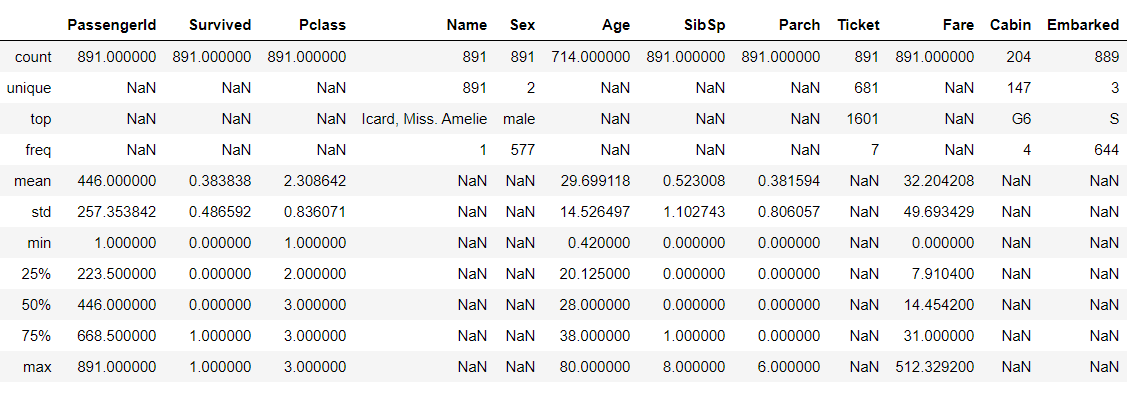
6. SibSp doesn’t have any null values. Mean for SibSp is 0.52 indicating most people didn’t have any Siblings and spouse. (Since its close to 0) Almost 75% people were not having any siblings or spouse.

7. Parch doesn’t have any null values. Mean for feature Parch is 0.38 indicating most people didn’t have any Siblings and spouse. (Since its close to 0) Almost 99.9% people were not having any parents or children.

8. 691 unique ticket features inform that tickets had duplicates.

9. Fare doesn’t have any null values and 75% people paid less than 31 as Fare.

10. Embarked has 'S' as the highest frequent value. and only 2 values are missing in Embarked.



* 1. **Predictive Models:**
     1. **Logistic Regression:**

Logistic Regression is used when target variable is a binary data. We found it to be suitable and used the same. All the independent features explain the relationship with the target variable. The accuracy is 76.076% when we used logistic regression.

* + 1. **Naïve Bayes:**

This method is used when we want to predict using multiple classes. The independent feature assumption of the Naïve Bayes makes it solvable when compared to other models. This model is suitable for high dimensional datasets. Our model shows us an accuracy of 74.162% for Naive Bayes classifier.

* + 1. **KNN:**

KNN is a classification algorithm for classifying based on k nearest features. We specify the number of neighbors we use in our algorithm. Based on our understanding of the dataset, we can try for different neighbors and obtain the accuracy. The accuracy for KNN for our dataset is 76.076%

* + 1. **SVM:**

SVM can be used for both regression and classifier. SVM tries to find the highest margin between the classes. SVM can be used for both linear and non-linear classification. Correlation between the independent features can reduce accuracy. After reducing the correlation, we achieved an accuracy of 73.684% for this model.

* + 1. **Random Forest:**

This algorithm on a random basis selects random samples and get the prediction based on the samples. We give the no of estimators we need to predict the model. The accuracy using random forest is 77.99%

* + 1. **Bagging:**

Bagging constructs n trees, uses those prediction results to produce a final prediction result. It reduces the variance and solves the problem of over-fitting. The accuracy of using bagging model is 77.033%

* + 1. **Neural Network:**

Neural network detects the patterns in the dataset. Hidden layers combine many input layers and produces an output layer. The accuracy of using neural network is 76.555%

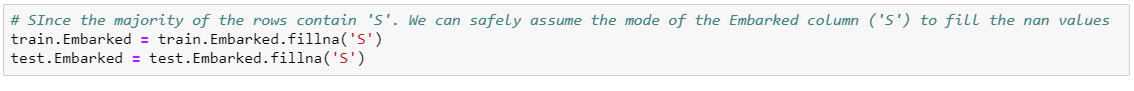
* + 1. **Decision Tree:**

The accuracy of using Decision Tree is 74.641%

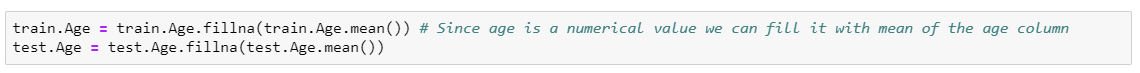
**1.5 Further tested versions of preprocessing:**

**Handling missing data:**

1. With ‘Embarked’ having the least number of NULL values, we began processing our data with this feature. The train.csv typically contains 3 values for the ‘Embarked’ feature viz., ‘S’ , ‘C’ and ‘Q’. Since both the train.csv file contains majority of their values as ‘S’ we replaced the rows of NULL values for ‘Embarked’ with ‘S’ in both train and test.

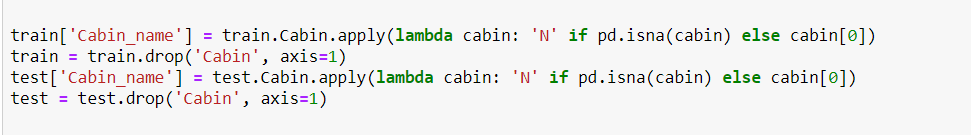


1. We next moved on to cleaning the data for the ‘Age’ feature. Since ‘Age’ has a numerical value, we replaced NULL values in the ‘Age’ column with the median of the age of the ‘**Gender\_title’**.(We will discuss about this new feature below in feature extraction phase.)

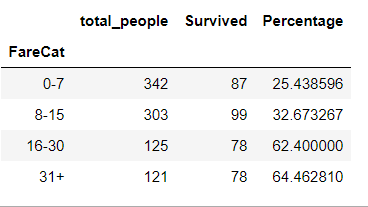


**Extracting new features:**

1. The third feature ‘Cabin’ contains the most number of NULL values in it. Instead of entirely removing the Cabin feature. We are extracting the first letter of the Cabin and named it ‘**Cabin\_name’** .



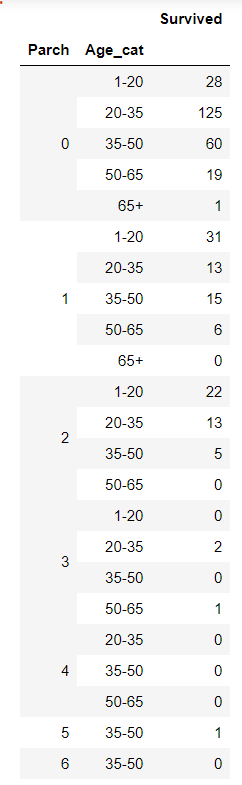
1. We tried to get the Fare Categories from the Fare because those were correlated with survival. Higher Fare people had higher survival rate.



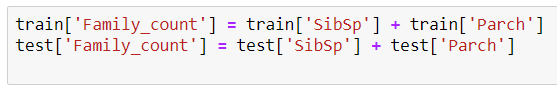
1. From the name, I have extracted people’s titles so as to estimate age. (Since I could fill the nan values of Age since Gender\_title are highly correlated with Age.) I used median of all Gender\_title age’s to fillna of ages. One more reason to use Gender\_title is that there is some correlation with survival rate. Below is the median age’s for each gender\_title. I used this to fill up the null values in the age. (Refers back to the Handling Missing Data part of the report)



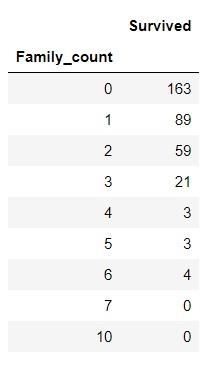
1. Parch when observed against the survival rate, we observed that people with Parch 0 had a higher survival rate. We extracted this as a feature named **‘has\_parents\_or\_children ‘**.



1. Parch & Age together were able to predict the survival rate of a person. Since there were 161 different ages. We created different age categories from 1-20, 20-35, 35-50, 50-65, 65+. Observing this data we can see that Parch with 0 and Age\_cat between 20-50 had the highest survival rate. This was used as a new feature with name ‘**has\_no\_parents\_or\_children\_and\_age\_20\_50’**.
2. Since Parch and SibSp had a very similar distribution we combine them into 1 feature named ‘**Family\_count**’.



1. Based on this new feature(Family\_count) we can see that the number of survived people for family count 0 is the highest. So, we can extract a new feature out of this named ‘Alone’.



1. One more feature that was extracted from Family\_count was ‘**No\_family\_and\_age\_20\_50’** because the people with ages in between 20 – 50 and family count 0 had the highest number of survivors.

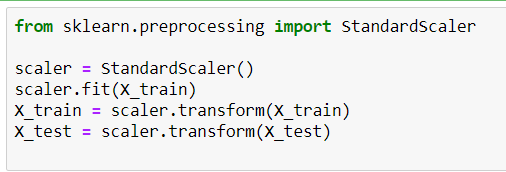
**Converting categorical features to numerical values:**

1. We converted the Sex feature to 0 for male and 1 for female.



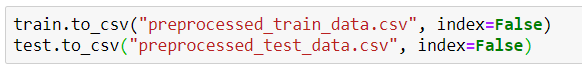
**Standardization:**

1. We standardized the entire data using StandardScaler() after preprocessing the data.



**Saving to csv:**

1. The preprocessed train and test files have been saved to “preprocessed\_train.csv” and “preprocessed\_test.csv” respectively.



**1.6 Conclusion:**

Random Forest produces the highest accuracy on Kaggle. Further more, it prevents overfitting the data by using only a limited set of samples for each estimator. The feature\_importances\_ in Random Forest was useful to know what features were useful and what features were not adding any new information to the existing model. The next best classifier was the Bagging classifier. Logistic Regression was the simplest model which could almost give the same amount of accuracy as the Random Forest & Bagging classifier. The remaining classifiers had accuracies around 75%. These accuracies may change and not be accurate since we are using GridSearch to tune the hyper parameters.

**1.7 References:**

* https://en.wikipedia.org/wiki/RMS\_Titanic
* <https://www.history.com/topics/early-20th-century-us/titanic>
* <https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html>
* <https://towardsdatascience.com/why-feature-correlation-matters-a-lot-847e8ba439c4>
* <https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa>
* All the materials provided in this course.