**TITANIC PROJECT**

1. Gathering Data
2. Data Pre-processing (Cleaning and Preparing Data)
   * Cleaning of data e.g., conversion of data, missing value imputation
   * Train/Test split
3. Choosing and training a model
4. Evaluating the model
5. Prediction

1. Gathering Data

I start by importing important libraries. Such as Pandas and Numpy are data manipulation libraries. For machine learning I will use classification algorithm Random Forest or Logistic Regression.

I use train\_test\_split function to split the data into train/ test to check and avoid overfitting.

I will now read the csv file in Pandas.

train.head() will show the first 5 rows of the data. There are few NaN values in the data which I have to impute.

### **2. Data Pre-processing**

Describe is a good command to get to know the data in a summarized way.

This is a case of supervised learning in which the model needs inputs and output to learn. Ill in this case ‘Survived’ Column is output column and rest all are input columns.

So there are 11 input columns. HoIver, not all columns are always important for the model to learn. If you remember the Titanic movie, you will know that the rich Ire more likely to survive. Also, the preference was given to children, women and aged persons.

So according to our hypothesis, older rich women and children Ire the most likely to survive and poor middle-aged men Ire the least likely to survive.

Age and Sex are directly provided in the data. I can presume whether a person is rich or poor by looking at Passenger class (Pclass).

So these are the 3 inputs to our machine learning algorithm: Passenger class, age and sex. I can see that Age has 177 missing values out of 891. Thus I can do the missing values imputation. For now, let’s not take the Age column. So for model input, I will have only Passenger class and Sex*.* The output is the Survived field.

Let’s extract selected 2 input columns into a new dataframe train\_x.

Similarly, extract the output to train\_y.

I split our data into a train set and a cross-validation set. The training set is used to train the machine learning algorithm. While the cross-validation set is used to find the model accuracy (as I have the actual output for the cross-validation set). Accuracy is calculated by comparing the actual output with the predicted output.

I will use the train\_test\_split function to create the test/ train (cross-validation) split. I will use 70% of the data to train and model and 30% of the data to check accuracy.

### **3. Choosing and training a model**

### This simple fit() function is used to train our algorithm. This function takes our input dataframe (tr\_x) and learns the expected output (tr\_y). That’s why we narrowed the input columns so that the algorithm is not confused by the noise.

### **4. Evaluating the model**

The instance has now “learned” how to predict Titanic survivors as the model is fitted. Now we can check how accurate our algorithm is on cross-validation data.

The score () function takes the cross-validation input and finds out the accuracy by comparing our predictive output and the known test outputs.

I also applied logistics regression model.

The accuracy for random forest model is 78.73134328358209% and the accuracy for the logistics regression model is 79.1044776119403%. So , I will choose the logistics regression model.