**Straightforward - The Titanic (Machine Learning from Disaster)**

**Step 1:** Exploratory study

* Already completed in previous submission

**Step** **2:** Investigate learning on Kaggle

1. **Participant**: Quick Titanic Model in Python – 5 steps

* **Competition public score:** 0.75358
* **Data Wrangling**: For data cleansing, this participant use the built-in function(isna(), sum(), etc ) to explore the number of missing values in both datasets( train and test ).
* It uses the heatmap() to show correlation between pairs of features. It also shows the impact for survival as lower impact and higher impact. Based on the survival rate, it can be classified the most negative and positive impact on survival.
* Based on the heatmap, we can describe the necessary and correlated features like Pclass, Age, Sex, Fare and trained the data.
* One can converge faster when the data is normalized, this is a key step in every ML situation. To do so I will use MinMaxScalar library from Scikit Learn, but first we need to drop some label. When scaling we only fit the scaler to the training dataset.
* **Data Analysis**: For the given trained data, we can visualize using python library matplotlib. We can use bar plot to show survived and dead rate for *sex, Pclass* feature.
* Drop unnecessary data those which seems to have the lowest correlation with survived feature.
* Converts categorical variables into dummy variables and fill the missing value with median value.
* **Model Training**: In this participant’s submission, they used random forest technique to evaluate the model. It selects random samples from a given dataset, then construct a decision tree for each sample and get a prediction results from each decision tree. Validate each predicted result and select the final prediction.
* I will choose a final model with best configuration which is the smallest mean absolute error.

1. **Participant:** LogReg + KNN

* **Competition public score:** 0.81100
* **Data Wrangling**: This participant uses the NumPy, pandas, matplotlib and seaborn library to work with Titanic datasets. For data preprocessing, it counts the total survived passengers with male and female. It also shows the histogram of total count with different age group, Pclass, Embarked and found the mean, median of both (men and female).
* **Data Analysis:** From the relation between survived and gender, I think men were survived more compared to women. The Reason behind this is workers on Titanic were men as death ration is more than women. So, women survived more than men. From the relation between survived and age distribution, I think older people died and younger people survived more. The people on the 3rd class are less likely to survive as there were more people dead.
* **Model Training:** Participant uses the two models (KNN and Logistic Regression) to predict the survival. It found the accuracy without feature scaling with logistic and knn. And with scaling StandardScaler with logistic and knn. There is one heatmap correlation matix, Age, sex, NameLength, Deck and Embarked feature takes into consideration. I think feature Cabin would also interpret some what as it has deck number. Based on the deck number, it can be classified the how many people survived from different decks. From the participant’s submission, we can see that KNN is better and stable to predict the survival on the titanic.

1. **Participant**: XGBoost – Titanic | Top 2% / Accuracy 98.8%

* **Competition public score**: 0.98803
* **Data Wrangling**: It uses the python library Numpy, pandas matplotlib and seaborn. For data preprocessing, it uses the MinMaxScaler, StandardScaler, Normalizer and PowerTransformer. It considered all the features to predict survival.
* **Data Analysis**: One of the different variables from the above two participants is Sibsp and ticket number. Considered Embarked and Fare as train and test, respectively and combine train and test sets for preprocessing.
* **Model Training**: It uses the xbgoost.XGBClassifier and DictVectorizer to check its accuracy with 10-fold cross-validation. Although, it is not great solution as it lose the ability to distinguish features that are missing and features that have zero value. It compares the AdaBoost, GradientBoost, RandomForest, ExtraTreeClassifier. From the given techniques gradient gives the good results.

**Step 3:**

* I will see some attributes that are irrelevant, such as PassengerId, Name , Ticket and Cabin. I will remove these attributes from the test and training sets. Remove null and zero value data in data preprocessing such as, isnull(), sum(), fillna(), sort\_value(),etc.
* For better processing, I will convert the data to the numpy(to\_numpy() ) format so that I can perform the classification. The can be normalized, in order to avoid the effect of the scale (StandardScale() ) of the attributes in the learning methods.
* After preprocessing, I will visualize some key point that needs to be considered to create model. From the given 3 participant’s work, I can learn that I should identify the attributes those data that are most explain the variance in the data. For example, the class in which the passenger traveled, the age and number of the siblings and spouses on board are the attributes that most explain the variability of the data.
* To create the model to predict the survival, I will use Classification method and Random forest. I classification, I will consider the k-nearest neighbors. Using cross validation, we can find the best k value. After finding the best value of k, we can do the classification using the k-neighbors method on the other data. For random forest, the parameters of the estimator used to apply these methods are optimized by cross-validated over parameter setting. One think about the Random Forest Classifier from Sklearn is that provides a very easy way of reviewing the important features for the classification. I can remove the less important features using the algorithm, and see if the accuracy improves.