## Analysis Report:

The Titanic machine learning from disaster dataset comprises of the training and the test data. With initially 11 features and one target in the training data set, and 11 features in the test data set. The test data set does not contain the target (‘Survived’).

In the data analysis and exploration part initially, we find the missing values in the feature set of the training and the test dataset. In the data processing and wrangling part, we replace the missing values with necessary substitutes and also create new features to remove insignificant features from the data set. Some of them are mentioned below,

1. The SibSp and the Parch features are together combined to form a new feature called ‘Relatives’ and also a new feature which tells us if the passenger is travelling alone or is accompanied by a companion. This feature is named ‘hasCompanion’ and holds the value 0 or 1.
2. We see the ‘Cabin’ feature has 77% missing data. So we replace this feature by adding a new feature named ‘Deck’, which holds a numerical value with respect to the alphabet found in the ‘Cabin’ feature values.
3. For the ‘Age’ feature we see it varies between 0.4 to 80 and all the values are float. Initially, we replace the missing values in the ‘Age’ feature and then we group certain age groups together to replace the existing values in the feature.
4. The same approach is followed for the ‘Embarked’ feature. Once all this is done we convert the float values to integer types to maintain consistency across all the feature values and in turn it would be easier for the model training.

## Creating the Machine learning models:

Firstly, we create X\_train, Y\_train and the X\_test data set required to create our machine learning models.

### Logistic Regression:

* We train the Logistic Regression model using the training values. And fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **81.48**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.79425**.
* This is the best score that I have obtained so far with a rank of **3023** at the time of submission.
* The screenshot of the score can be seen in the iPython file.

### Stochastic Gradient Descent:

* We train the Logistic Regression model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **73.06**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.75598**.
* This was the third submission to the Kaggle site and the rank remained unchanged after the 2nd submission which was approximately ~16900
* The screenshot of the score can be seen in the iPython file.

### Gaussian Naïve Bayes:

* We train the Gaussian Naïve Bayes model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **77.78**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.72248**.
* This was the first submission made and the rank obtained at the time of submission was approximately ~18300
* The screenshot of the score can be seen in the iPython file.

### Random Forest Classifier:

* We train the Random Forest model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **92.82**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.76075**.
* After this submission, the rank remained unchanged at 3023 at the time of submission. So, this model did not yield a better score than logistic regression.
* The screenshot of the score can be seen in the iPython file.

### K Nearest Neighbour:

* We train the K Nearest Neighbour model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **85.63**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.70813**.
* Since this submission was done after the Logistic Regression, the rank remained unchanged at 3023. This is lowest score obtained so far.
* The screenshot of the score can be seen in the iPython file.

### Decision Tree Classifier:

* We train the Gaussian Naïve Bayes model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **92.82**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.72248**.
* This was the last submission made and the rank remained unchanged at 3023 since the score is not better than that we obtained from Logistic Regression.
* The screenshot of the score can be seen in the iPython file.

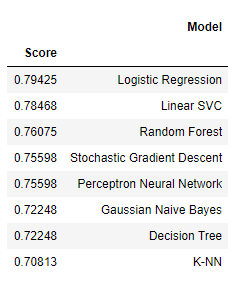
### SVC Classifier:

* We train the Gaussian Naïve Bayes model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **81.37**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.78468**.
* Since the score was not better than that of Logistic Regression the rank remained unchanged at 3023.
* The screenshot of the score can be seen in the iPython file.

### Perceptron Neural Network:

* We train the Gaussian Naïve Bayes model using the training values and fit them to the model which is further used to predict the outcome.
* We calculate the score of our model, in our case the score of the Logistic Regression model is **80.13**
* Based on the predicted values we create the output CSV file. This file is uploaded to Kaggle.com and the score obtained is **0.75598**.
* This being the second submission and better than that of Gaussian Naïve Bayes the rank obtained at the time of submission was approximately ~16900
* The screenshot of the score can be seen in the iPython file.

Finally, we obtain the table of scores with respect to the machine learning models as shown below,



From the above table we can see that the KNN predictions gave us the lowest score on Kaggle and the Logistic Regression predictions gave the best score among all.

==================================================================================

**END OF FILE**

==================================================================================