Sushanth Avvaru
RedID:- 819693674

Obtaining Datasets:-

Data have seen obtained from www.kaggle.com. We have test data in test.csv and training data in train.csv. The training data contains details of the passenger and if they survived or not. I used python notebook to do this project. The packages imported are pandas, numpy and random for data analysis. Seaborn for visualization. I combine both training and testing data at first to process them.

The following operations are done on the data to prepare the model.

- Classifying:- To divide data based on categories to analyze the solution.
- Correlating:- To see if there is relation between any feature vector and solution.
- Converting:- making name features to ordinal values in the same magnitude for efficient output.
- Completing:- Completing missing information on test data.
- Correcting:- Checking if any of the data are errors.
- Creating:- Combining features to see if the new feature turn out better.
- Charting: plotting all the obtained results.

Operations:-

Classify:- We analyze the survival rate bases on each category and tabulate them.

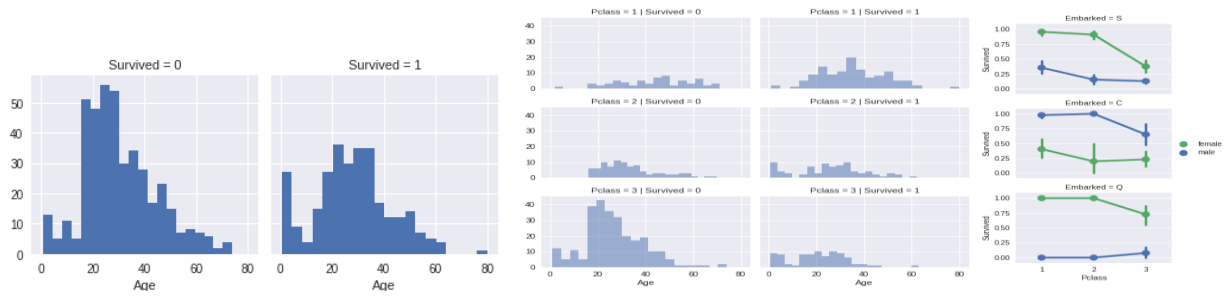
	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

	Sex	Survived
0	female	0.742038
1	male	0.188908

	SibSp	Survived
1	1	0.535885
2	2	0.464286
0	0	0.345395
3	3	0.250000
4	4	0.166667
5	5	0.000000
6	8	0.000000

	Parch	Survived
3	3	0.600000
1	1	0.550847
2	2	0.500000
0	0	0.343658
5	5	0.200000
4	4	0.000000
6	6	0.000000

Correlating and making assumptions:- We combine feature and plot histogram for visualization. We check which feature contributes more for survival rate.



Sushanth Avvaru
RedID:- 819693674

Converting numerical and alphabetical features into ordinals:- We convert Sex, Embarks and their name designations to ordinals for easier evaluation and grouping.

Completing Missing values:- We complete missing values in the test and train data. We combine pclass and sex feature find out what age people are more in each combination and then fill in the missing the missing values. Some Embarks are missing. We fill in the missing embarks by most frequent occurring embarks. Some fare values are missing. We complete them the missing values by median occurrence value. We fill in the missing values and then correlate again.

Combining features to create a new feature:- We combine Parch and SibSp feature to produce family feature. Then we perform correlation to see the survival rate based on family size. Now we drop Parch and SibSp feature as they are no longer required. We also create isAlone from family feature and Delete damily. Age*Class feature after this is created to see if combinations of age and class produce significant results in correlation.

Deleting Data:- After all this we drop features which does not contribute information towards the survival rate. We drop Ticket, Name, PassengerID and Cabin feature.

After processing we now tabulate feature and its correlation value. Positive correlation increases the chance of survival as value increases. Negative correlation decreases the chance of survival as value increase. We can see that Age*Class feature we created has second highest negative value. Which means the feature contribution is significant in detecting the chance of survival.

	Feature	Correlation
1	Sex	2.201527
5	Title	0.398234
2	Age	0.287163
4	Embarked	0.261762
6	IsAlone	0.129140
3	Fare	-0.085150
7	Age*Class	-0.311200
0	Pclass	-0.749007

KNN Algorithm:- We use KNN classification algorithm. We use this to find the output, that is if user survived or not. We then calculate the accuracy of the output. Here processed training data is stored and data should be consistent. KNN makes predictions using the training dataset directly. Predictions are made for a new instance (x) by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances. For regression this might be the mean output variable, in classification this might be the mode (or most common) class value. When KNN is used for classification, the output can be calculated as the class with the highest frequency from the K-most similar instances. Each instance in essence votes for their class and the class with the most votes is taken as the prediction. KNN is imported from **sklearn.neighbors**.

Sushanth Avvaru
RedID:- 819693674

Acccuracy:- 84.739999999999995

Preceptron Algorithm:-

The perceptron is an algorithm for supervised learning of binary classifiers (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not). It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time. Perceptron is imported from **sklearn.linear_model**.

Accuracy:- 78.0