MACHINE LEARNING

PROJECT REPORT

Prediction of the survival of passengers on the Titanic using K-Nearest Neighbors as the Classification Algorithm (built from scratch)

Problem: To predict whether a passenger survived or not, based on the different features given in the Titanic dataset.

Dataset:- Titanic Dataset (taken from Kaggle)

Classification Algorithm: K-Nearest Neighbors

Prepared By: *Sarthak Goel*

Eche 19-047

Submitted On: *1-11-2022*

Dataset: The Titanic dataset contains several features like Pclass(Passenger Class), Sex, Fare, Age, SibSp (No. of Siblings/ Spouses), Ticket, Cabin, Embarked among others. It has total 10 features. The features shall be used to predict the outcome whether the passenger survived or not. Hence "Survived" is the outcome label.

Passenger survived has been assigned with the number 1. If he/she did not survive, it is assigned with 0.

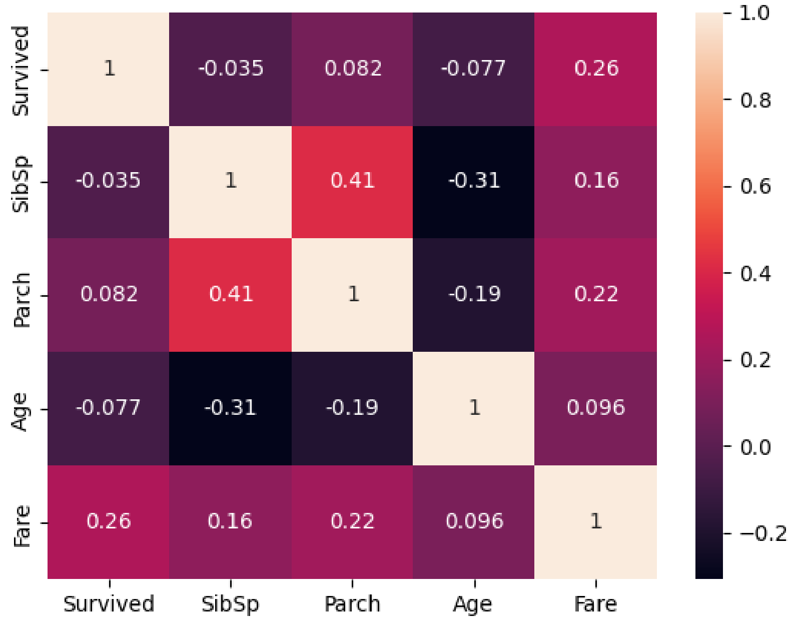
Pre-processing Of Dataset: It was seen that several columns like "Age" and "Embarked" had several entries missing and that needed to be filled before the training of Model.

1. For "Age", the distribution was assumed to gaussian/normal and hence missing values were assigned random numbers between the range of (Mean-Standard Deviation, Mean + Standard Deviation). This technique served the purpose of not altering the original distribution and also filling out the missing values.
2. For "Embarked" (Place of onboarding), missing values were filled with the entry which had the most common occurrence in the dataset.
3. Also, the dataset had "Categorical Features", which were given Integer Values for the training of K-Nearest Neighbors Classification Algorithm.
   "Male": 0 " Female" : 1
   "S": 0 "C" : 1 "Q" :2

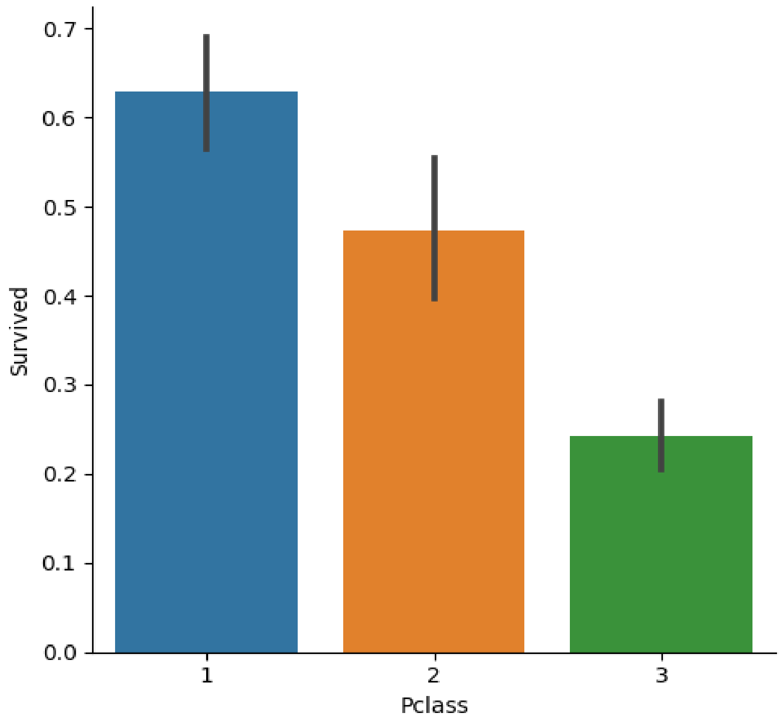
Analysis Of Correlation Between Outcome and Feature Variable:

Analysing the dataset in terms of understanding the relationship between features and label is important to understand which features have a significant correlation with the variable and hence must be included in the model, to predict the label. Whereas the features with a low correlation can be dropped from the features list.

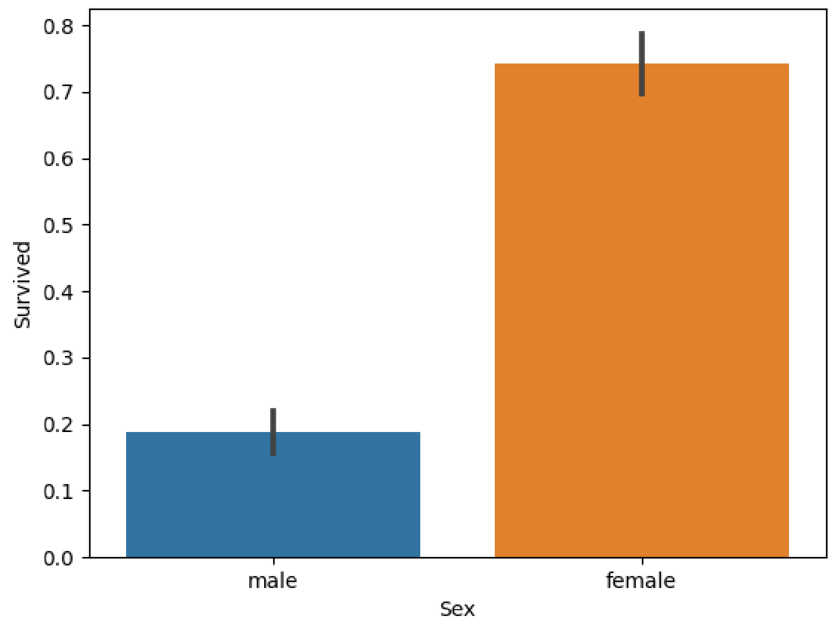
- The analysis was done by plotting Heatmaps, several categorical plots, bar graphs using the Python Inbuilt Libraries of Seaborn and Matplotlib. Graphs were drawn of different features with label and then studied.



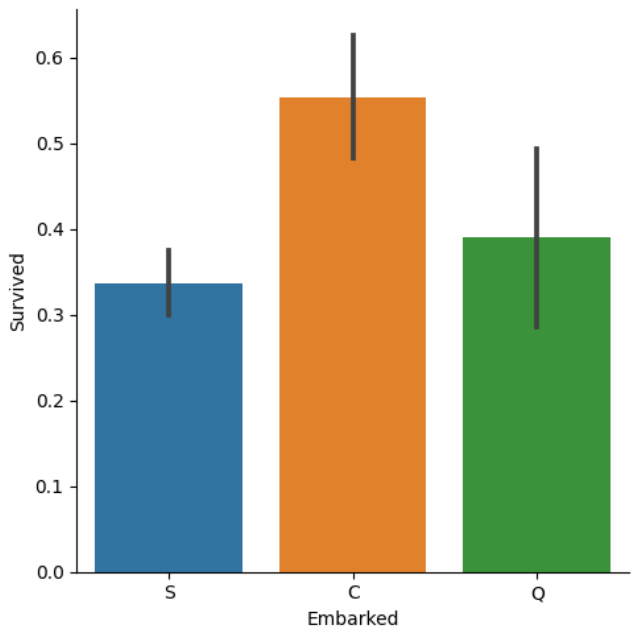
This heatmap shows the correlation between the different columns of the dataset. We observe that Fare has the highest correlation with the Survived outcome.



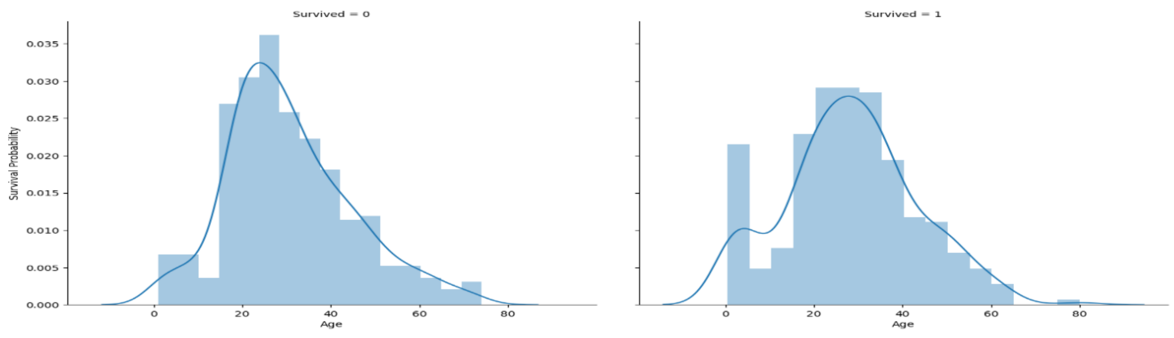
This graph demonstrates that the highest survival probability is of the people who had 1st Class ticket, as compared to 2nd and 3rd Class. Hence, it shows correlation of PClass (Passenger Class) with Survival outcome.

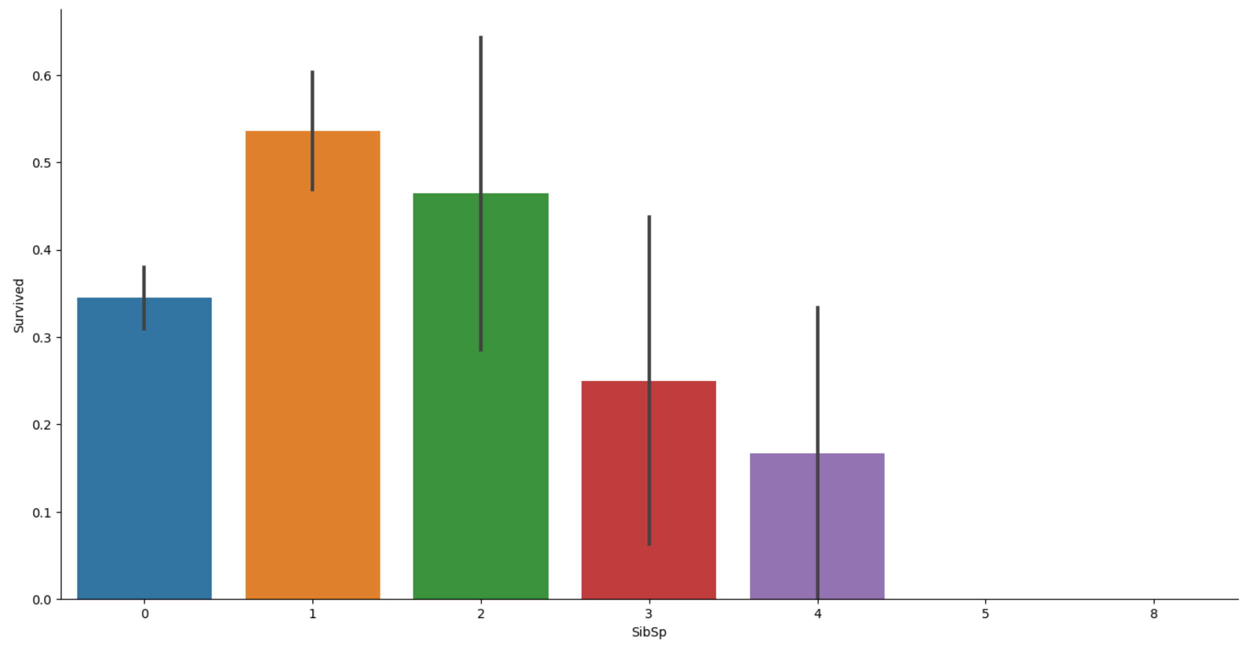


This graph clearly shows that females survived more than men and hence have a higher chance of surviving the Titanic sinking.



This graph shows that people who boarded from the C port have a higher chance of surviving than the other two ports of S and Q.





This graph shows that people who had 1 or 2 siblings or spouses, had a greater survival ratio, than people who had higher no. of siblings or spouses.

With all these graphs, we got to knew features' relationship with the label(outcome variable) better and made us realise which features to be included in K Nearest Neighbors model.

Algorithm For Building the K-Nearest Neighbors Algorithm:

1. After pre-processing the data and analysing the relationship between features and labels, the dataset was ready to be put to train. The dataset was shuffled using a python function.
2. The data was split into a (0.8:0.2) ratio with 80% data used as the training data and 20% being used as the testing data.
3. Two dictionaries were made of train and test data respectively, where the groups that were made as keys (elements of dictionaries) were 0(Not survived) and 1(Survived) and the different rows having a specific label were added to that key as list of lists.
4. Now the algorithm takes the train dataset dictionary, prediction point and value of K (3 is taken) {Since, only binary classification is done here.}, as arguments in the method.

5. The method calculates the Euclidean distance of that prediction point from each and every point of the train dataset.
6. We sort the distance list and the three groups that have least distances, that are stored. And the group which has two least distances out of 3 least ones is assigned with that prediction point.
7. If the group to which it is assigned matches with ground truth, then a correct and total pointer is increased by 1, which keeps track of how many points are being correctly classified.
8. This same process is repeated with all the prediction points of the test dataset.
9. Once the whole test dataset is iterated through with, accuracy value of the model is calculated which helps in estimating how good the model is working.

```python
def k_nearest_neighbors(data, predict, k=3):
    if len(data) >= k:
        warnings.warn('k is set to a value less than total voting groups!')
    distances = []
    for group in data:
        for features in data[group]:
            euclidean_distance = np.linalg.norm(np.array(features-np.array(predict)))
            distances.append([euclidean_distance, group])
        #print(distances)
    #print(distances)
    distances = sorted(distances)
    #print(distances)
    votes = [i[1] for i in distances[:3]]
    #print(votes)
    #print(Counter(votes).most_common(1))
    vote_result = Counter(votes).most_common(1)[0][0]
    return vote_result
```

RESULT: The accuracy, defined as the no.of training data points correctly classified out of total no. of points, of the K-Nearest Neighbors Algorithm comes out to be greater than 75% on the testing data.

Calculation of Specificity and Sensitivity:

```python
for group in test_set:
    correct = 0
    total = 0
    for data in test_set[group]:
        vote = k_nearest_neighbors(train_set, data, k = 5)
        if group == vote:
            correct+=1
        total+=1
    print("Accuracy:", correct/total)
```

This code is used for calculation of sensitivity and specificity. When the group has value ‘0’ means people who did not survive: Accuracy gives us Specificity which comes out to be 91.66% and when group has value of ‘1’, means people who did survive, the accuracy gives us value of sensitivity which comes out to be 51.42%.

The specificity is higher than sensitivity, because the whole dataset has more points of people who did not survive, hence is trained better at classifying people who didn’t survive, than compared to people who did survive.