MACHINE LEARNING

Titanic Disaster

PROJECT REPORT

A REPORT BY:

Akshat Rastogi

B.Tech CSE,
Bennett University

Introduction

REPORT TITLE

Complete Titanic Survival Prediction Analysis

AUTHOR

Akshat Rastogi

TOPICS COVERED

- Scene Setting
- Problem Analysis
- Descriptive Analysis of Dataset
- Data Preprocessing
- Model Fitting
- Results

SETTING

The scene is set in the North Atlantic Ocean where the HMS Titanic Crashed. Using advanced machine learning techniques, I can make the machine accurately predict the chances of a passenger surviving based on the data I have on the passenger.

OPINION

I faced a lot of challenges while trying to accurately predict the outcome, but by overcoming each challenge, I leaned something new.

SCENE SETTING

One of the most well-known shipwrecks in history was the sinking of the RMS Titanic. RMS Titanic sunk on April 15, 1912, after striking an iceberg while on a journey. As a result of the lack of available lifeboats, a total of 1502 passengers and crew members perished.

More than 2200. Even if chance had a role, it appears that certain individuals had a higher probability of surviving than others.

So in this project, there are two sets of Titanic passenger data: a training set and a test set, both of which are.csv files. The "Survived" response variable, as well as 11 other 891-passenger informative factors, were included in the training dataset.

The goal of this project is to create a machine learning models in order to forecast which people survived the shipwreck. The response variable Survived will be modeled in specific, given 10 different predictors. The rest of this paper goes through the procedures that were used to create the predictive model. We'll create models to forecast which individuals are more likely to survive. Also, compare the models to see which is the most effective.

PROBLEM ANALYSIS

An examination of the Titanic's historical report provides useful insight into the passenger data in terms of survival.

1. Because of a "women and children first" protocol for filling lifeboats, a disproportionate number of men were left on board.

2. Passengers in first and second class were the most likely to make it to the lifeboats. To get to the boat deck, third-class passengers had to navigate a tangle of passageways and staircases.

3. Many lifeboats were barely partially loaded when they were deployed.

Given that the Titanic sank after striking with an iceberg in the North Atlantic Ocean, it's safe to assume that everybody who didn't make it onto a lifeboat died of exposure to the cold. Anyone who was not in a group of female youngsters and who did not board a lifeboat before they were all launched was unlikely to live.

The training dataset contains 891 rows. We can also see that the training dataset only has 714 Age values. We know from our historical research that women and children were given priority while loading lifeboats. As a result, we will require a method of identifying children. Due to the enormous number of missing age data, this is a difficult task.

DESCRIPTIVE ANALYSIS OF DATASET

I performed a descriptive analysis of the dataset in order to gain maximum insight into the passengers aboard the Titanic. This analysis helped me identify things like gender ratio, people having different class tickets, age variance, etc.

But first, I identified what the given dataset contained. The values in the train dataset had the following structure and values

```
train.dtypes
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

```
[8]  train.head()
```

	PassengerId	Survived	Pclass	Name	Sex
0	1	0	3	Braund, Mr. Owen Harris	male
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	3	1	3	Heikkinen, Miss. Laina	female
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	5	0	3	Allen, Mr. William Henry	male

Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
22.0	1	0	A/5 21171	7.2500	NaN	S
38.0	1	0	PC 17599	71.2833	C85	C
26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
35.0	1	0	113803	53.1000	C123	S
35.0	0	0	373450	8.0500	NaN	S

The dataset contained 5 categorical columns as shown below :

```
Name : 891 labels
Sex : 2 labels
Ticket : 681 labels
Cabin : 148 labels
Embarked : 4 labels
```

But the main categorical data that can be identified are the fields of sex and embarked. Also, all of this categorical data needed to be converted into numerical data as many machine learning models are unable to work with categorical fields.

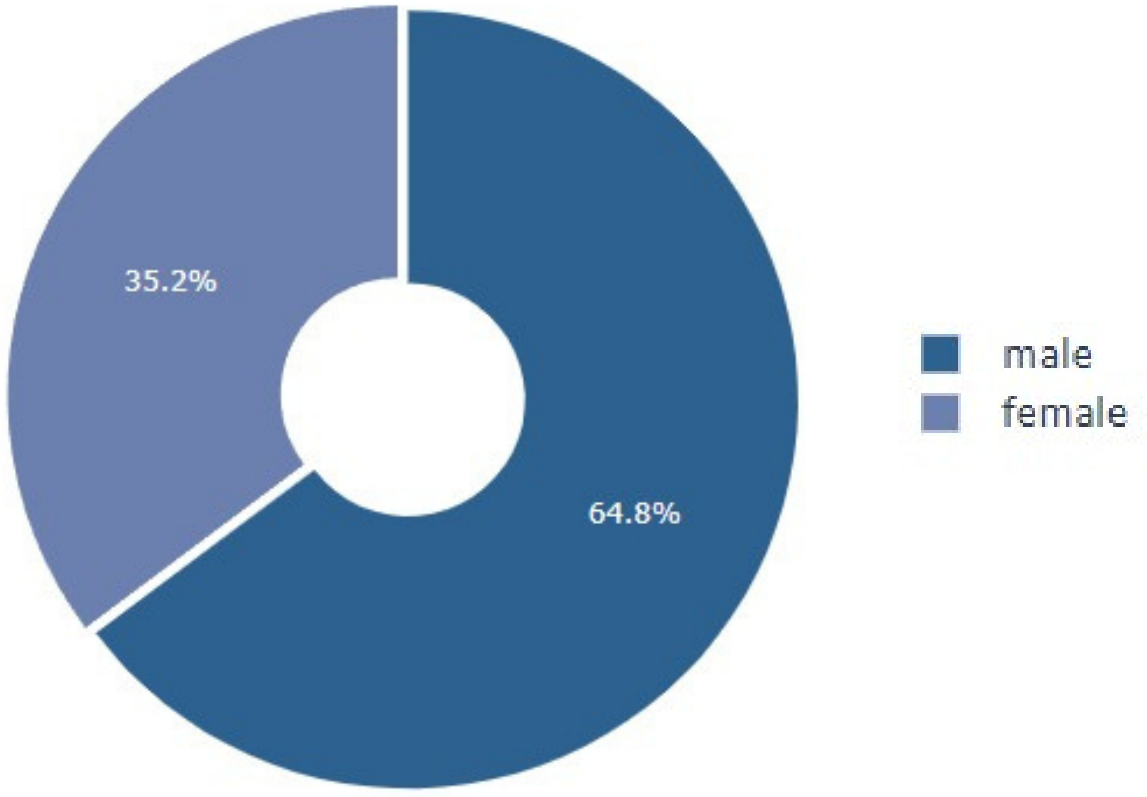
Also to be addressed were the missing values in the age and cabin fields as there are 891 rows in the training dataset. Using describe() we see the training dataset contained only 714 values for Age.

```
train.describe()
```

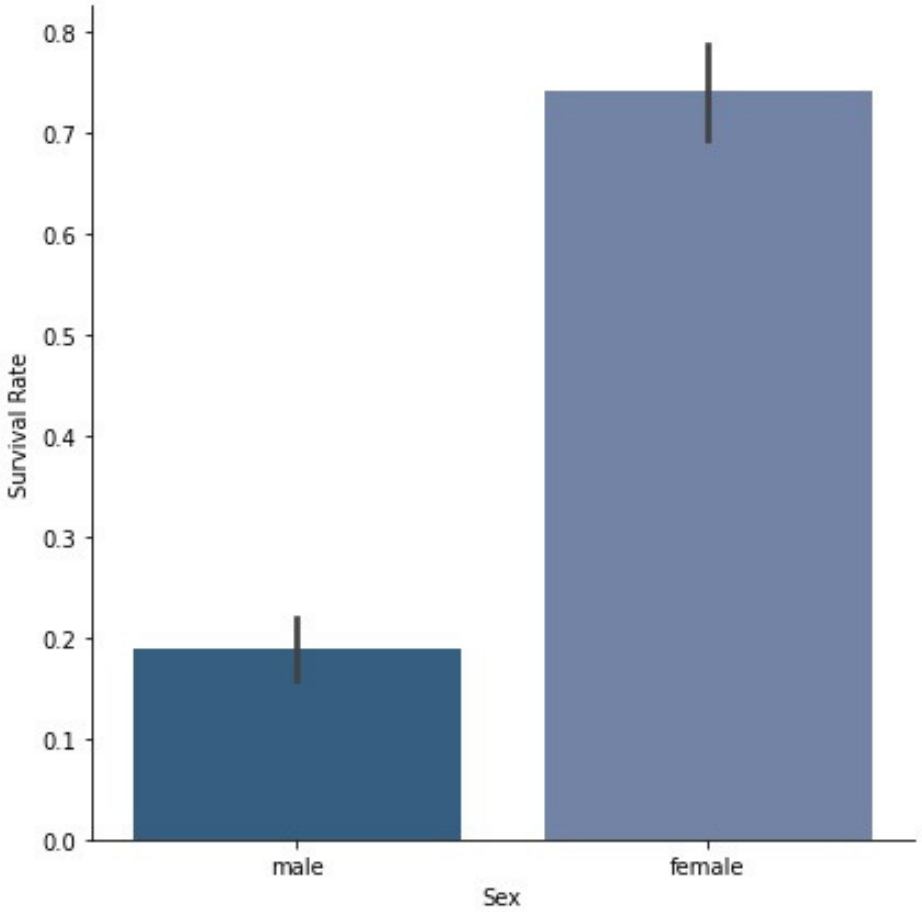
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

I know from my historical research that women and children were given priority while loading lifeboats. As a result, I required a method of identifying children. Due to the enormous number of missing age data, this was a difficult task.

Next, I did a gender analysis to identify the male-female distribution on the ship. The following results were observed:



The total number of males was significantly higher. Keeping that in mind I also did a gender vs survival rate analysis to confirm the assumption I gathered from historical report of the Titanic.

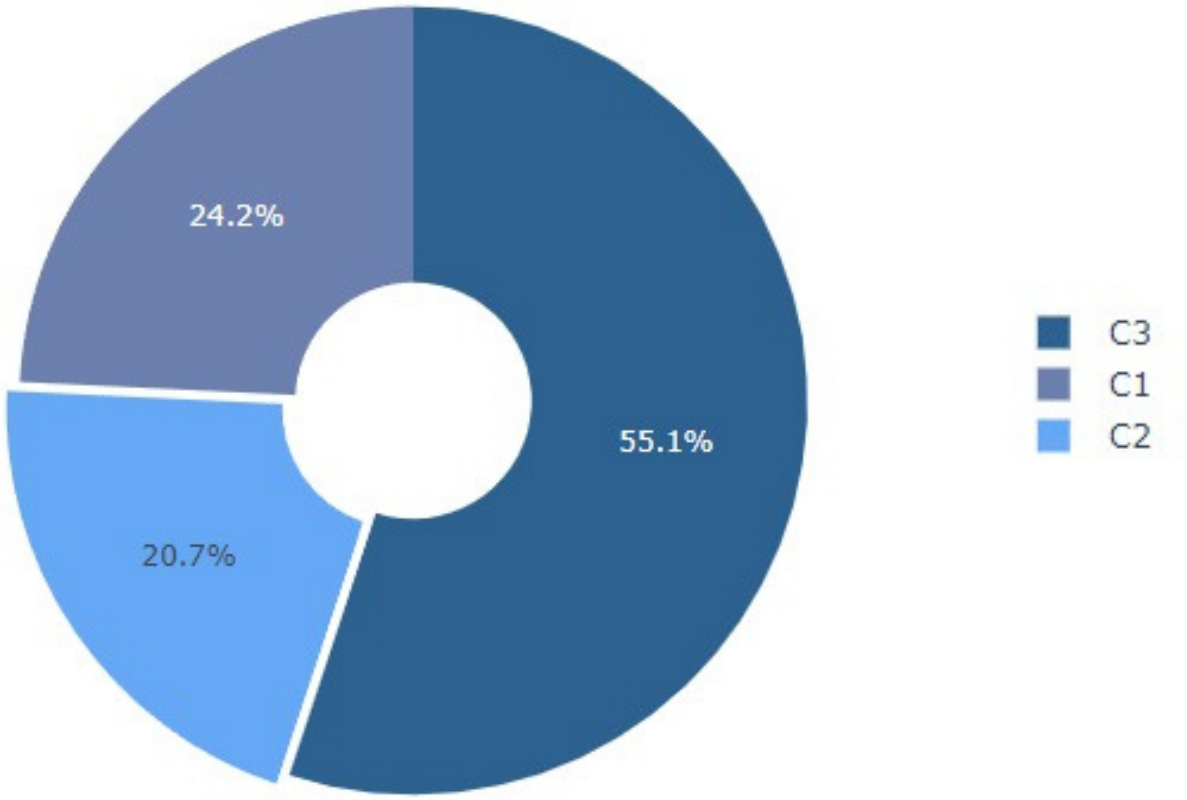


From the above graph, it can be observed that even though the number of males was much higher than females, the survival rate of men was significantly lower. This also supports my historical assumption about the Ladies and Children First protocol.

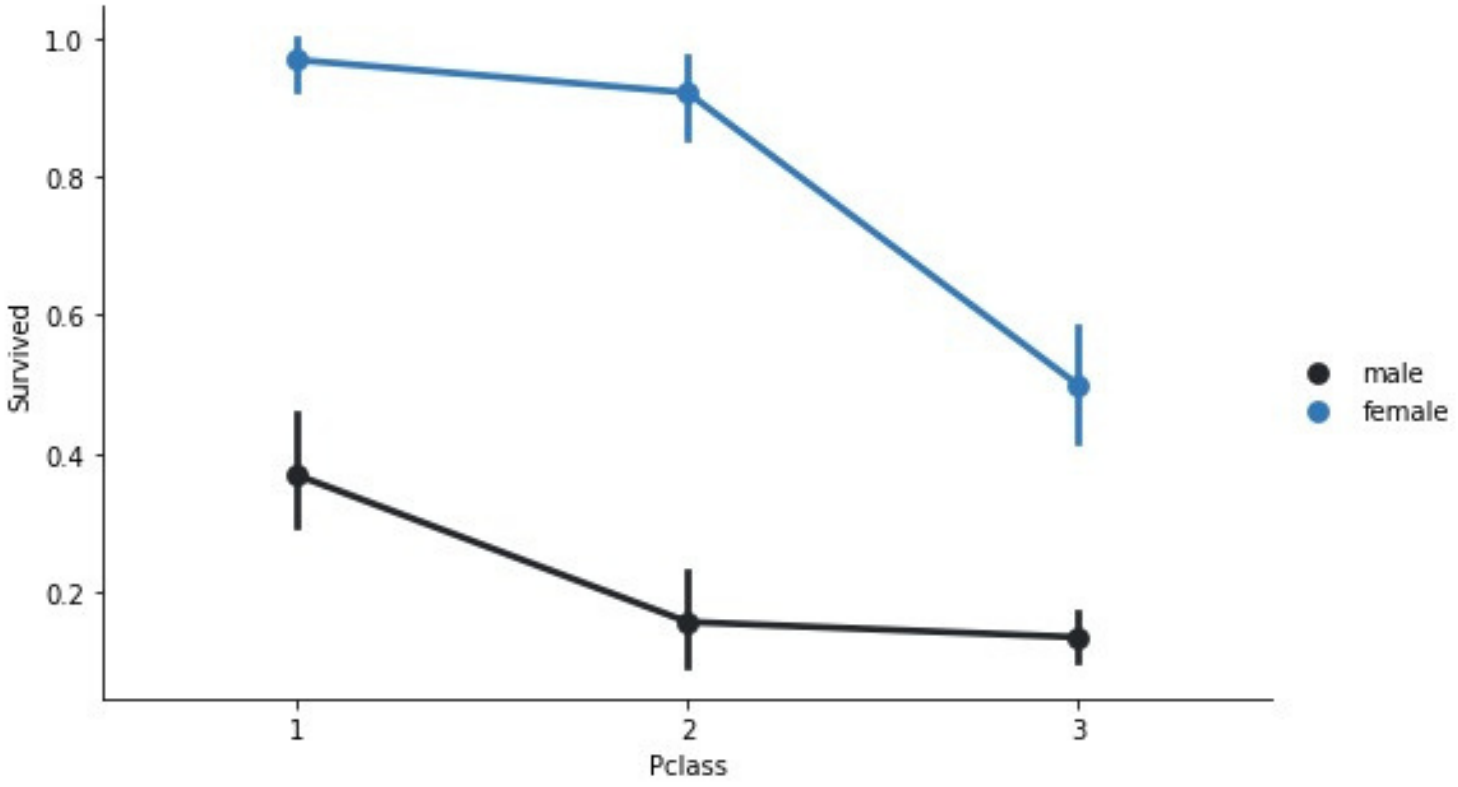
Next up is the analysis of the distribution of people in different passenger classes.

C1 : First Class
C2 : Second Class
C3 : Third Class

Graph on next page

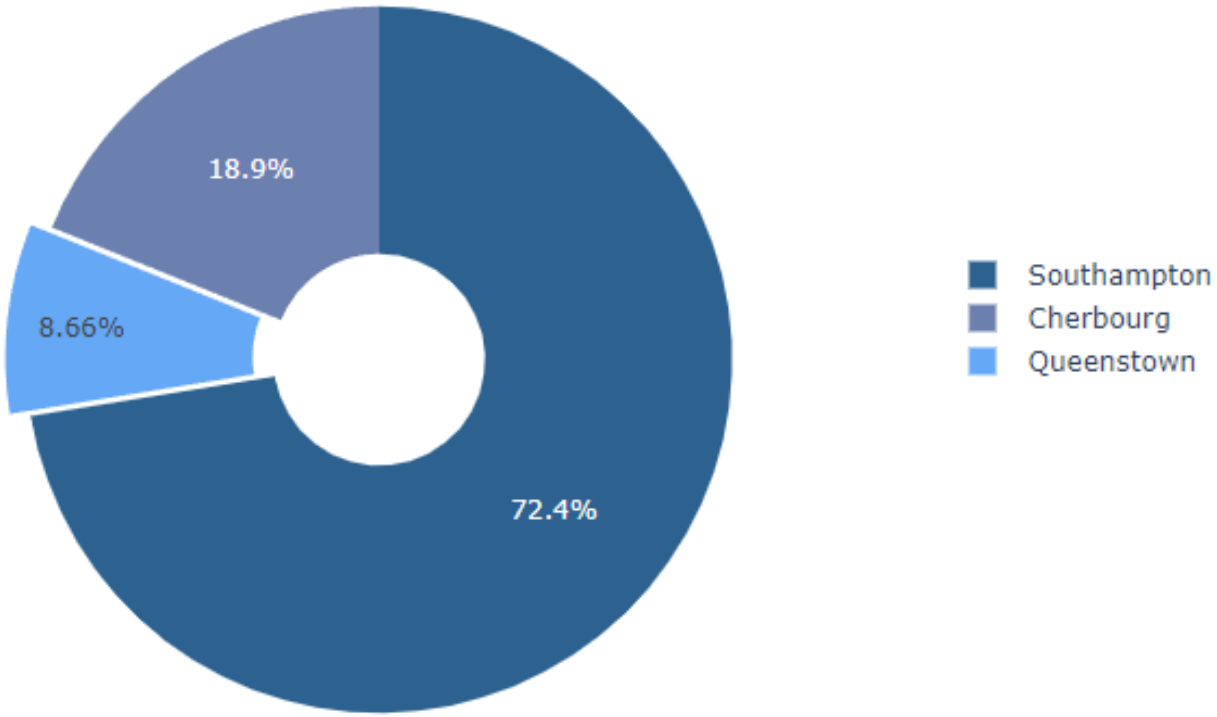


According to the aforementioned analysis, the bulk of passengers are having a third-class ticket. From the historical data, we know that third-class people had to travel a lot in order to reach the safe area when compared to the people having first and second-class tickets. Keeping that in mind I also did a passenger class vs survival rate analysis on the basis of gender to confirm the assumption I gathered from the historical report of the Titanic.



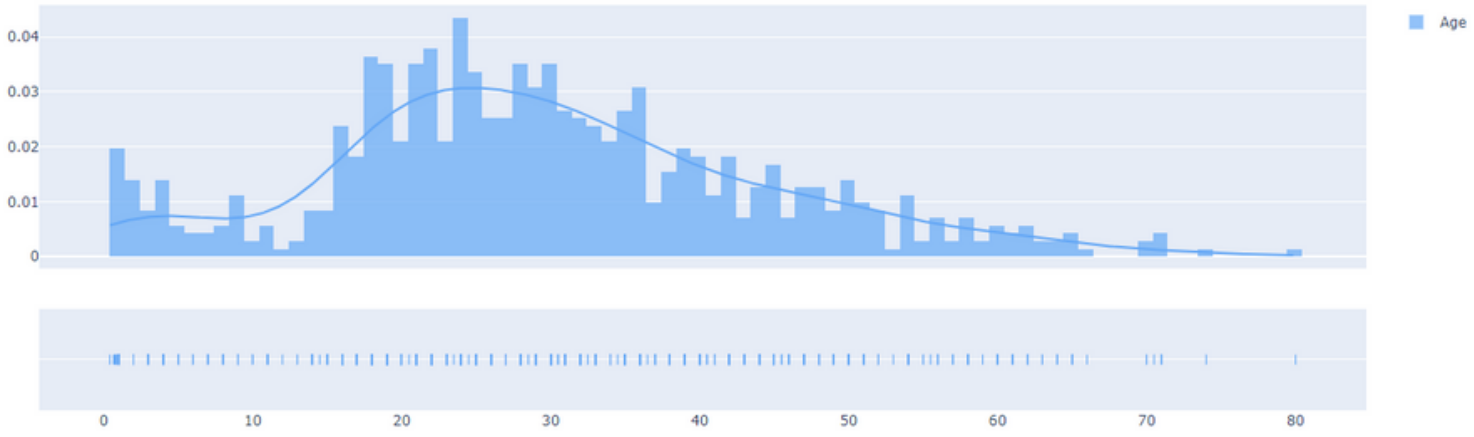
According to the aforementioned analysis, females traveling in the third class had a considerably lower survival rate. This backs up the historical assumption that first and second-class passengers were the most likely to make it to the lifeboats, which were hurriedly released partially loaded.

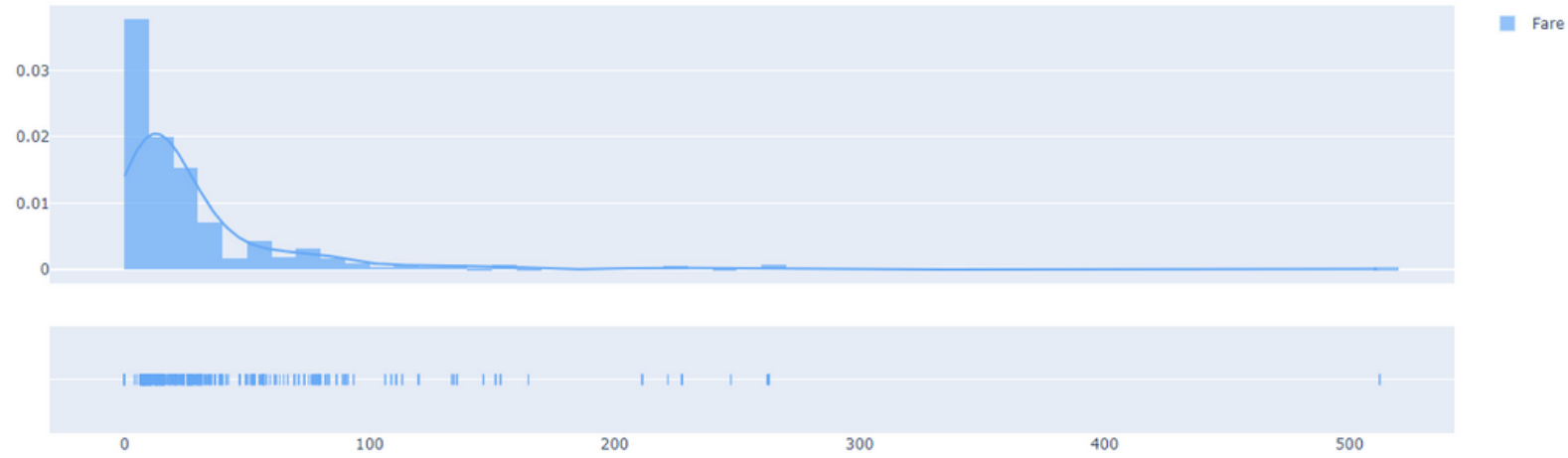
Next up is the analysis of the distribution of people based on their port of embarkation.



According to the aforementioned analysis, it s observed that an overwhelming number of people embarked from Southampton also encoded as S.

Upon further analysis, It was found out that apart from the missing values in the age variable, the age and fare values were positively skewed.





These skewed values had to be normalized by using log transformations.

FEATURE ENGINEERING

One-Hot Encoding is a crucial step in preparing data for machine learning models. Many machine learning algorithms are unable to directly work on categorical data such as gender (male/female). This implies that category data must be transformed into numerical data. Feature Engineering is a critical phase in designing any prediction system because the data may have missing fields, incomplete fields, or fields containing secret information. Age, Fare, and Embarked, for example, had missing values in the training and testing data that needed to be filled up. I also used the passenger's surname to distinguish families on board the Titanic.

First, I removed the name, passenger id, and ticket variables because they all have unique values, and creating dummies for them would increase the dimensionality. The structure of train dataset after dropping values was:

```
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Sex       891 non-null    object
 3   Age       714 non-null    float64
 4   SibSp     891 non-null    int64
 5   Parch     891 non-null    int64
 6   Fare      891 non-null    float64
 7   Cabin     204 non-null    object
 8   Embarked  889 non-null    object
dtypes: float64(2), int64(4), object(3)
memory usage: 62.8+ KB
```

Now I addressed the missing values in the dataset. The following are the percentages of the missing values.

```
Age 19.87  % missing values.

Cabin 77.1  % missing values.

Embarked 0.22  % missing values.
```

Because the cabin variable had over 77 percent missing values, I eliminated it. However, I can impute missing values as "other" by introducing another category. After doing so, only 2 categorical columns remained ie Sex and Embarked .

Since there were only two missing values in Embarked variable so I imputed them with the mode of the rest of the values. Next to fill up the missing age values I used random values in between the 25th and 75th percentile. In the test set there were also missing values for fare variable. Those were also imputed from the values range in the training set.

Age

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

Fare

```
count    891.000000
mean      32.204208
std       49.693429
min        0.000000
25%        7.910400
50%       14.454200
75%       31.000000
max      512.329200
Name: Fare, dtype: float64
```

Now I had to convert the categorical data into numerical one. So the male and female of the sex field were encoded as 1 and 0 respectively.

Next I had to create dummy variables for different Embarked classes like Embarked_C, Embarked_Q, Embarked_S. Now these values will be binary encoded so that at a time only one of these values will be 1 signifying that the given person embarked from that location. So for example a person embarked from a location C, then in the row, where that person's details are mentioned, the Embarked_C column will contain the value 1 while others ie Embarked_Q and Embarked_S value will contain 0.

WHAT ARE THE CORRELATIONS IN THE DATASET ?

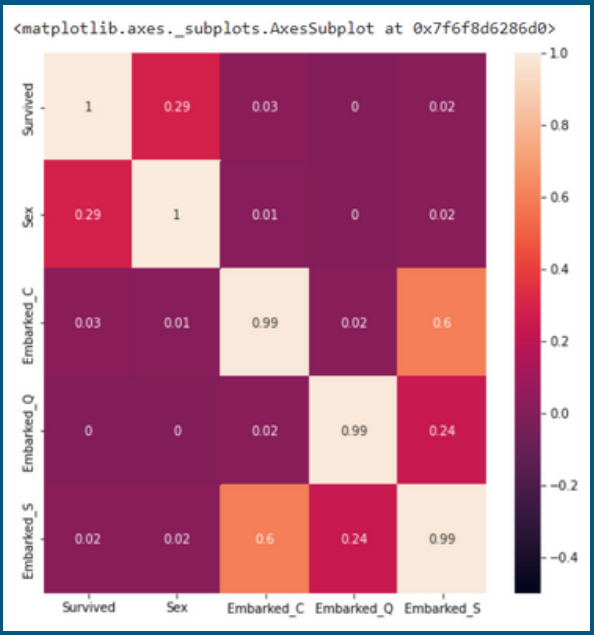
Next, we move on to the correlations in the dataset.

Through the use of data correlations, we can determine how many variables and attributes are associated in your dataset. If one or more attributes are dependent on another attribute, or if one or more attributes act as a catalyst for another attribute, we can infer this from correlation.

In this we analyze correlations between numerical and numerical variables, numerical and categorical variables and categorical and categorical variables.

	Survived	Sex	Embarked_C	Embarked_Q	Embarked_S
Survived	1.00	0.29	0.03	0.00	0.02
Sex	0.29	1.00	0.01	0.00	0.01
Embarked_C	0.03	0.01	0.99	0.02	0.61
Embarked_Q	0.00	0.00	0.02	0.99	0.24
Embarked_S	0.02	0.01	0.61	0.24	0.99

To generate the above table we use Cramer's V test. Cramer's V table helps the association or correlation between two variables



To visualize the correlation matrix, I used a seaborn heatmap. The diagonal of the correlation map is all 1 which is because each variable is correlated to itself.

From the other values in the heatmap, we can see that most of the features, don't have a high correlation with any of the other features. With this I concluded the Feature Engineering and Data Preprocessing of the dataset.

MODEL FITTING

To create a machine learning model I used the following classifiers:

1. Logistic regression classifier
2. Random forest classifier
3. Decision tree classifier
4. Adaboost classifier
5. Ridge Classifier

Logistic Regression Classifier

A classification process known as logistic regression gives observations to a set of discrete groups. Online transactions that are either fraudulent or lawful, and tumours that are either malignant or benign are examples of classification challenges. The Logistic Sigmoid function is used to convert the results of a logistic regression into a probability value.

Logistic Regressions are of two types:

1.Binary (eg Email spam or not)
2.Multi Linear (eg Cat, Dog, or Rabbit)

Logistic Regression is mostly used for classification problems as its predictive analysis algorithm is based on probability

LOGISTIC REGRESSION RESULTS

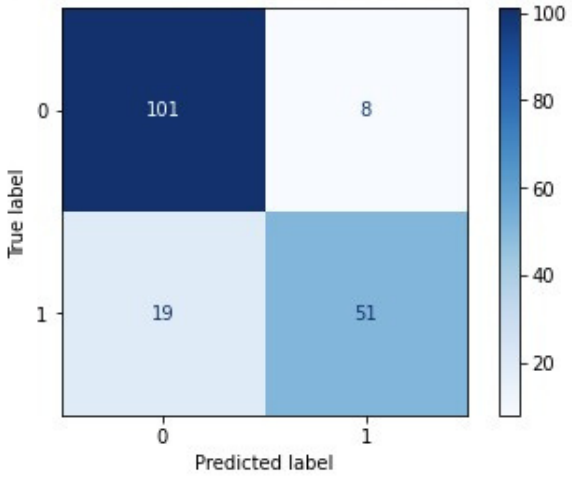
```
train accuracy: 0.797752808988764
 test accuracy: 0.8491620111731844


classification report for logistic regression
              precision    recall  f1-score   support

           0       0.93      0.84      0.88       120
           1       0.73      0.86      0.79        59

    accuracy                           0.85       179
   macro avg       0.83      0.85      0.84       179
weighted avg       0.86      0.85      0.85       179
```

confusion matrix for logistic regression



Random Forest Classifier

Random forest, a machine learning approach, may be used to address classification and regression problems. The majority vote is used for categorization, whereas the average is used for regression.

The Random Forest Algorithm can handle both continuous and categorical data sets, hence it may be useful for both regression and classification. When it comes to classification issues, it's the best.

A random forest that has too many trees is slow and ineffective at making real-time forecasts, which is the primary drawback of random forests. Algorithms that learn rapidly can take a long time to provide predictions, which is not uncommon.

RANDOM FOREST RESULTS

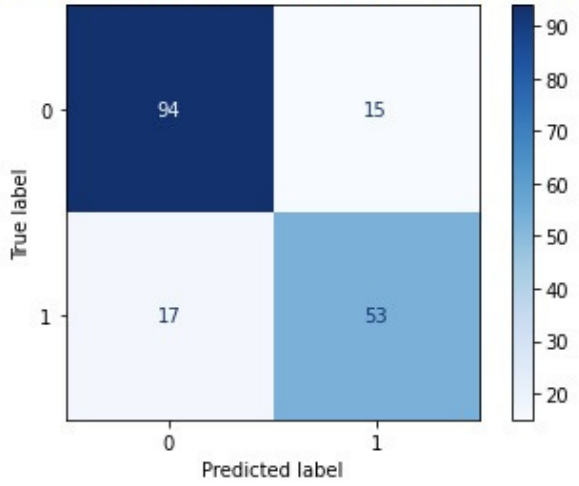
```
train accuracy: 0.9831460674157303
 test accuracy: 0.8212290502793296


classification report for random forest classifier
              precision    recall  f1-score   support

           0       0.86      0.85      0.85       111
           1       0.76      0.78      0.77        68

    accuracy                           0.82       179
   macro avg       0.81      0.81      0.81       179
weighted avg       0.82      0.82      0.82       179


confusion matrix for random forest classifier
```



Decision Tree Classifier

Machine Learning Algorithms like Decision Trees can be used to make judgments in the same way that humans do, by following a set of rules. It is a member of the supervised learning family of algorithms. Regression and classification problems can't be solved using other supervised learning techniques, such as the decision tree methodology. A decision tree can be used to learn the class or value of a target variable and then used to generate predictions about the value of that variable.

Despite being a basic algorithm, decision trees have various advantages:

1. Visualization: The decision tree can be visualized.
2. No data preprocessing is required: you do not need to prepare the data before developing the model.
3. Data robustness: the method is capable of handling a wide range of data.

DECISION TREE RESULTS

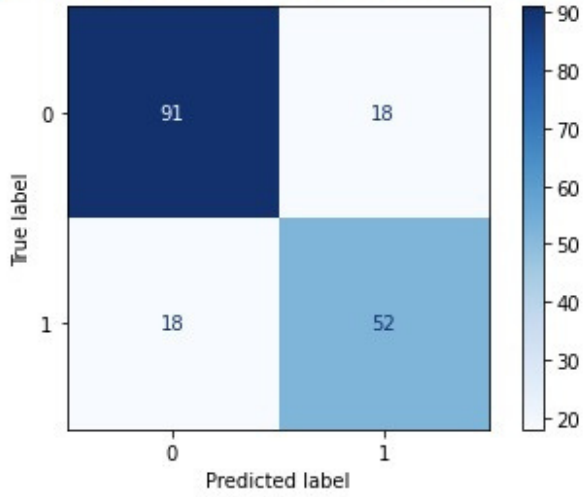
```
train accuracy: 0.9831460674157303
 test accuracy: 0.7988826815642458


classification report for decision tree classifier
              precision    recall  f1-score   support

           0       0.83      0.83      0.83       109
           1       0.74      0.74      0.74        70

    accuracy                           0.80       179
   macro avg       0.79      0.79      0.79       179
weighted avg       0.80      0.80      0.80       179
```

confusion matrix for decision tree classifier



Adaboost Classifier

Machine learning is used in Adaboost Classifier.
AdaBoost, which stands for "adaptive boostering," is a common approach in machine learning. Each occurrence is re-weighted, with bigger weights assigned to cases that were incorrectly identified. For this, "Adaptive Boosting" is the term.

By modifying the weighted training data, it is possible to forecast a series of weak learners. All observations are weighted the same at the beginning of the experiment. If the first learner is wrong, observations that they had wrongly predicted are given more weight. Continuous growth is achieved by an iterative learning process that adds new students until the desired number of models or accuracy is obtained.

AdaBoost methods can be used to address classification and regression problems.

ADABOOST RESULTS

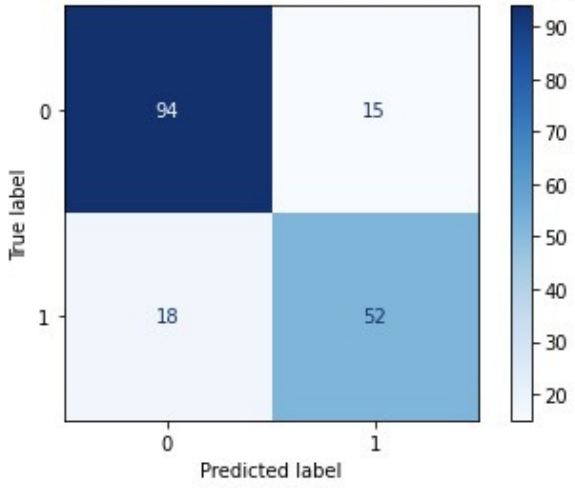
```
train accuracy: 0.8314606741573034
 test accuracy: 0.8156424581005587


classification report for adaboost classifier
              precision    recall  f1-score   support

           0       0.86      0.84      0.85       112
           1       0.74      0.78      0.76        67

    accuracy                           0.82       179
   macro avg       0.80      0.81      0.80       179
weighted avg       0.82      0.82      0.82       179
```

confusion matrix for adaboost classifier



Ridge Classifier

The Ridge Classifier solves the challenge by employing the Ridge regression technique to translate the label data into the interval [-1, 1]. Using multi-output regression, the target class with the highest prediction value is chosen for multiclass data.

Ridge regression employs a shrinkage estimator known as a ridge estimator. Decreased estimations may produce more accurate figures that are in line with the actual population. If multicollinearity is present, the ridge estimator has a particularly strong influence on the least squares estimate.

Ridge regression is a subset of L2 regularization-based regression.

RIDGE RESULTS

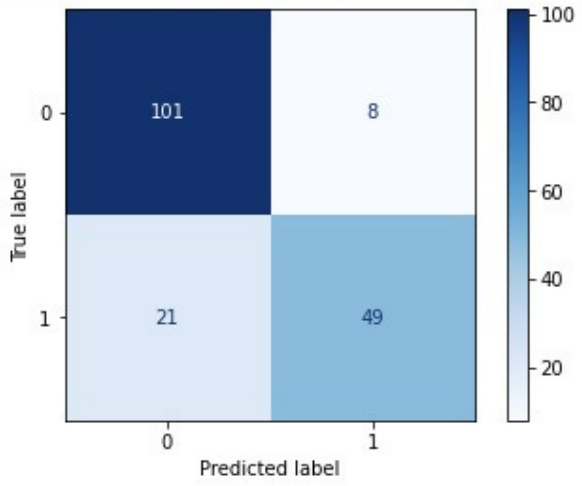
```
train accuracy: 0.7893258426966292
 test accuracy: 0.8379888268156425


classification report for logistic regression
              precision    recall  f1-score   support

           0       0.93      0.83      0.87       122
           1       0.70      0.86      0.77        57

    accuracy                           0.84       179
   macro avg       0.81      0.84      0.82       179
weighted avg       0.85      0.84      0.84       179
```

confusion matrix for logistic regression



OVERALL RESULTS

The table of the score model is given below. According to this Logistic Regression was able to gain the highest test accuracy out of all other classifiers with an accuracy score of 84.9%.

	Train Accuracy	Test Accuracy
Logistic Regression	79.77	84.9
Random Forest	98.31	82.12
Decision Tree	98.31	79.88
Adaboost	83.14	81.56
Ridge	78.93	83.79

Result Visualization

CONCLUSION

The role of machine learning applications in disaster management and predictions has been increasing rapidly over the past years. Following this trend, I was given an assignment to predict the survival of passengers aboard the infamous Titanic Ship sailing across the North Pacific Region.

As a consequence of my efforts, I obtained important knowledge in the development of prediction algorithms and set a high of 84.9 percent accuracy **in the "Titanic - Machine Learning from Disaster" competition organized by** Kaggle.

From my work in building machine learning models to predict the survival of passengers aboard the titanic, I'd rather be a young female with a first class ticket. The accuracy that I was able to obtain can be further increased by tuning hyper parameters of these classifiers.