EXTRA CREDIT ASSIGNMENT
PART-2: REPORT

Name: Visrutha Balashanmugam **UTA ID:** 1002074535

1. ABSTRACT

Titanic is a classic incident in history that reminds everyone about the disaster of the unsinkable ship and one of the largest, luxurious ships of the time. The survival of passengers in the Titanic during shipwreck has been created as a dataset that can be used for the purpose of data analysis. The main aim of the project is to predict the survival of the passenger in The Titanic after its impact with the iceberg. Here machine learning algorithms are used to determine whether a passenger would have survived the accident based on their attributes. In this case study, the data is explored and exhibited in the form of visual graphs and plots.

2. MOTIVATION

The business problem helps us to identify the dependent and independent variables, the real challenge is to select the most impacting variables. Understanding the data and identifying variables using frequency analysis in Python, then using data analysis techniques and implementing them. Converting our assumptions into practice is using regression techniques is needed here.

The data given is mostly a mix of categorical and nominal data where we need to find the variables associated with survival of passengers having a significant correlation between them. Manually trying to weigh these factors against every other features of the passenger would be extremely time-consuming, which makes this a great problem for data modelling to solve.

As our basis of modelling, we will sort the data where passenger features are important and survival (Y/N) is a dependent variable (response variable) and others are independent variable (predictor).

3. RESEARCH QUESTIONS

Some research questions that can be explored on with this study are

- Who is more likely to survive this disaster: male or female?
- Do people belonging to high class, travelling the first and second class have more chances of persist through the impact?
- Are people who travel with their family and friends more likely to expire in this catastrophe?
- Is age a determining factor that impacts the endurance of a passenger in this situation?

4. ABOUT THE DATA

The titanic dataset contains information of the passengers who boarded the RMS Titanic which shipwrecked. The data contains 1039 records of 11 different attributes (train and test data combined). The attributes of the data include:

- Passenger id: Unique passenger id for each passenger
- Survival: If the passenger survived or not (0 = No, 1 = Yes)
- Pclass: Class of the passenger's ticket (classes 1 = 1^{st}, 2 = 2^{nd}, 3 = $3^{rd)}$
- Sex: Gender of the passenger (male/female)
- Age: Age of passenger in years
- Sibsp: Number of siblings (or) spouses on the ship,
- Parch: Number of parents (or) children on the ship
- Ticket: Ticket number
- Fare: Ticket price
- Cabin: Cabin number
- Embarked: Port of Embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

The first few rows of the data are presented below:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Of these features, they are of different types.

Numerical features:

Continuous: Age, Fare

Discrete: SibSp, Parch

Categorical features: Survived, Sex, Embarked, Pclass

#	Column	Non-Null Count	Dtype
0	PassengerId	1309 non-null	int64
1	Survived	891 non-null	float64
2	Pclass	1309 non-null	int64
3	Name	1309 non-null	object
4	Sex	1309 non-null	object
5	Age	1046 non-null	float64
6	SibSp	1309 non-null	int64
7	Parch	1309 non-null	int64
8	Ticket	1309 non-null	object
9	Fare	1308 non-null	float64
10	Cabin	295 non-null	object
11	Embarked	1307 non-null	object

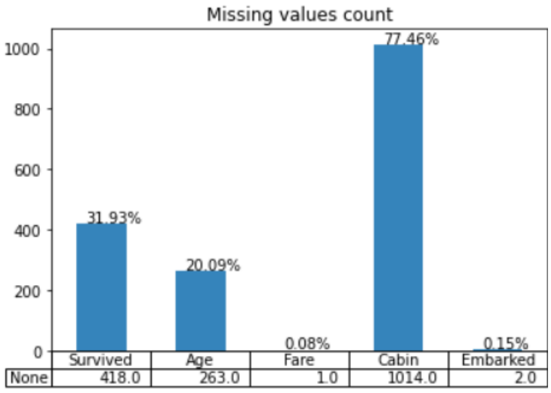
The description of data is presented below:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	1309.000000	891.000000	1309.000000	1046.000000	1309.000000	1309.000000	1308.000000
mean	655.000000	0.383838	2.294882	29.881138	0.498854	0.385027	33.295479
std	378.020061	0.486592	0.837836	14.413493	1.041658	0.865560	51.758668
min	1.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	328.000000	0.000000	2.000000	21.000000	0.000000	0.000000	7.895800
50%	655.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	982.000000	1.000000	3.000000	39.000000	1.000000	0.000000	31.275000
max	1309.000000	1.000000	3.000000	80.000000	8.000000	9.000000	512.329200

5. DATA CLEANING

In data cleaning process, we are going to prepare our data for analysis and modelling by removing and correcting irrelevant data, dealing with missing values, filtering the outliers and removing duplicate data. In this dataset, I am cleaning the data by following 3 steps:

- Dropping columns with high amount of missing values
- Imputing values to the data for necessary features
- Correcting outliers



Step - 1: Dropping columns.

From the bar chart on the right, it can be observed that the feature names ‘Cabin’ has the highest number of missing values which is 77%. It is a tiresome task to impute those values, since it is missing at random.

```
trainTest.drop(['Cabin'], axis=1, inplace=True)
```

Step – 2: Imputing values to the data for necessary features

The primary method for imputing data is replacing null values of numerical and categorical features with mean or median, and mode respectively. In this dataset, the numerical features with missing values are ‘Age’ and ‘Fare’. The categorical variable with missing value is ‘Embarked’. I am trying to replace the values of ‘Age’ and ‘Fare’ with median and the variable ‘Embarked’ with the mode of the feature values.

FARE

For ‘Fare’ feature, I am grouping three other features of the dataset – ‘Parch’, ‘SibSp’ and ‘PClass’, with which the median is replaced to the ‘Fare’ feature instead of the null values. The null values in this variable are

```
trainTest[trainTest['Fare'].isnull()]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1043	1044	NaN	3	Storey, Mr. Thomas	male	60.5	0	0	3701	NaN	NaN	S

Replacing median by groupby() function

```
trainTest['Fare'] = trainTest['Fare'].fillna(trainTest.groupby(['SibSp','Parch','Pclass']).Fare.median()[0][0][3])
```

```
trainTest['Fare'].isnull().sum()
```

```
0
```

AGE

Age is also filled with the median of the feature values. But it is a more complicated process than that of ‘Fare’ replacement. We first develop a new feature called ‘Title’ that closely resembles Name, and we then fill in the missing age for a certain passenger using the median of the Title that this passenger possesses.

The null values of ‘Age’ are

```
trainTest[trainTest['Age'].isnull()]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
5	6	0.0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
17	18	1.0	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	S
19	20	1.0	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C
26	27	0.0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN	C
28	29	1.0	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN	Q
...	...	...	...	...	...	...	...	...	...	...	...	...
1299	1300	NaN	3	Riordan, Miss. Johanna Hannah""	female	NaN	0	0	334915	7.7208	NaN	Q
1301	1302	NaN	3	Naughton, Miss. Hannah	female	NaN	0	0	365237	7.7500	NaN	Q
1304	1305	NaN	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
1307	1308	NaN	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
1308	1309	NaN	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

263 rows × 12 columns

Creating and extracting 'Title' with 'Name' feature and grouping those features into 3 groups with highest frequencies. Followed by, replacing null values by the median. The screenshots are attached below.

```
trainTest['Title'] = trainTest['Name'].str.split(', ', expand = True)[1].str.split('.', expand = True)[0]
trainTest['Title'].value_counts()
```

```
Mr              757
Miss            260
Mrs             197
Master           61
Rev               8
Dr                8
Col               4
Mlle              2
Major             2
Ms                2
Lady              1
Sir               1
Mme               1
Don               1
Capt              1
the Countess      1
Jonkheer          1
```

```
common_title = ['Mr', 'Mrs', 'Master']
trainTest['Title'].replace(['Mlle','Ms','Mme', 'Miss', 'Lady', 'Dona', 'the Countess'],'Mrs', inplace=True)
trainTest['Title'].replace(['Sir', 'Rev', 'Capt', 'Don', 'Major', 'Col', 'Dr', 'Jonkheer'],'Mr', inplace=True)
# Check the Title we have
trainTest['Title'].value_counts()
```

```
Mr        783
Mrs       465
Master     61
Name: Title, dtype: int64
```

```
age_median_by_Title = trainTest.groupby('Title')['Age'].median()
for title in age_median_by_Title.index:
    trainTest['Age'][(trainTest.Age.isnull()) & (trainTest.Title == title)] = age_median_by_Title[title]
```

EMBARKED

Since 'Embarked' is a categorical variable, I am replacing the null values with the mode of the feature values. The missing values of the feature are given below.

```
trainTest[trainTest['Embarked'].isnull()]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
61	62	1.0	1	Icard, Miss. Amelie	female	38.0	0	0	113572	80.0	B28	NaN	Mrs
829	830	1.0	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572	80.0	B28	NaN	Mrs

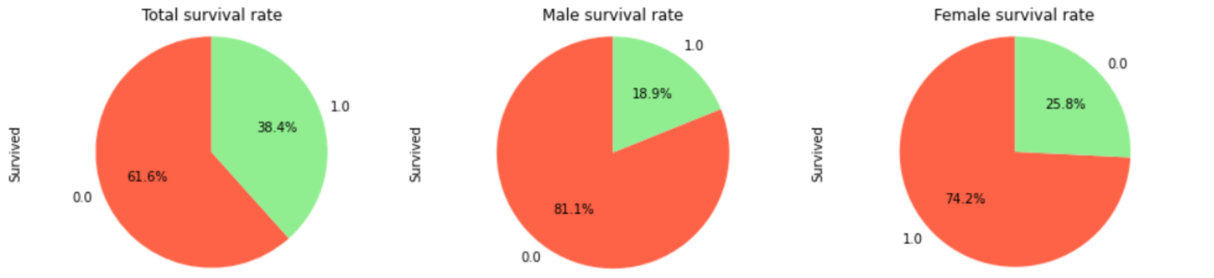
```
trainTest['Embarked'].fillna(trainTest['Embarked'].mode()[0], inplace = True)
```

```
trainTest['Embarked'].isnull().sum()
```

```
0
```

6. ANALYSIS

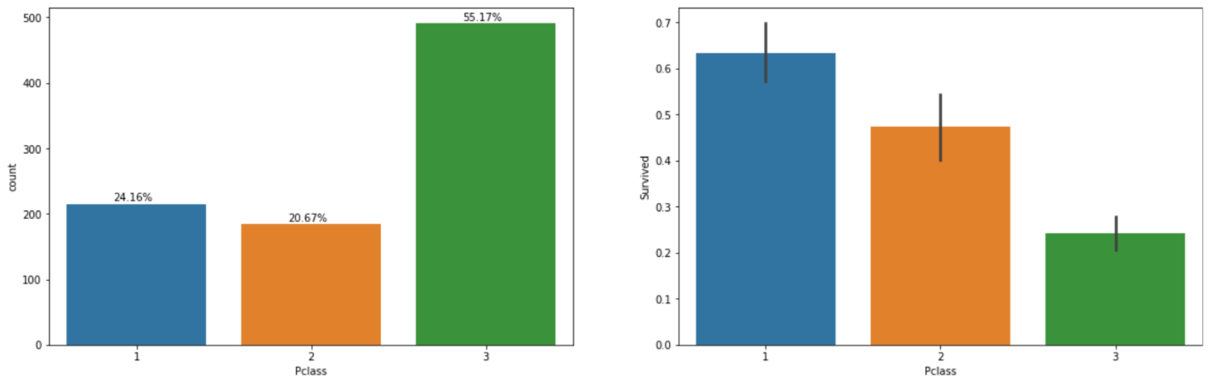
For the first analysis, I am going to compare the survival rate of passengers based on their gender. It will answer the question, who is most likely to survive, men or women (M/F)? The analysis is done using pie charts which describes the survival of each gender and survival as a whole.



From the above chart, it can be observed that more than half of the total population ie., nearly 62% of the passengers did not survive the disaster. With respect to gender, it can be seen that among the 62% survived, most of them are female than male. It can be seen, among men and women, just 19% of the men survived but almost 75% of the women survived.

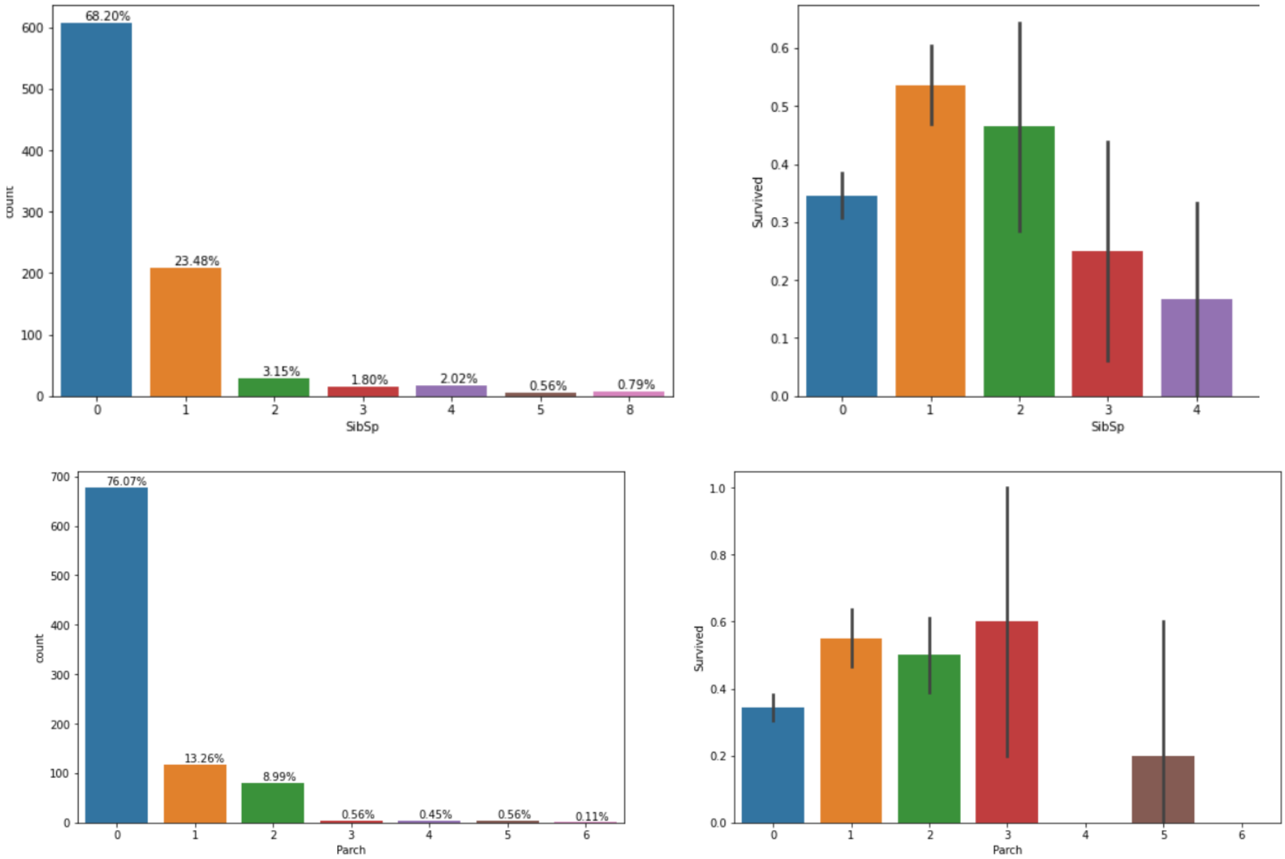
After this, I have plotted the graphs of categorical variables into two types – count plot and bar graph. This is find the relationship of the respective variables and the 'Survived' feature. I will explain them one after the other in the below section.

PCLASS



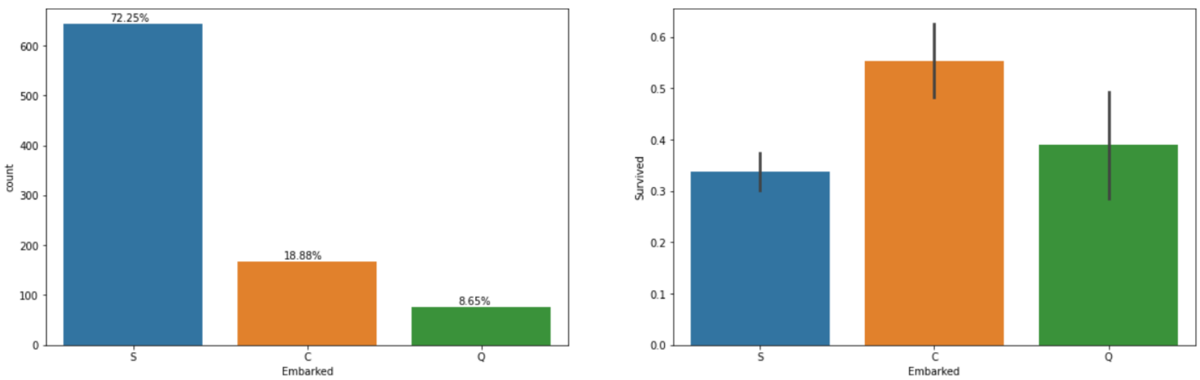
From the count plot, it is seen that more than 50% of the passengers belong to the 3rd class but the proportion of them who survived are extremely less (as seen in the bar plot). However, only 24% of the passengers belong to the 1st class and almost 60% of them survived the disaster.

SIBSP AND PARCH



From the plot of SibSp count plot, it can be seen that most of the passengers who travelled are alone. But, from the bar plot, the survival rate of single person is lesser compared to that of passenger with one spouse/ sibling. Similarly, looking into the count plot of 'Parch', passengers travelling with one accompany has more chances of survival comparatively than the others.

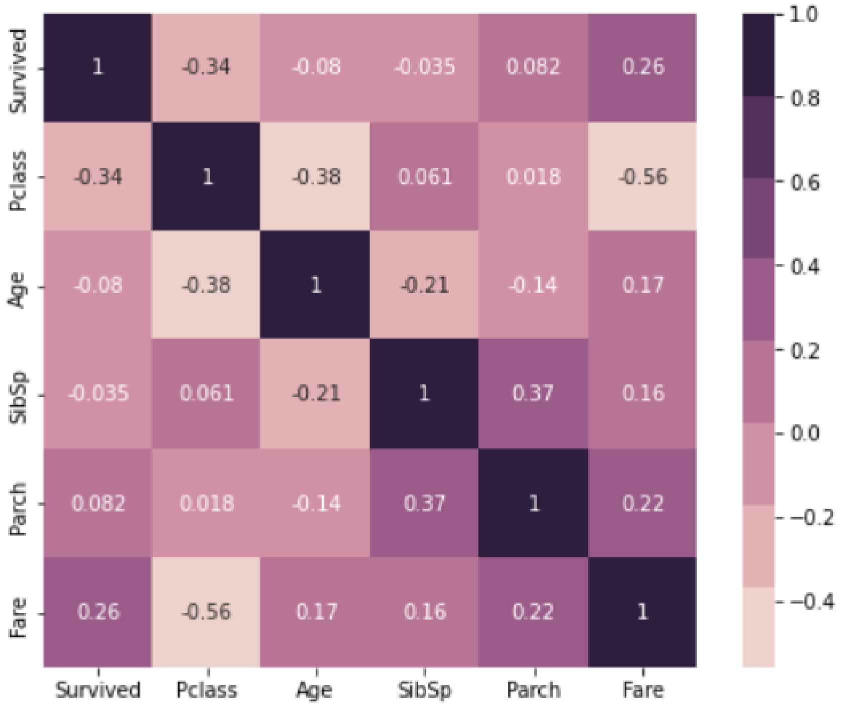
EMBARKED



With respective to the 'Embarked' feature, the passengers who boarded the ship in Southampton is nearly 75% of the population. However, only around 35% of them survived the impact. The people who survive the impact mostly belongs to Cherbourg.

Correlation of variables

Correlation is a measure that is used to determine the relationship between any two quantitative or categorical variables. Correlation analysis is used to compute the magnitude of the relationship between two random variables. The correlation of the variables is represented in the form of a correlation matrix where the rows and columns coverages to the value that represents the correlation between the two variables. For visualization purpose, we display the correlation matrix as heat-map. Heat-maps will have represented correlation of two discrete variables in a monochromatic scale.



7. MODELLING

For modelling purpose, I am using 4 different types of classification models

- Logistic Regression
- Support Vector Classifier
- Decision Tree Classifier
- Random Forest Classifier

After fitting the train and test data into these models, the accuracy is calculated by using the `score()` function and the best model is determined.

Initially, the data is split into train and test set using `train_test_split()` function in the ratio of 70/30. 70% of data belong to train set and 30% belong to test set.

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state=42)
```

LOGISITIC REGRESSION

Logistic regression is used when the response variable is a categorical variable. Here the predicted variable is ‘Survived’ which is a categorical variable with Y/N values. Here the logit function is used in the process of classification of the dependent variable with the independent variable. It is used when we need to find the likelihood of an event happening, here about the survival of the passengers. The implementation of Logistic Regression on the data is given below.

```
[280]  1 regr = LogisticRegression()

[ ]    1 regr.fit(X_train, y_train)

[ ]    1 regr.score(X_train, y_train)

[ ]    1 regr.score(X_test, y_test)
```

SUPPORT VECTOR CLASSIFIER

Support vector machine classifier (SVM) uses supervised learning to classify or predict the behaviour of groupings of data. Both the input and the required output data are provided by supervised learning systems, and they are labelled for classification. One of the main reasons to use SVM is that it can handle both classification and regression on linear and non-linear data. This method is efficient on high dimensional data. It has high speed and performance with a limited number of samples. The implementation of SVC on the data is given below.

```
1 svc = SVC(probability=True)

1 svc.fit(X_train, y_train)

1 svc.score(X_train, y_train)

1 svc.score(X_test, y_test)
```

DECISION TREE CLASSIFIER

The decision tree classifier creates the classification model by building a decision tree. It which is utilized for both classification and regression tasks. It has a tree-like hierarchy is composed of a root node, branches, internal nodes, and leaf nodes. Some of the reasons to use decision tree classifier is that

- It is easy to compute
- It is easier to deliver why one variable has a higher feature importance than the other
- It is easier to visualise which helps to explain the implementation of the model

The implementation of Decision Tree Classifier on the data is given below.

```
dt = DecisionTreeClassifier()
```

```
dt.fit(X_train, y_train)
```

```
dt.score(X_train, y_train)
```

```
dt.score(X_test, y_test)
```

RANDOM FOREST CLASSIFIER

Random forest classifier algorithm is used for both regression and classification. Each decision tree in the ensemble of the random forest method is built of a data sample taken from a training set with replacement, and the ensemble as a whole is composed of a collection of decision trees. A random forest generates accurate predictions that are simple to comprehend. Large datasets can be handled effectively. In comparison to the decision tree method, the random forest algorithm offers a higher level of accuracy in outcome prediction. The implementation of Random Forest Classifier on the data is given below.

```
rf = RandomForestClassifier()
```

```
rf.fit(X_train, y_train)
```

```
rf.score(X_train, y_train)
```

```
rf.score(X_test, y_test)
```

8. RESULT

The tabulation of results on test and train data is given separately below.

- Accuracy on train-data

	Name	Train Accuracy
2	DecisionTreeClassifier	89.6067
3	RandomForestClassifier	89.6067
0	LogisticRegression	85.5337
1	SVC	85.5337

- Accuracy on test-data

	Name	Test Accuracy
3	RandomForestClassifier	85.4749
1	SVC	84.3575
2	DecisionTreeClassifier	84.3575
0	LogisticRegression	83.7989

It can be seen that the Random Forest Algorithm works well on both train and test set of data with nearly 89% and 85% prediction accuracy respectively.

9. CONCLUSION

Numerous machine learning models should have an extended hyper-parameter tweaked in order to further enhance the final outcome. It may be improved much further by utilizing ensemble learning. This study work started with data exploration, which led to screening for missing data and discovering what attributes are crucial. Exploratory data analytics makes it possible to identify the properties of the dataset and the dependency connection. In order to determine how the dataset's features, relate to one another, EDA is used. Various graphical techniques are used to achieve this. Applying EDA, some inferences are made, and the information is gathered.

It has been noted that female survival rates are extremely high (about 74%) whereas male survival rates are extremely low. This truth can also be confirmed by extracting titles from the name column. Mr. has a roughly 16% survival chance, and Mrs. has a 79% survival rate. In order to determine a given passenger's family size, we combined the parch and sibsp columns. It has been found that the survival rate rises if the family has size of 0 and 3. The survival rate, on the other hand, tends to decline as family size increases above 3. Utilizing the exploratory data analytics method, feature engineering identifies the precise parameters that must be employed while designing the prediction and training model.

Machine learning methods assess the values of the passengers who survived. Numerous techniques, including Logistic Regression, Decision Tree, SVC, and Random Forest, are utilized to produce predictions in classification problems. With 86% accuracy, the Random Forest Classifier algorithm stood out as being the most effective of them.

10. REFERENCES

- https://www.researchgate.net/publication/351155499_Predicting_the_Likelihood_of_Survival_of_Titanic's_Passengers_by_Machine_Learning
- https://www.kaggle.com/competitions/titanic/data
- https://www.academia.edu/38724879/Predicting_the_Survivors_of_the_Titanic_Kaggle_Machine_Learning_From_Disaster
- https://machinelearningmastery.com/calculate-feature-importance-with-python/
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#:~:text=A%20random%20forest%20classifier.,accuracy%20and%20control%20over%2Dfitting.
- https://pandas.pydata.org/