A PRELIMENERY REPORT

ON

“Titanic survivor prediction”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,
PUNE IN HE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

GUIDED BY
Prof Bhosale.S.S

SUBMITTED BY

JADHAV TEJAS VINOD (72155650H)

DEPARTMENT OF COMPUTER ENGINEERING

HSBPVT’s FACULTY OF ENGINEERING, KASHTI

SAVITRIBAI PHULE PUNE UNIVERSITY

SUBMISSION: 2023-2024

CERTIFICATE

This is to certify that the mini project report on

“Titanic survivor prediction”

SUBMITTED BY,

This is to certify that **Mr. Tejas Vinod Jadhav** has successfully completed the mini project work entitled “**Titanic survivor prediction**” under my supervision, in the partial fulfillment of Bachelor in Engineering (Computer) of Savitribai Phule Pune University, Pune.

Guide (Bhosale S. S.)
Department of Computer Engineering

Head (Hirnawale S. B.)
Department of Computer Engineering

Place:

Kashti

Date:

ACKNOWLDGEMENT

The present world of competition there is a race of existence in which those are having will to comeforward succeed. Project is like a bridge between theoretical and practical working. First of all, I would like to thank the supreme power the Almighty God who is obviously the one has always guided me to work on the right path of life.

I am indebted to our project guide **Miss. Bhosale S.S**, Department of Computer Science of faculty of engineering, kashti. I feel it's a pleasure to be indebted to our guide for his valuable support, advice and encouragement and thankful for HOD to cooperate I think him for his superb and constant guidance.

Mr. Tejas Vinod Jadhav

INDEX

INTRODUCTION

Using the well-known Titanic dataset as a starting point, I would create a machine learning model. This provides a prognosis of the Titanic's likelihood of surviving, taking into consideration a number of variables like economic standing (class), sex, age, etc.

This is taken into account, and many features are compared and found to have relationships in order to estimate whether a passenger would survive on the Titanic. because it is a component of "Titanic: Machine Learning from Disaster." In this exercise, we must determine whether a Titanic passenger would have survived or not.

The RMS Titanic was the largest ship a float at the time it entered service and was the second of three Olympic-class ocean liners operated by the White Star Line. The Titanic was built by the Harland and Wolff shipyard in Belfast. Thomas Andrews, her architect, died in the disaster.

The RMS Titanic was a British passenger liner that sank in the North Atlantic Ocean in the early morning hours of 15 April 1912, after it collided with an iceberg during its maiden voyage from

PROBLEM DEFINITION

Titanic Survival Prediction Using Machine Learning:

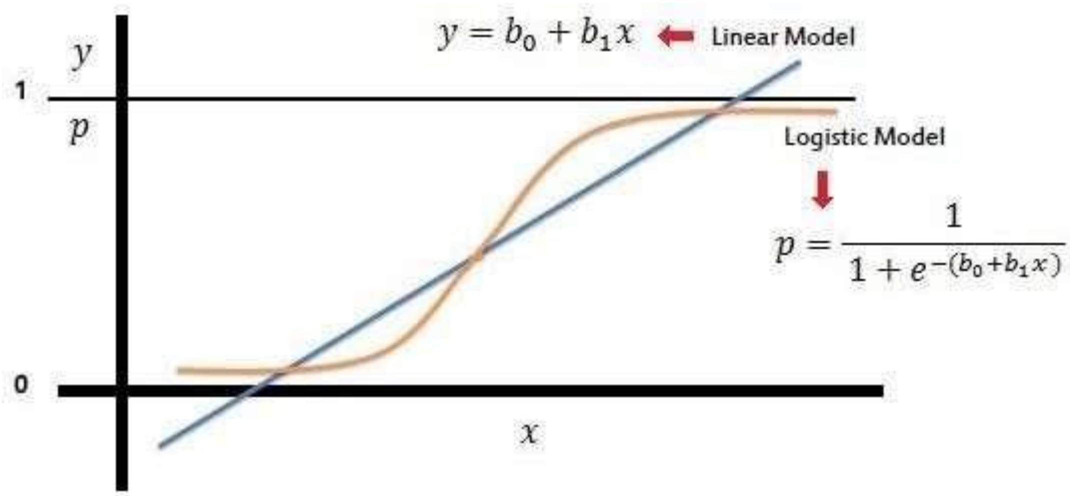
Build a machine learning model that predicts the type of people who survived the titanicshipwreck using passenger data (i.e. name, age, gender, socioeconomic class, etc).

SYSTEM ARCHITECTURE

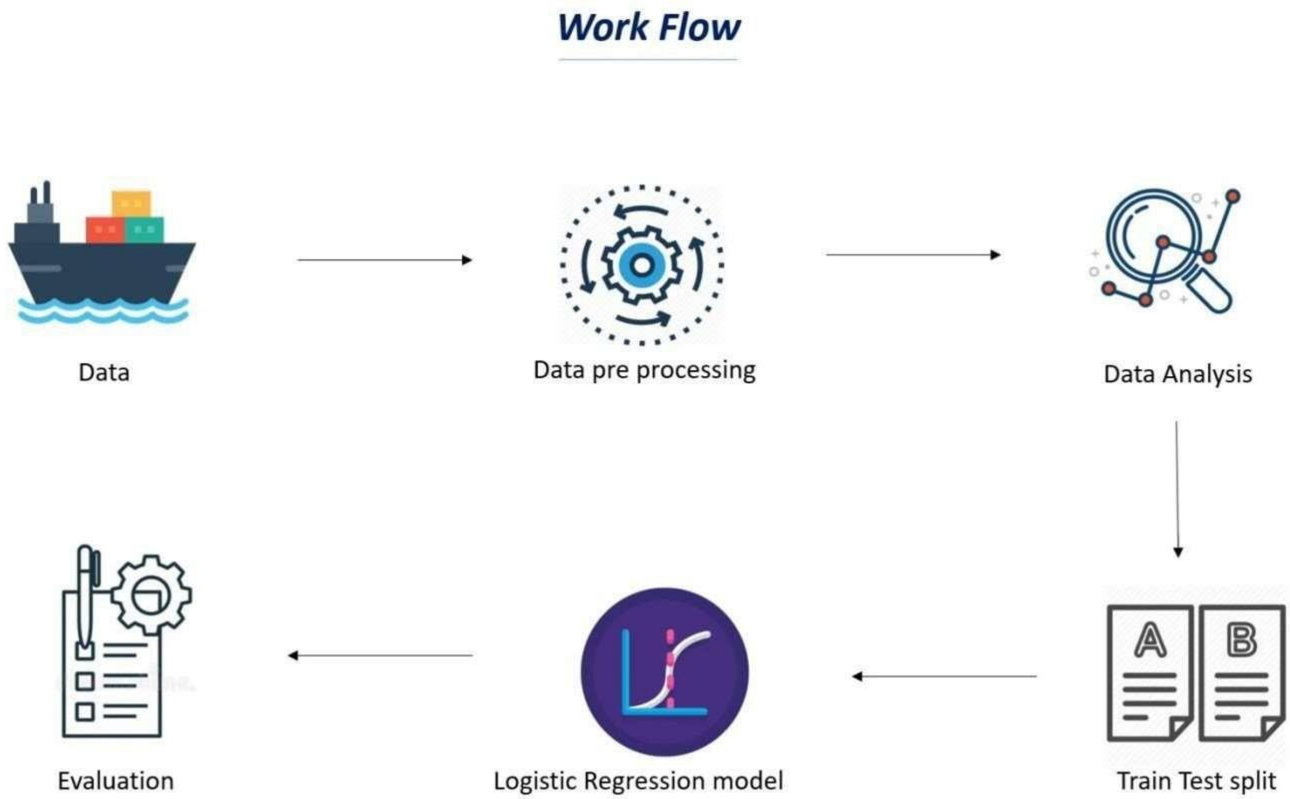
I will be understanding, how to analyze and predict, whether a person, who had boarded the RMSTitanic has a chance of survival or not, using Machine Learning's Logistic Regression model.

Brief description about Logistic Regression:

A simple yet crisp description of Logistic Description would be, "it is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes." The graphof logistic regression is as shown below:



For better understanding, let's split the task into smaller parts and depict them in a work flow as shownbelow :



As I now know what I have to do, to accomplish this task, I shall begin with the very first and the most important thing needed in machine learning, a **Dataset.**

What is a dataset:

A data set, as the name suggests, is a collection of data. In Machine Learning projects, we need atraining data set. It is the actual data set used to train the model for performing various actions.

DESCRIPTION

Data Set Column Descriptions:

- **pclass:** Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
- **survived:** Survival (0 = No; 1 = Yes)
- **name:** Name
- **sex:** Sex
- **age:** Age
- **sibsp:** Number of siblings/spouses aboard
- **parch:** Number of parents/children aboard
- **fare:** Passenger fare (British pound)
- **embarked:** Port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)
- **adult_male:** A male 18 or older (0 = No, 1=Yes)
- **deck:** Deck of the ship
- **who:** man (18+), woman (18+), child (<18)
- **alive:** Yes, no
- **embarked_town:** Port of embarkation (Cherbourg, Queenstown, Southampton)
- **class:** Passenger class (1st; 2nd; 3rd)
- **alone:** 1= alone, 0= not alone (you have at least 1 sibling, spouse, parent or child on board)
  age:

Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

Sibsp:

The dataset defines family relations in this way:

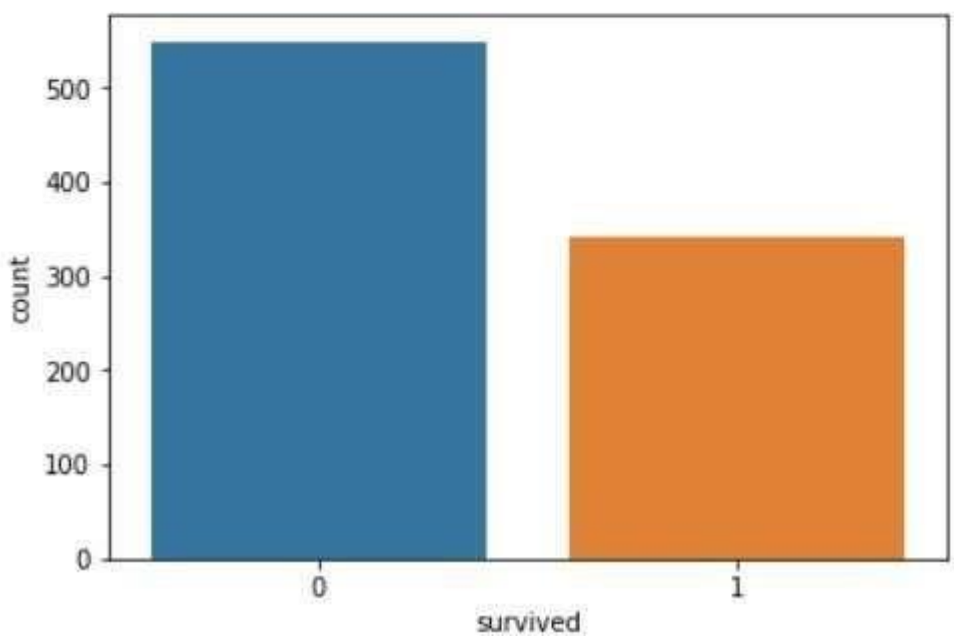
- Sibling= brother, sister, stepbrother, stepsister
- Spouse= husband, wife (mistresses and fiancés were ignored)

Parch:

The dataset defines family relations in this way:

- Parent= mother, father
- Child= daughter, son, stepdaughter, stepson

  Some children traveled only with a nanny, therefore parch=0 for them.



Visualize the count of survivors for the columns who, sex, pclass, sibsp, parch, and embarked.

- From the charts below, we can see that a man (a male 18 or older) is not likely to survive from thechart who.

- Females are most likely to survive from the chart sex.
- Third class is most likely to not survive by chart pclass.
- If you have 0 siblings or spouses on board, you are not likely to survive according to chart sibsp.
- If you have 0 parents or children on board, you are not likely to survive according to the parchchart.
- If you embarked from Southampton (S), you are not likely to survive according to the embarkedchart.

IMPLEMENTATION

Start Programming: Now import the packages /libraries to make it easier to write the program.

Build a machine learning model that predicts the type of people who survived the titanic shipwreck using passenger data age name gender socio-economics class etc

In [44]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report
## The matplotlib and seaborn library for result visualization and analysis
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style='darkgrid')
```

In [45]:
```
train = pd.read_csv('C:/Users/DHAWADE/Desktop/BE/ML/train.csv')
test = pd.read_csv('C:/Users/DHAWADE/Desktop/BE/ML/test.csv')
```

In [134]:
```
train.shape, test.shape
```

Out[134]:
```
((891, 12), (418, 11))
```

In [135]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          891 non-null    int32
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        891 non-null    object
 11  Embarked     891 non-null    object
dtypes: float64(1), int32(1), int64(5), object(5)
memory usage: 80.2+ KB
```

In [136]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          418 non-null    int32
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         418 non-null    float64
 9   Cabin        418 non-null    object
 10  Embarked     418 non-null    object
dtypes: float64(1), int32(1), int64(4), object(5)
memory usage: 34.4+ KB
```

In [137]:
```
train.head()
```
Out[137]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	X	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38	1	0	PC 17599	71.2833	C	C
2	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250	X	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1000	C	S
4	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500	X	S

In [138]:
```
test.head()
```
Out[138]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	22	0	0	330911	7.8292	X	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	38	1	0	363272	7.0000	X	S
2	894	2	Myles, Mr. Thomas Francis	male	26	0	0	240276	9.6875	X	Q
3	895	3	Wirz, Mr. Albert	male	35	0	0	315154	8.6625	X	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	35	1	1	3101298	12.2875	X	S

In [139]:
```
train.describe()
```
Out[139]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.388328	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.525408	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.000000	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	37.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

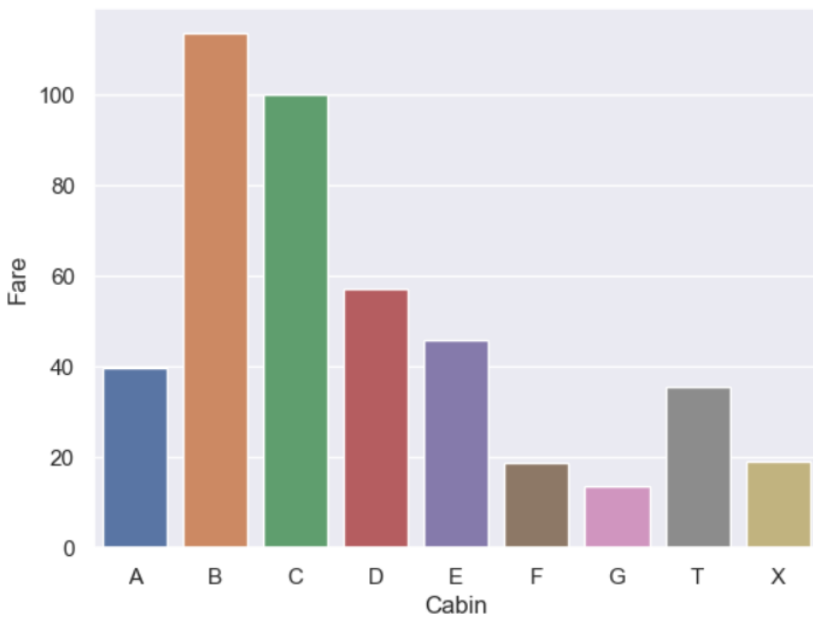
In [140]:
```
test.describe()
```
Out[140]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000
mean	1100.500000	2.265550	28.519139	0.447368	0.392344	35.627188
std	120.810458	0.841838	13.157991	0.896760	0.981429	55.840500
min	892.000000	1.000000	0.000000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	20.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	36.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	71.000000	8.000000	9.000000	512.329200

In [141]:
```
train.nunique()
```
Out[141]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             71
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin            9
Embarked         3
dtype: int64
```

In [142]:
```
test.nunique()
```
Out[142]:
```
PassengerId    418
Pclass           3
Name           418
Sex              2
Age             62
SibSp            7
Parch            8
Ticket         363
Fare           170
Cabin            8
Embarked         3
dtype: int64
```

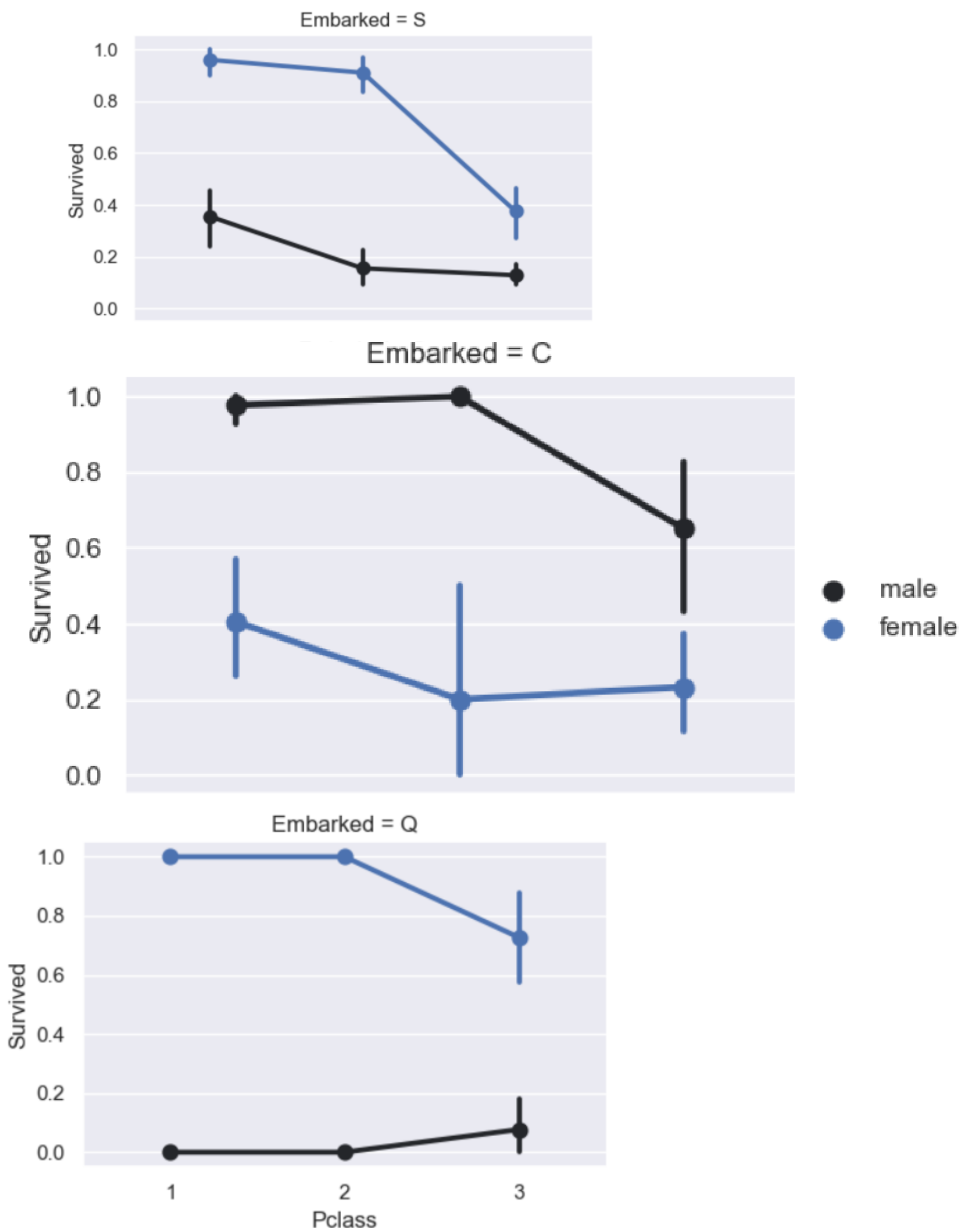
In [143]:
```
train['Cabin'].fillna(value='X', inplace=True)
train['Cabin'] = train['Cabin'].str[0]
df_tr = train[['Cabin', 'Fare']].groupby('Cabin').mean().reset_index()
a = sns.barplot(x=df_tr['Cabin'], y=df_tr['Fare'])
```



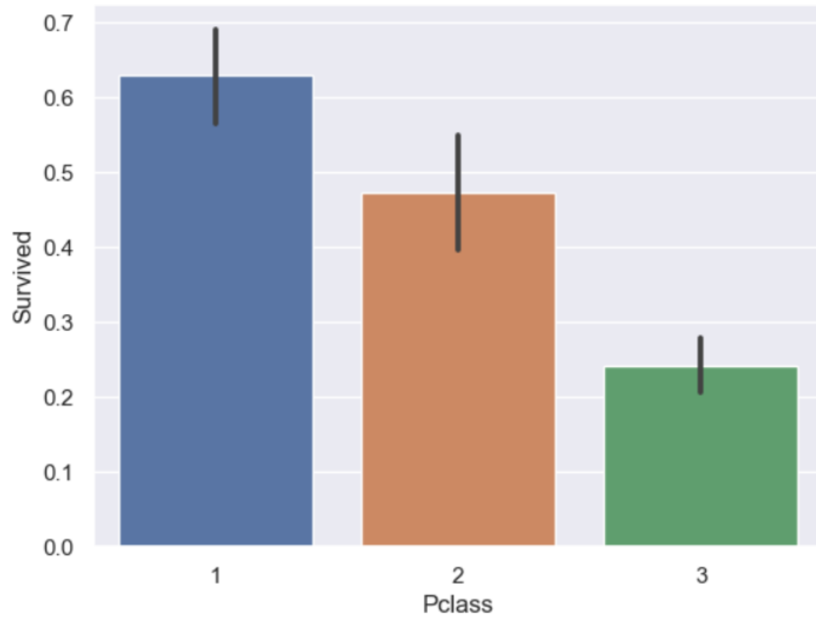
In [149]:
```
FacetGrid = sns.FacetGrid(train, row='Embarked',  aspect=1.6)
FacetGrid.map(sns.pointplot, 'Pclass', 'Survived', 'Sex', palette=None,  order=None, hue_order=None )
FacetGrid.add_legend()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

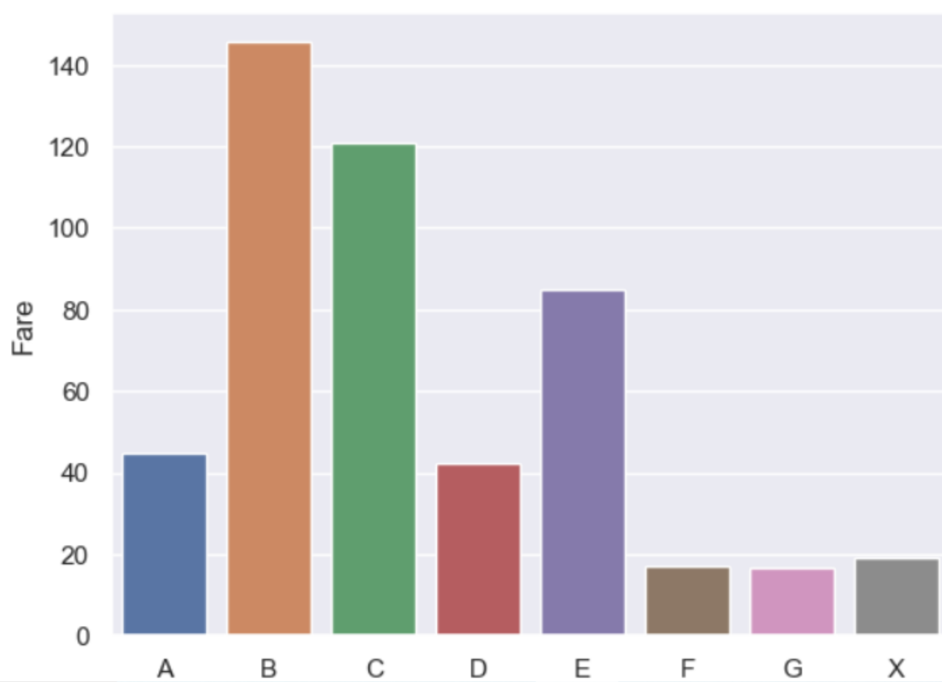
Out[149]: `<seaborn.axisgrid.FacetGrid at 0x1efda6f2a50>`



In [150]:
```
sns.barplot(x='Pclass', y='Survived', data=train)
```
Out[150]: <Axes: xlabel='Pclass', ylabel='Survived'>



In [151]:
```
test['Fare'].fillna(value=test.Fare.mean(), inplace=True)
test['Cabin'].fillna(value='X', inplace=True)
test['Cabin'] = test['Cabin'].str[0]
df_te = test[['Cabin', 'Fare']].groupby('Cabin').mean().reset_index()
a = sns.barplot(x=df_te['Cabin'], y=df_te['Fare'])
```



In [152]:
```
train['Embarked'] = train.Embarked.fillna(train.Embarked.dropna().max())
```

In [153]:
```
guess_ages = np.zeros((2,3))
combine = [train , test]
# Converting Sex categories (male and female) to 0 and 1:
for dataset in combine:guess_ages
```

In [154]:
```
data = [train, test]

for dataset in data:
    mean = train["Age"].mean()
    std = test["Age"].std()
    is_null = dataset["Age"].isnull().sum()
    # compute random numbers between the mean, std and is_null
    rand_age = np.random.randint(mean - std, mean + std, size = is_null)
    # fill NaN values in Age column with random values generated
    age_slice = dataset["Age"].copy()
    age_slice[np.isnan(age_slice)] = rand_age
    dataset["Age"] = age_slice
    dataset["Age"] = train["Age"].astype(int)
train["Age"].isnull().sum()
```

Out[154]:
```
0
```

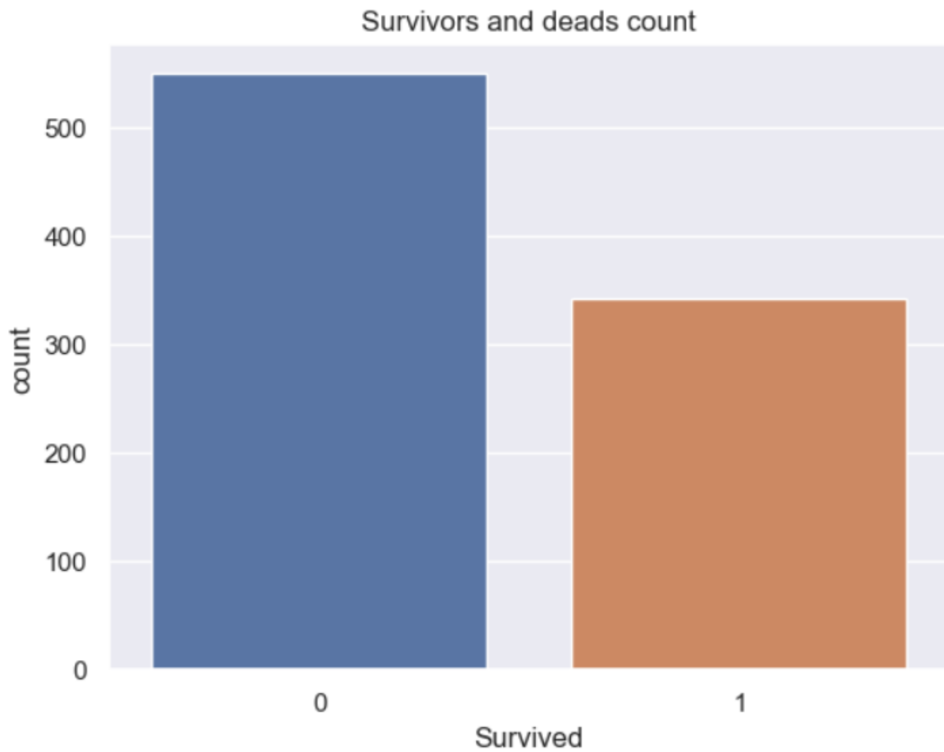
In [155]:
```
train.isna().sum()
```

Out[155]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [156]:
```
test.isna().sum()
```

Out[156]:
```
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
```

```
In [157]: g = sns.countplot(x=train['Survived']).set_title('Survivors and deads count')
```



```
results = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic
Regression',
              'Random Forest', 'Naive Bayes', 'Perceptron',
              'Stochastic Gradient Decent',
              'Decision Tree'],
    'Score': [acc_linear_svc, acc_knn, acc_log,
              acc_random_forest, acc_gaussian, acc_perceptron,
              acc_sgd, acc_decision_tree]})
result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
result_df.head(9)
```

	Model
Score	
92.82	Random Forest
92.82	Decision Tree
87.32	KNN
81.14	Logistic Regression
80.81	Support Vector Machines
80.70	Perceptron
77.10	Naive Bayes
76.99	Stochastic Gradient Decent

```
train_df.head(10)
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	relatives	not_alone	Deck	Title
0	0	3	0	2	1	0	7	0	1	0	8	1
1	1	1	1	5	1	0	71	1	1	0	3	3
2	1	3	1	3	0	0	7	0	0	1	8	2
3	1	1	1	5	1	0	53	0	1	0	3	3
4	0	3	0	5	0	0	8	0	0	1	8	1
5	0	3	0	4	0	0	8	2	0	1	8	1
6	0	1	0	6	0	0	51	0	0	1	5	1
7	0	3	0	0	3	1	21	0	4	0	8	4
8	1	3	1	3	0	2	11	0	2	0	8	3
9	1	2	1	1	1	0	30	1	1	0	8	3

EXPECTED OUTPUT

Before:

	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

After:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	relatives	not_alone	Deck	Title	Age_Class	Fare_Per_Persc
0	0	3	0	2	1	0	0	0	1	0	8	1	6	0
1	1	1	1	5	1	0	3	1	1	0	3	3	5	1
2	1	3	1	3	0	0	0	0	0	1	8	2	9	0
3	1	1	1	5	1	0	3	0	1	0	3	3	5	1
4	0	3	0	5	0	0	1	0	0	1	8	1	15	1

CONCLUSION:

We started with the data exploration where we got a feeling for the dataset, checked about missing data and learned which features are important. During this process we used seaborne and matplotlib to do the visualizations. During the data preprocessing part, we computed missing values, converted features into numeric ones, grouped values into categories and created a few new features.

REFERENCES:

1.Michalski R S,et al.Machine Learning:Challenges of the eighties.Machine Learning,1986,99-102.

2.Analyzing Titanic disaster using machine learning algorithms-Computing, Communication and Automation (ICCCA), 2012 International Conference on 21 December 2017, IEEE.

3.Eric Lam, Chongxuan Tang, "Titanic Machine Learning From Disaster", LamTang-Titanic Machine Learning From Disaster, 2012.