Logistic Regression on Titanic DataSet

Business Question: What factor made people more likely to survive the sinking of the Titanic ship?
Dataset Link: [Titanic data - Titanic - Machine Learning from Disaster | Kaggle]

First thing I would do is to explore the titanic dataset and explore the people, both those who survived and those who did not. I would use Logistic Regr to predict this

Import and Read File

```
import numpy as np # linear algebra
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math


import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

titanic_data=pd.read_csv("/kaggle/input/titanic-dataset/Titanic-Dataset.csv")
```

```
/kaggle/input/titanic-dataset/Titanic-Dataset.csv
```

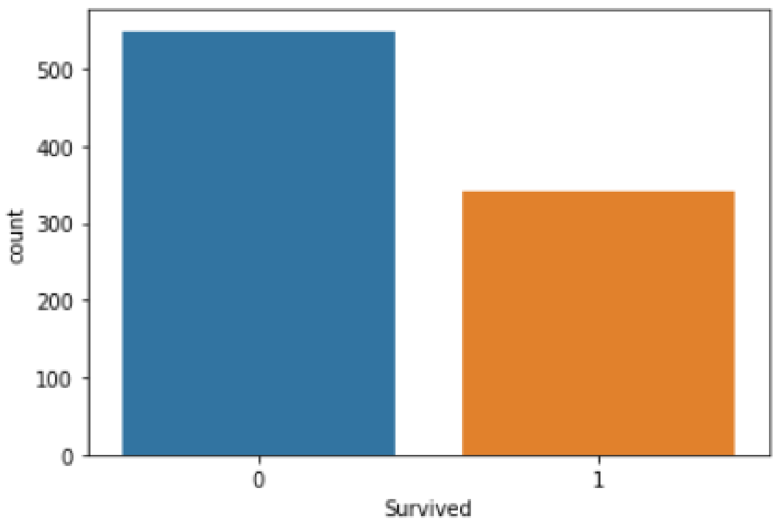
```
titanic_data.head(10)
```

	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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

Analyze Data/ Data Exploration

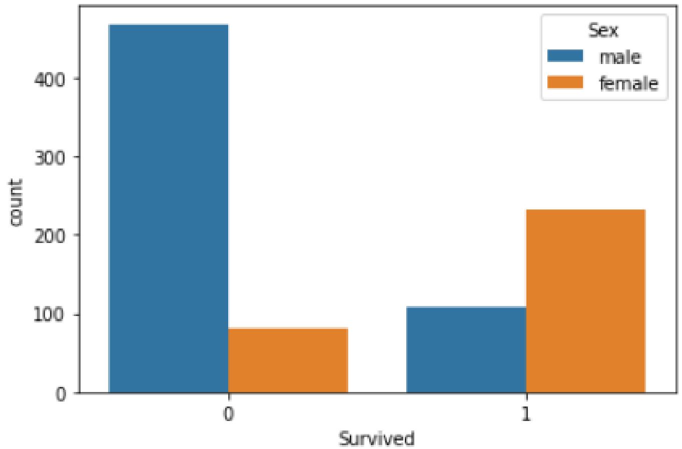
```
sns.countplot(x="Survived", data=titanic_data)
```

<AxesSubplot:xlabel='Survived', ylabel='count'>



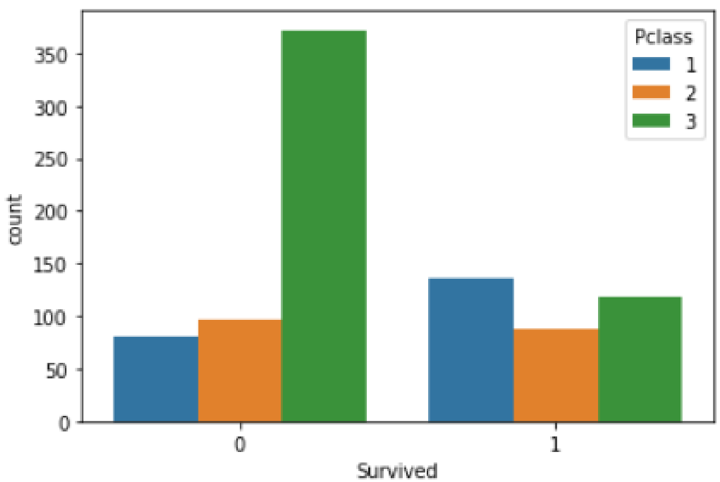
```
sns.countplot(x="Survived", hue="Sex", data=titanic_data)
```

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



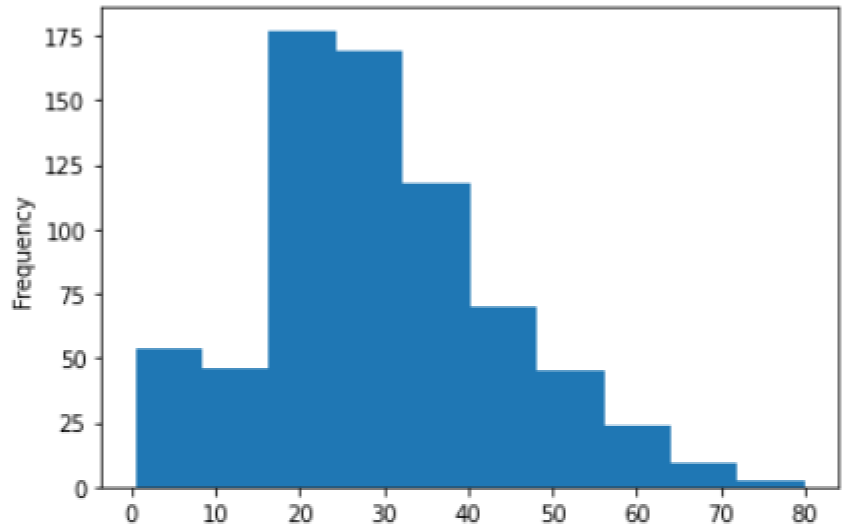

```
sns.countplot(x="Survived", hue="Pclass", data=titanic_data)
```

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



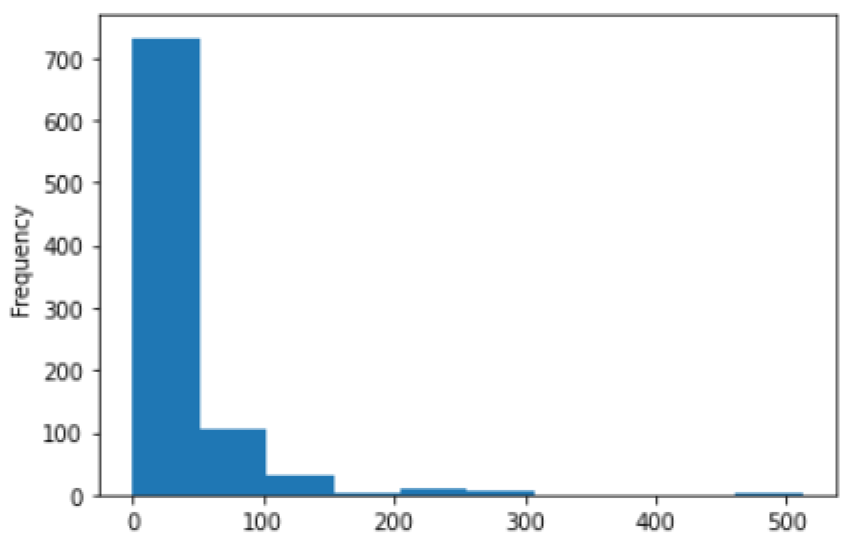
```
titanic_data["Age"].plot.hist()
```

<AxesSubplot:ylabel='Frequency'>




```
titanic_data["Fare"].plot.hist()
```

<AxesSubplot:ylabel='Frequency'>

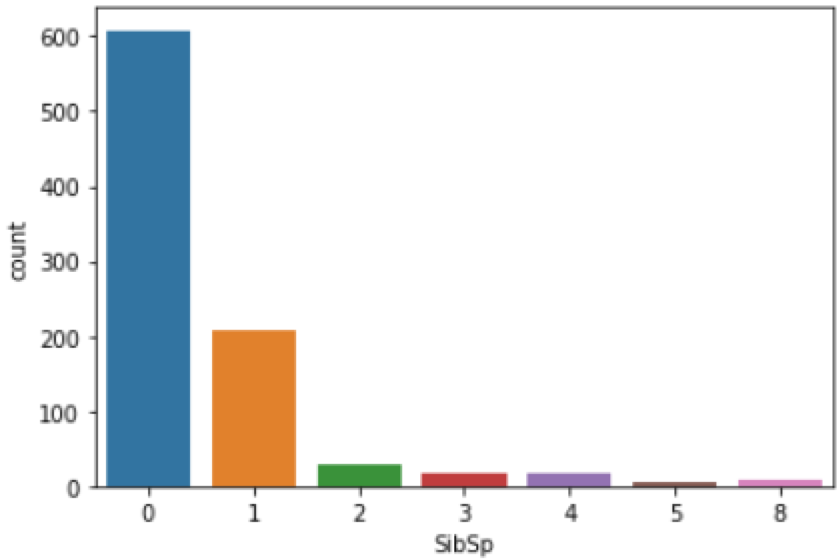


```python
titanic_data.info()
```

```
<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          714 non-null    float64
 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        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
sns.countplot(x="SibSp", data=titanic_data)
```

```
<AxesSubplot:xlabel='SibSp', ylabel='count'>
```



Checking for Null Values

```
titanic_data.isnull()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

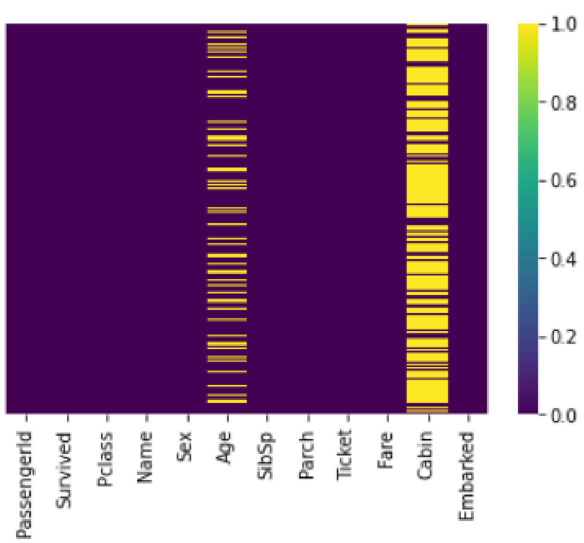
```
titanic_data.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Created a visualization to see how frequent to null values are

```
sns.heatmap(titanic_data.isnull(), yticklabels=False, cmap='viridis')
```

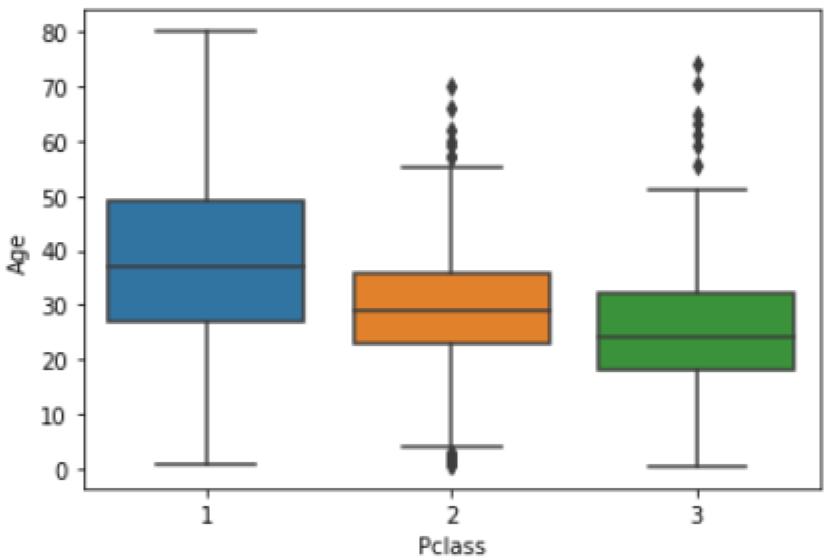
```
<AxesSubplot:>
```



Next, Look at the Age range in Pclass

```
sns.boxplot(x="Pclass", y="Age", data= titanic_data)
```

```
<AxesSubplot:xlabel='Pclass', ylabel='Age'>
```



Drop Null Values

```
titanic_data.head()
```

	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

```python
titanic_data.drop("Cabin", axis=1, inplace=True)
```

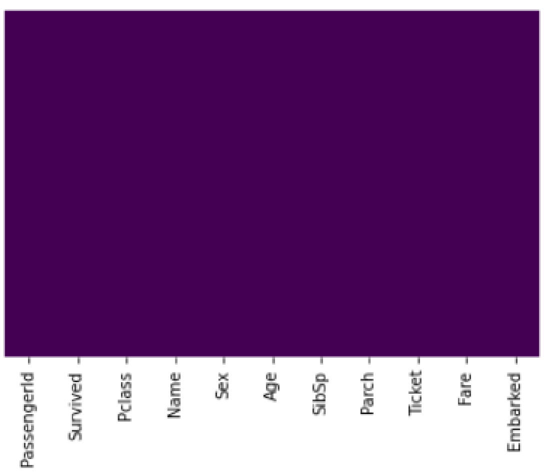
```python
titanic_data.head()
```

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

Check to confirm that there is no Null Values

```python
sns.heatmap(titanic_data.isnull(), yticklabels=False, cbar=False, cmap='viridis')
```

```
<AxesSubplot:>
```




```python
titanic_data.isnull().sum()
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

Data Wrangling - Cleaning Unwanted Values

```
titanic_data.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C

Here i am converting the Sex column to numerical data using "get_dummies" because my data needs to be numeric to do Regression.

```
pd.get_dummies(titanic_data['Sex'])
```

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1
...	...	...
885	1	0
886	0	1
887	1	0
889	0	1
890	0	1

Also, since the columns are not in numerical function 0 and 1. I can drop one column since I know that if the value of male is 1 then it's a male, if the value of male is 0 then it's a female. Because one column is enough to tell if it's a male or female.

```
sex =pd.get_dummies(titanic_data['Sex'],drop_first=True)
sex.head(5)
```

	male
0	1
1	0
2	0
3	0
4	1

I repeated the same for the Embarked column

```
embark=pd.get_dummies(titanic_data["Embarked"],drop_first=True)
embark.head(5)
```

	Q	S
0	0	1
1	0	0
2	0	1
3	0	1
4	0	1

I repeated the same for Pclass column. I only got rid of 1 colum since we have 3 columns

```
Pclass=pd.get_dummies(titanic_data["Pclass"],drop_first=True)
Pclass.head(5)
```

	2	3
0	0	1
1	0	0
2	0	1
3	0	0
4	0	1

Next i used the concat to concatenate the following column, sex, embark and Pclass

```
titanic_data=pd.concat([titanic_data, sex, embark, Pclass], axis=1)
titanic_data.head()
```

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

Then i drop the unwanted Columns

```
titanic_data.drop(['Sex', 'Embarked','PassengerId','Name', 'Ticket','Pclass'], axis=1, inplace=True)
```

```
titanic_data.head()
```

	Survived	Age	SibSp	Parch	Fare	male	Q	S	2	3
0	0	22.0	1	0	7.2500	1	0	1	0	1
1	1	38.0	1	0	71.2833	0	0	0	0	0
2	1	26.0	0	0	7.9250	0	0	1	0	1
3	1	35.0	1	0	53.1000	0	0	1	0	0
4	0	35.0	0	0	8.0500	1	0	1	0	1

Train & Test Data

Y is the Column i want to predict to why passenger survived or Not

```
X= titanic_data.drop("Survived", axis=1) # x is the independent variable
y= titanic_data["Survived"]
```

Import Logistic Regression and Train test Split

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

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

Create an instance Model for logistic Regression and fit to X_train and y_train

```
logmodel=LogisticRegression()
```

+ Code + Markdown

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

Now i need to make prediction, Create Variable and pass my model “Logmodel” to predict the value X_test

```
pred = logmodel.predict(X_test)
```

I used the classification report to show the precision, recall, F1 Score, and support of your trained classification model.

```
from sklearn.metrics import classification_report
```

```
classification_report(y_test,pred)
```

```
'              precision    recall  f1-score   support\n\n           0       0.81      0.87      0.84       132\n           1       0.76      0.67      0.71        82\n\n    accuracy                           0.79
214\n   macro avg       0.79      0.77      0.78       214\nweighted avg       0.79      0.79      0.79       214\n'
```

Lastly check for Accuracy score

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,pred)
```

```
0.794392523364486
```

The score here is 0.79. This means that 79% survived and 21% died.