Titanic project

The objective of this project is develop a predicive model that classifies passengers on the Titanic as either survivors or non-survivors based on various features.

Importing necessary libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

Import data

In [3]:
```
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
```

Part 1: Data Understanding

In [4]:
```
train.head(2)
```

Out[4]:

	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

In [5]:
```
train.columns
```

Out[5]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [6]:
```
train.dtypes
```

Out[6]:
```
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
```

Part 2: Data Cleaning

Droping columns that clearly doesn't give any useful information

In [7]:
```
train = train.drop(columns = ["Name", "Ticket", "Cabin", "Embarked","PassengerId"])
```

Getting rid of NaN values in dataset

In [22]:
```

```

Out[22]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [8]:
```
train.isna().sum()
```

Out[8]:
```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
dtype: int64
```

In [9]:
```
train.shape
```

Out[9]:
```
(891, 7)
```

In [10]:
```
train.dropna(inplace = True)
```

Preparation for ploting relationship graph

In [11]:
```
train.Sex = train.Sex.replace({"female" : 1 , "male" : 0})
```

In [12]:
```
train.head(5)
```

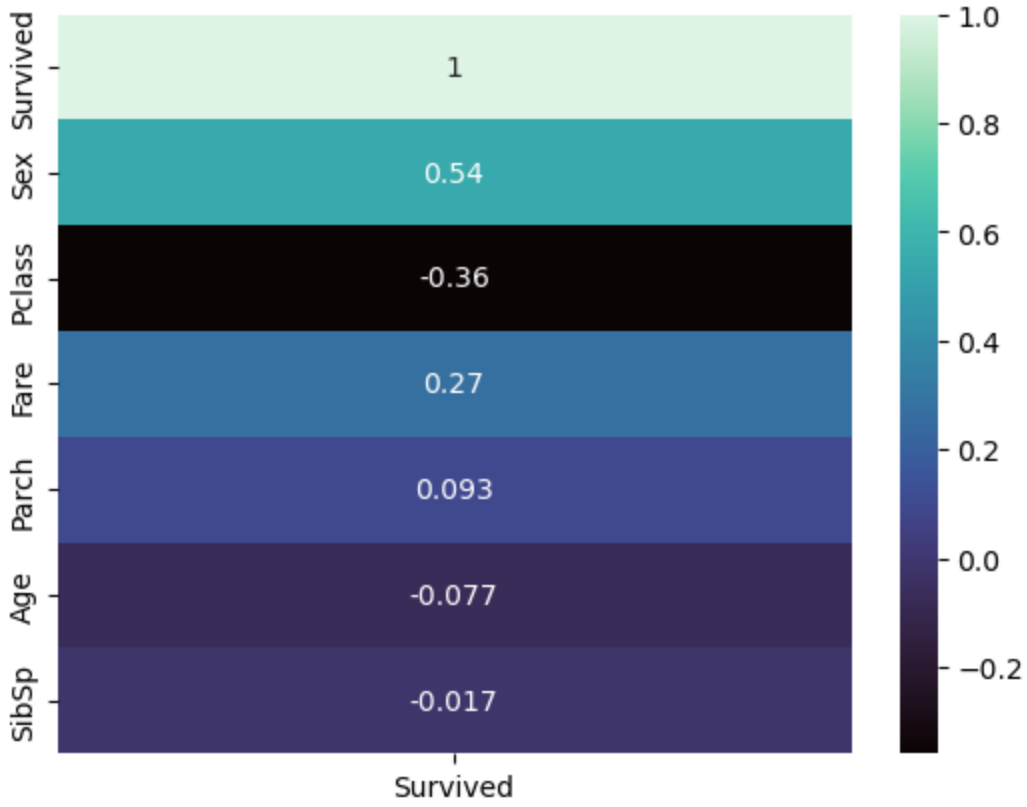
Out[12]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare
0	0	3	0	22.0	1	0	7.2500
1	1	1	1	38.0	1	0	71.2833
2	1	3	1	26.0	0	0	7.9250
3	1	1	1	35.0	1	0	53.1000
4	0	3	0	35.0	0	0	8.0500

Part 3: Data Visualization

Plotting relationship graph

In [13]:
```
train_corr = train.corr()
fig = sns.heatmap(train.corr()[["Survived"]].sort_values(by = ["Survived"], ascending =
plt.show()
```

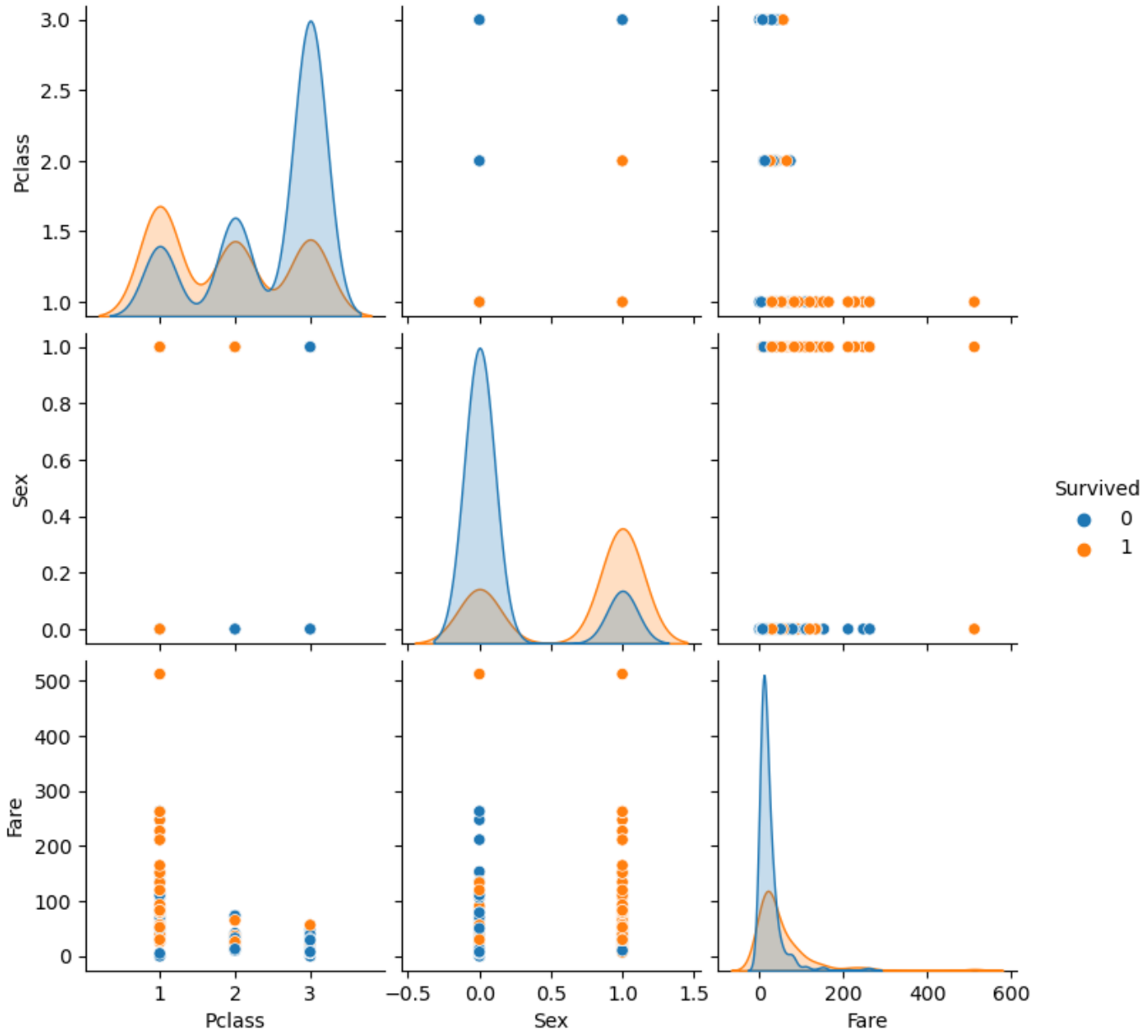


Deleting parameters which have poor correlation with survival rate

In [14]:
```
train.drop(columns = ["Parch", "Age" ,"SibSp"],inplace = True)
```

Creating plots to decide which model is the best for our data

In [15]:
```
warnings.filterwarnings('ignore')
figure = sns.pairplot(train, hue = "Survived")
plt.show()
```



Part 4 : Model Building

We will use classification model ,because data we need to predict boolean variable

From the graph is clear that Logistic Regression model is the best, because in the graphs overlapping is minimal

In [16]:
```
x_train , x_test , y_train , y_test = train_test_split(train[["Pclass","Sex","Fare"]], t
```

Logistic Regression

In [17]:
```
log_reg = LogisticRegression(random_state = 69).fit(x_train,y_train)
log_reg.score(x_test,y_test)
```

Out[17]:
```
0.813953488372093
```

In conclusion, the Titanic project successfully explored, analyzed, and modeled the dataset to predict passenger survival outcomes during the sinking of the Titanic.