

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

import warnings
warnings.filterwarnings("ignore")

titanic_data = pd.read_csv('train.csv')
titanic_test = pd.read_csv('test.csv')
titanic_data.head()

```

```

    PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3

```

```

                                     Name    Sex  Age
SibSp  \
0                                     Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                                     Heikkinen, Miss. Laina  female  26.0
0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0
1
4                                     Allen, Mr. William Henry    male  35.0
0

```

```

    Parch    Ticket    Fare Cabin Embarked
0      0  A/5 21171    7.2500   NaN        S
1      0    PC 17599   71.2833   C85        C
2      0  STON/O2. 3101282    7.9250   NaN        S
3      0    113803   53.1000  C123        S
4      0    373450    8.0500   NaN        S

```

```
titanic_data.shape
```

```
(891, 12)
```

```
titanic_data.describe()
```

```

count    PassengerId  Survived  Pclass    Age    SibSp  \
mean      446.000000    0.383838    2.308642   29.699118    0.523008

```

std	257.353842	0.486592	0.836071	14.526497	1.102743
min	1.000000	0.000000	1.000000	0.420000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000
50%	446.000000	0.000000	3.000000	28.000000	0.000000
75%	668.500000	1.000000	3.000000	38.000000	1.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000

	Parch	Fare
count	891.000000	891.000000
mean	0.381594	32.204208
std	0.806057	49.693429
min	0.000000	0.000000
25%	0.000000	7.910400
50%	0.000000	14.454200
75%	0.000000	31.000000
max	6.000000	512.329200

```
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
```

```
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
```

```
titanic_data = titanic_data.drop(columns='Cabin', axis = 1)
```

```
titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace= True)
```

```
print(titanic_data['Embarked'].mode()[0])
```

S

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

```
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
```

```
titanic_data.shape
```

```
(891, 11)
```

```
titanic_data.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp
Parch \					
PassengerId	1.000000	-0.005007	-0.035144	0.033207	-0.057527
0.001652					
Survived	-0.005007	1.000000	-0.338481	-0.069809	-0.035322
0.081629					
Pclass	-0.035144	-0.338481	1.000000	-0.331339	0.083081
0.018443					
Age	0.033207	-0.069809	-0.331339	1.000000	-0.232625
0.179191					
SibSp	-0.057527	-0.035322	0.083081	-0.232625	1.000000
0.414838					
Parch	-0.001652	0.081629	0.018443	-0.179191	0.414838
1.000000					
Fare	0.012658	0.257307	-0.549500	0.091566	0.159651
0.216225					

```

            Fare
PassengerId  0.012658
Survived     0.257307
Pclass       -0.549500
Age          0.091566
SibSp        0.159651
Parch        0.216225
Fare         1.000000

```

```
titanic_data['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
titanic_data['Sex'].value_counts()
```

```
male    577
female  314
Name: Sex, dtype: int64
```

```
titanic_data.replace({'Sex':{'male':0,'female':1}}, inplace = True)
```

```
titanic_data['Embarked'].unique()
```

```
array(['S', 'C', 'Q'], dtype=object)
```

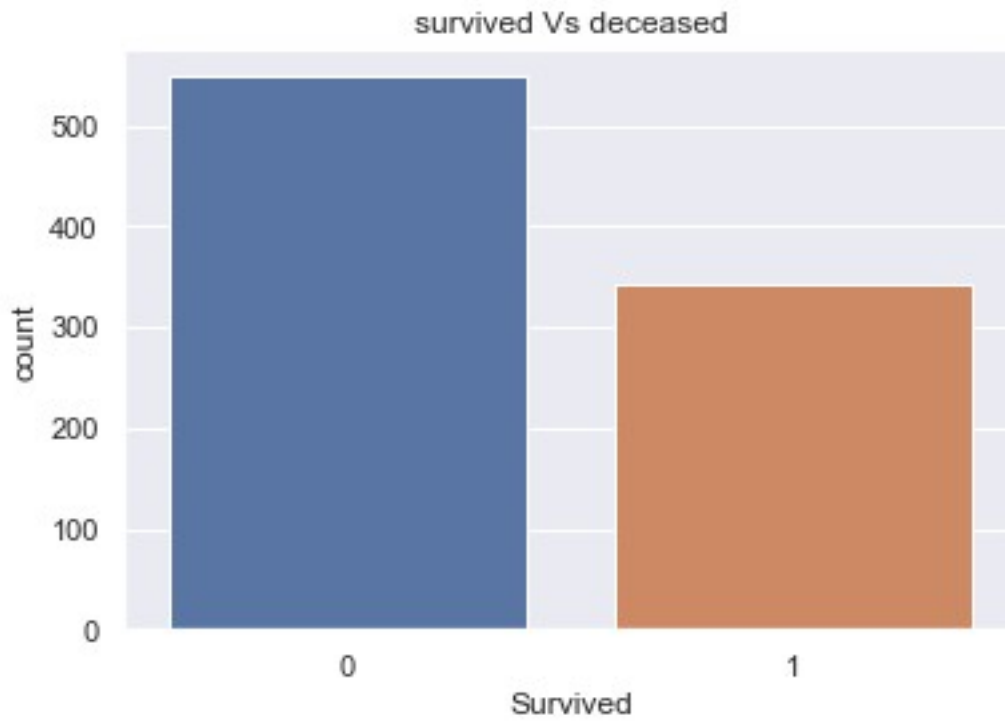
```
titanic_data.replace({'Embarked':{'S':0,'C':1, 'Q':2}}, inplace =
True)
```

```
titanic_data['Parch'].unique()
```

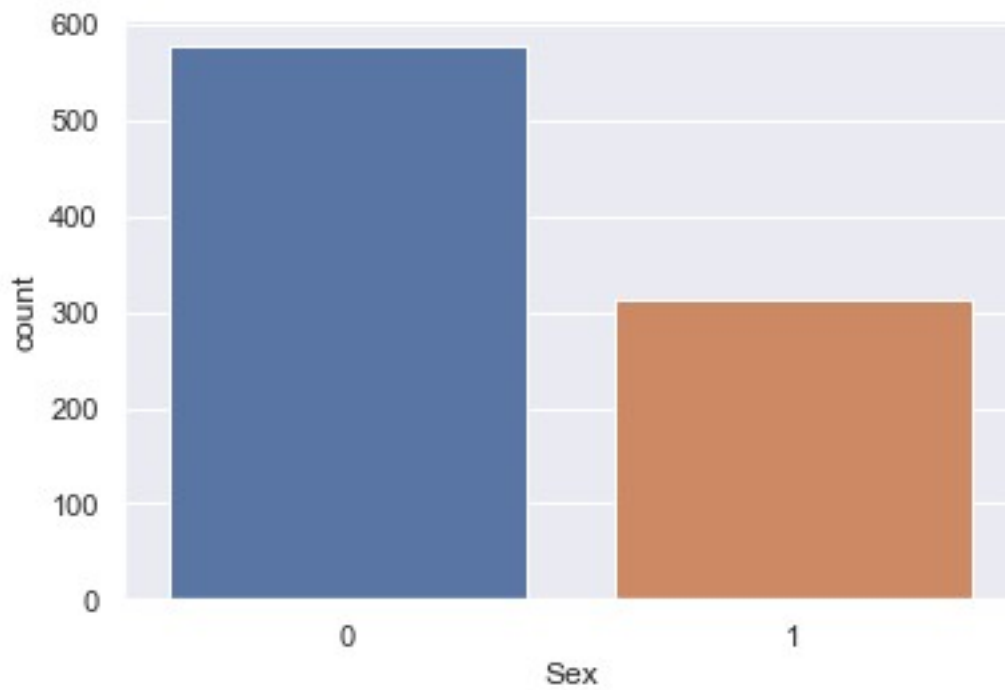
```
array([0, 1, 2, 5, 3, 4, 6], dtype=int64)
```

```
sns.set()
```

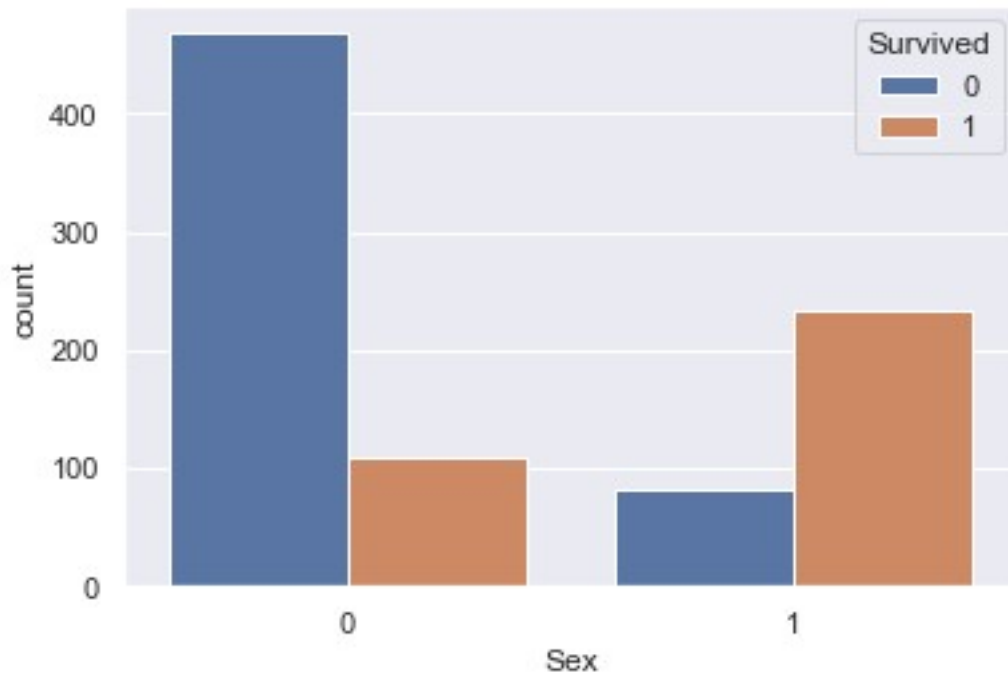
```
sns.countplot(x = titanic_data['Survived']).set_title('survived Vs
deceased');
```



```
sns.countplot(x = titanic_data['Sex']);
```



```
sns.countplot('Sex', hue='Survived', data = titanic_data);
```



```
titanic_data['Pclass'].value_counts()
```

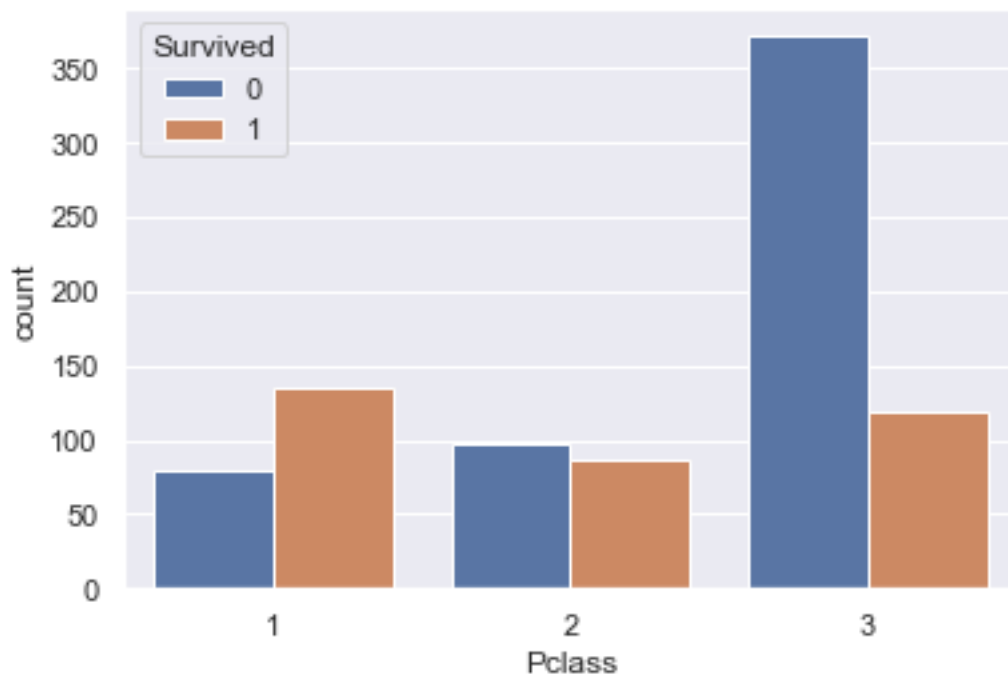
```
3    491
```

```
1    216
```

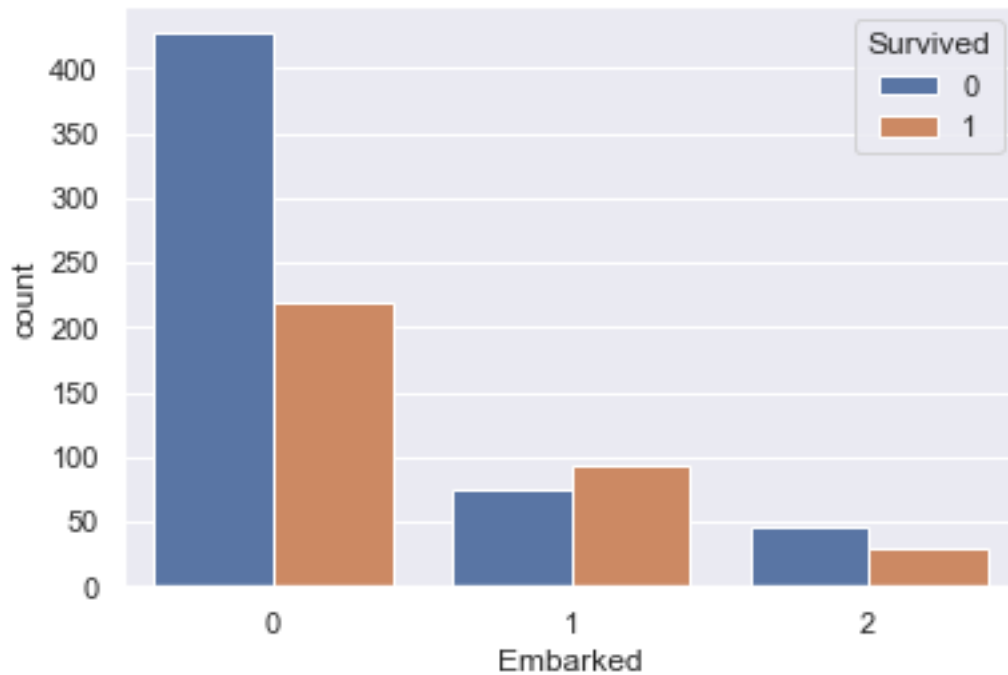
```
2    184
```

```
Name: Pclass, dtype: int64
```

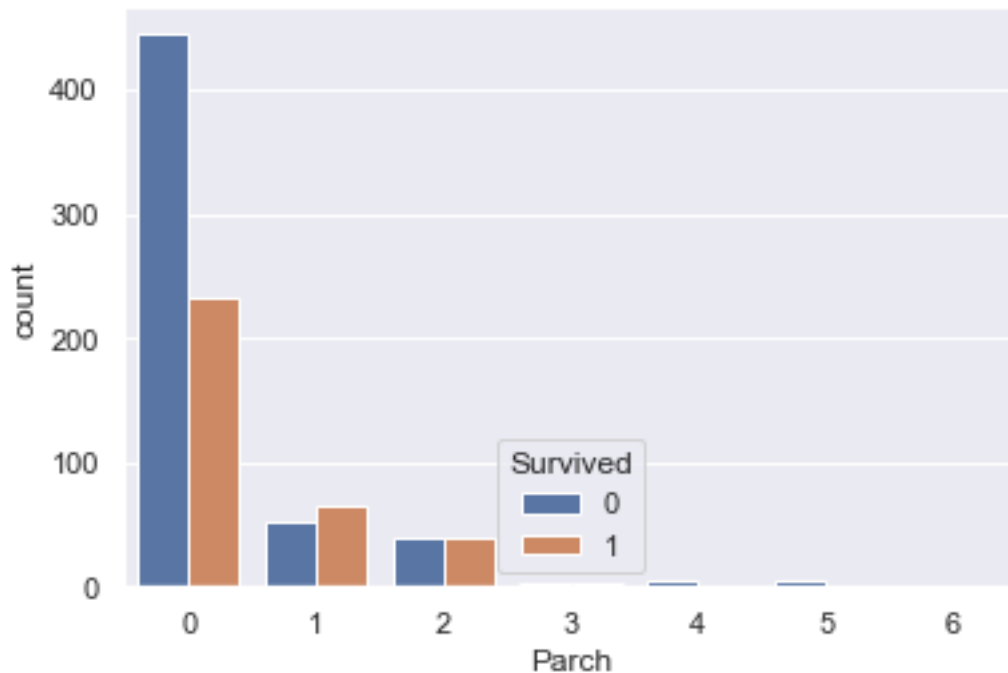
```
sns.countplot('Pclass', hue='Survived', data = titanic_data);
```



```
sns.countplot('Embarked', hue='Survived', data = titanic_data);
```



```
sns.countplot('Parch', hue='Survived', data = titanic_data);
```



```
titanic_data
```

```
0    PassengerId  Survived  Pclass \
0         1         0         3
```

1	2	1	1
2	3	1	3
3	4	1	1
4	5	0	3
..
886	887	0	2
887	888	1	1
888	889	0	3
889	890	1	1
890	891	0	3

SibSp \	Name	Sex	Age
0	Braund, Mr. Owen Harris	0	22.000000
1			
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.000000
1			
2	Heikkinen, Miss. Laina	1	26.000000
0			
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.000000
1			
4	Allen, Mr. William Henry	0	35.000000
0			
..
...			
886	Montvila, Rev. Juozas	0	27.000000
0			
887	Graham, Miss. Margaret Edith	1	19.000000
0			
888	Johnston, Miss. Catherine Helen "Carrie"	1	29.699118
1			
889	Behr, Mr. Karl Howell	0	26.000000
0			
890	Dooley, Mr. Patrick	0	32.000000
0			

	Parch	Ticket	Fare	Embarked
0	0	A/5 21171	7.2500	0
1	0	PC 17599	71.2833	1
2	0	STON/O2. 3101282	7.9250	0
3	0	113803	53.1000	0
4	0	373450	8.0500	0
..
886	0	211536	13.0000	0
887	0	112053	30.0000	0
888	2	W./C. 6607	23.4500	0
889	0	111369	30.0000	1
890	0	370376	7.7500	2

[891 rows x 11 columns]


```
titanic_data.dtypes
```

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

```
X = titanic_data.drop(columns=
['PassengerId', 'Name', 'Ticket', 'Survived'], axis=1)
Y = titanic_data['Survived']
```

```
print(X,Y)
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	22.000000	1	0	7.2500	0
1	1	1	38.000000	1	0	71.2833	1
2	3	1	26.000000	0	0	7.9250	0
3	1	1	35.000000	1	0	53.1000	0
4	3	0	35.000000	0	0	8.0500	0
..
886	2	0	27.000000	0	0	13.0000	0
887	1	1	19.000000	0	0	30.0000	0
888	3	1	29.699118	1	2	23.4500	0
889	1	0	26.000000	0	0	30.0000	1
890	3	0	32.000000	0	0	7.7500	2

```
[891 rows x 7 columns] 0      0
```

```
1      1
2      1
3      1
4      0
..
886    0
887    1
888    0
889    1
890    0
```

```
Name: Survived, Length: 891, dtype: int64
```

```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=
0.2,random_state=2)
```

```
print(X_train.shape,X_test.shape,Y_train.shape,Y_test.shape)
```

```
(712, 7) (179, 7) (712,) (179,)
```

Model Training:

Logistic Regression

```
logreg = LogisticRegression()
```

```
logreg.fit(X_train,Y_train)
```

```
LogisticRegression()
```

Model Evaluation:

```
X_train_pred = logreg.predict(X_train)
```

```
X_train_pred.shape
```

```
(712,)
```

```
ac_training = accuracy_score(Y_train,X_train_pred)
```

```
print('Training Accuracy= ', round(ac_training * 100), '%')
```

```
Training Accuracy= 81 %
```

```
X_test_pred = logreg.predict(X_test)
```

```
X_test_pred.shape
```

```
(179,)
```

```
ac_testing = accuracy_score(Y_test,X_test_pred)
```

```
print('Testing Accuracy= ', round(ac_testing * 100), '%')
```

```
Testing Accuracy= 78 %
```

```
from sklearn.metrics import confusion_matrix
```

```
cf=confusion_matrix(Y_test,X_test_pred)
```

```
cf
```

```
array([[91,  9],  
       [30, 49]], dtype=int64)
```