

TASK-1 : TITANIC SURVIVAL PREDICTION ANALYSIS

Author : VISHRUT VERMA  
Batch : JULY Batch 4  
Domain : Data Science  
Aim : To Build a Model that predicts whether the Titanic is survived or not.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df= pd.read_csv("/content/archive.zip")
df.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

Next steps:

Generate code with df

View recommended plots

df.shape

(891, 12)

df.describe()

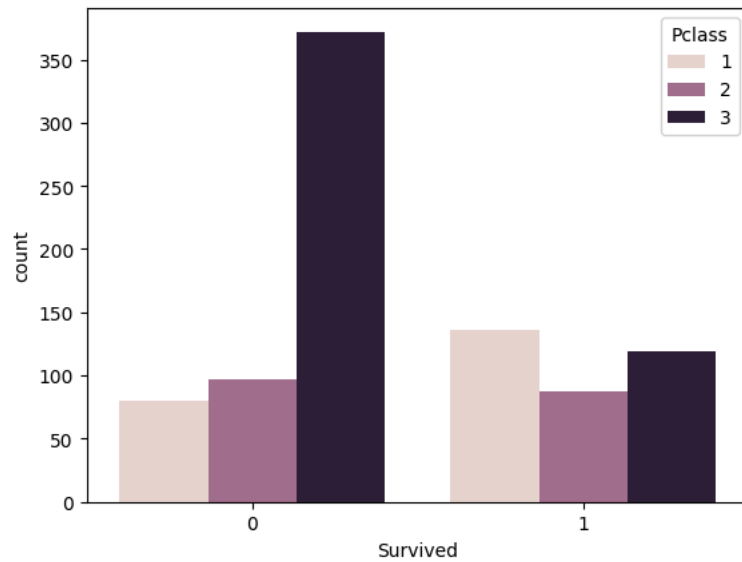
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.2042
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.6934
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.9104
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.4542
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.0000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.3290

df['Survived'].value\_counts()

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

```
#Let visualize the count of Survivals wrt Pclass
sns.countplot(x='Survived', hue='Pclass', data=df)
```

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

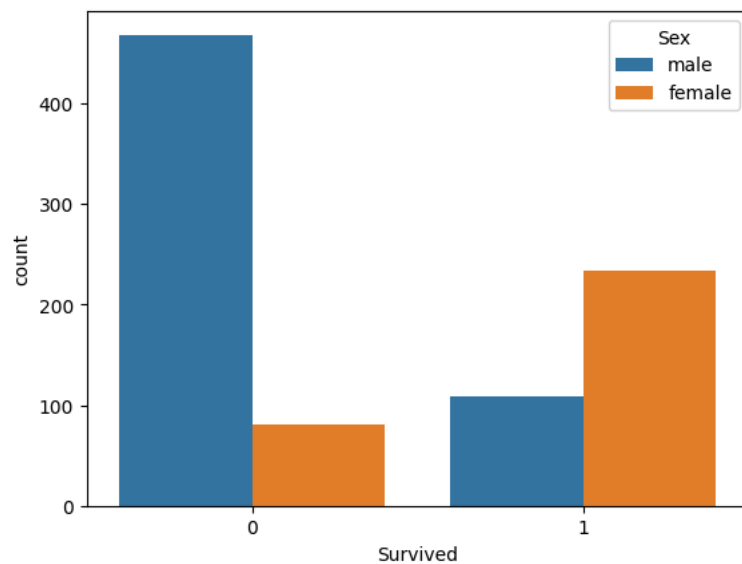


```
df["Sex"]
```

```
0    male
1    female
2    female
3    female
4    male
...
886   male
887   female
888   female
889   male
890   male
Name: Sex, Length: 891, dtype: object
```

```
#Let's Visualize the count of Survivals wrt Gender
sns.countplot(x='Survived', hue='Sex', data=df)
```

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



```
# Look at Survival rate by Sex
df.groupby('Sex')[['Survived']].mean()
```

```
Survived
Sex
female 0.742038
male   0.188908
```

```
df['Sex'].unique()
```

```
array(['male', 'female'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Sex'] = le.fit_transform(df['Sex'])

df.head()
```

```
PassengerId  Survived  Pclass      Name Sex  Age  SibSp  Parch  Ticket  Fare
0           1         0       3  Braund, Mr. Owen Harris    1  22.0    1    0  A/5 21171  7.2500
1           2         1       1  Cumings, Mrs. John Bradley (Florence Briggs)  0  38.0    1    0  PC 17599  71.2834
```

Next steps:

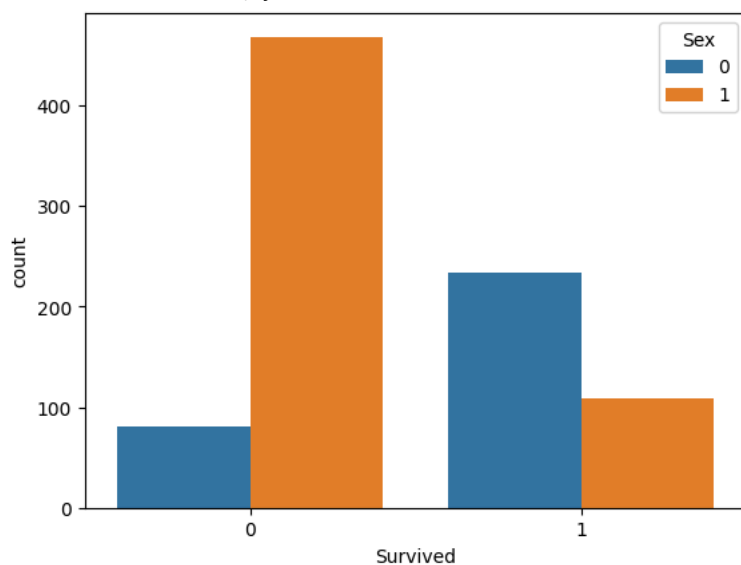
[Generate code with df](#)[View recommended plots](#)

```
df['Sex'], df['Survived']
```

```
(0    1
 1    0
 2    0
 3    0
 4    1
 ..
886   1
887   0
888   0
889   1
890   1
Name: Sex, Length: 891, dtype: int64,
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)
```

```
sns.countplot(x='Survived', hue='Sex', data=df)
```

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



```
df.isna().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
```

```
# After Dropping non required column
df.drop(['Cabin'], axis=1, inplace=True)
```

```
df.final = df
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs T. B.)	0	38.0	1	0	PC 17599	71.2834

Next steps: [Generate code with df](#) [View recommended plots](#)

```
X = df[['Pclass', 'Sex']]
Y = df['Survived']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression(random_state=0)
log.fit(X_train, Y_train)
```

```
LogisticRegression
LogisticRegression(random_state=0)
```

```
pred = print(log.predict(X_test))
```

```
[0 0 0 1 1 0 1 1 0 1 0 1 1 1 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 1 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 1 1 0 0 0
 0 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 1 1 0 1 0
 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 1 1 1 0 1
 1 0 0 1 1 0 1 0 1 0 1 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0]
```

```
print(Y_test)
```

```
495    0
648    0
278    0
31     1
255    1
..
780    1
837    0
215    1
833    0
372    0
Name: Survived, Length: 179, dtype: int64
```

```
print(X_test)
```

```
Pclass Sex
495     3   1
648     3   1
278     3   1
31      1   0
255     3   0
..
780     3   0
837     3   1
215     1   0
833     3   1
372     3   1
```

[179 rows x 2 columns]

Suggested code may be subject to a license | cdkushan/AD-SALE-PREDICT |

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
res= log.predict([[2,0]])
```

```
if (res == 1):
```

```
    print("Survived")
```

```
else:
```

```
    print("Not Survived")
```

```
→ Survived
```