

TASK 1:TITANIC SURVIVAL PREDICTION

NAME: NUTAN SANTOSH BHILARE EMAIL: [nutan10232@gmail.com](mailto:nutan10232@gmail.com) domain: data science

Importing important libraries

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

Importing dataset

```
df=pd.read_csv("/content/archive.zip")
df.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
				Futrelle, Mrs. Jacques						

Next steps:

Generate code with df

View recommended plots

```
df.shape
(891, 12)
```

```
df.describe()
```

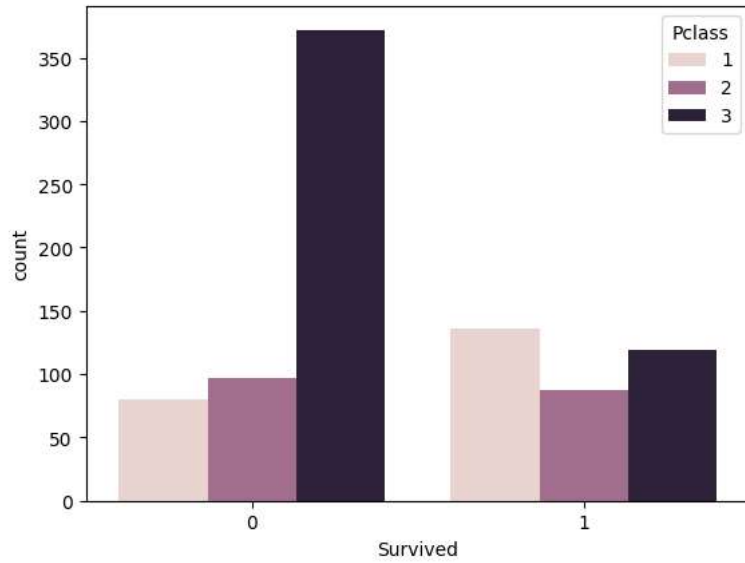
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

from above cell there are two mising values in age column

```
df['Survived'].value_counts()
0    549
1    342
Name: Survived, dtype: int64
```

```
#visualising the count of survivals pclass
sns.countplot(x=df['Survived'], hue=df['Pclass'])
```

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

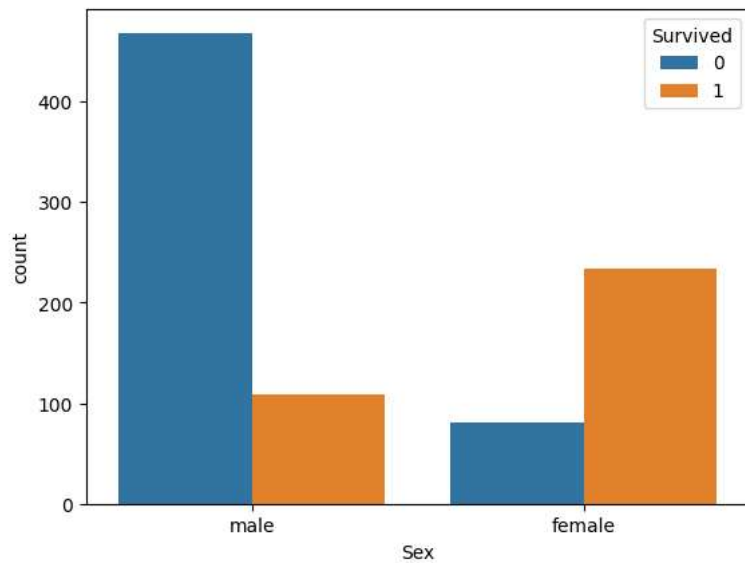


```
df["Sex"]
```

```
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
```

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

<Axes: xlabel='Sex', ylabel='count'>



```
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
labelencoder = LabelEncoder()
```

```
df['Sex']= labelencoder.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	0	38.0	1	0	PC 17599	71.2833

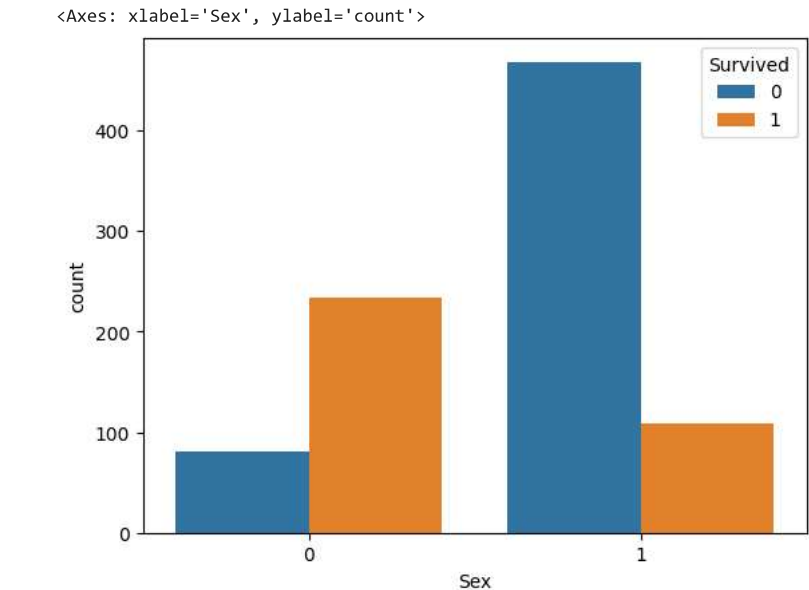
Next steps:

[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=df['Sex'], hue=df["Survived"])
```



```
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
```

```
df=df.drop(['Age'], axis=1)
```

```
df_final = df
df_final.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	1	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	0	0	0	STON/O2. 3101282	7.9250	NaN
				Futrelle, Mrs. Jacques						

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

Model Training

```
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)
```

## Model Prediction

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

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

```
import warnings
warnings.filterwarnings("ignore")
```

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

```
if(res==0):
    print("so sorry! not survived")
```