

Task-1 : TITANIC SURVIVAL PREDICTION

```
In [1]: import pandas as pd
import matplotlib
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
```

```
In [3]: import os
```

```
In [4]: os.chdir("C:\\Users\\fatin\\OneDrive\\Desktop")
```

```
In [5]: task1 = pd.read_csv("Titanic-Dataset.csv")
```

```
In [6]: task1.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	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

```
In [7]: task1.shape
```

```
Out[7]: (891, 12)
```

```
In [8]: task1.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	

```
In [9]: task1['Age']
```

```
Out[9]: 0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
...
886     27.0
887     19.0
888      NaN
889     26.0
890     32.0
Name: Age, Length: 891, dtype: float64
```

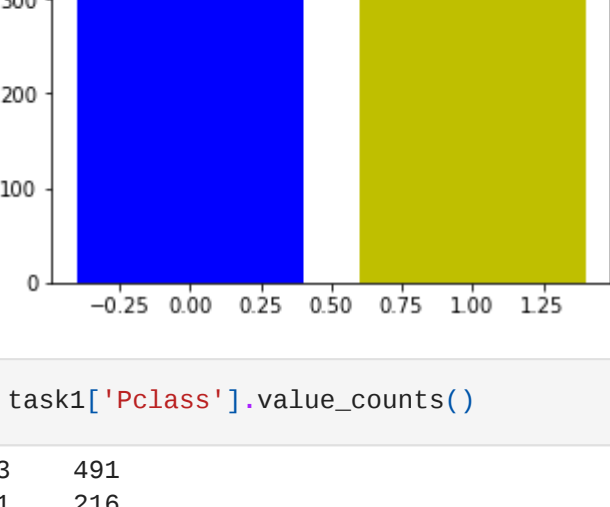
```
In [10]: task1['Survived'].value_counts()
```

```
Out[10]: 0      549
1       342
Name: Survived, dtype: int64
```

```
In [11]: task1['Survived'].value_counts().keys()
```

```
Out[11]: Int64Index([0, 1], dtype='int64')
```

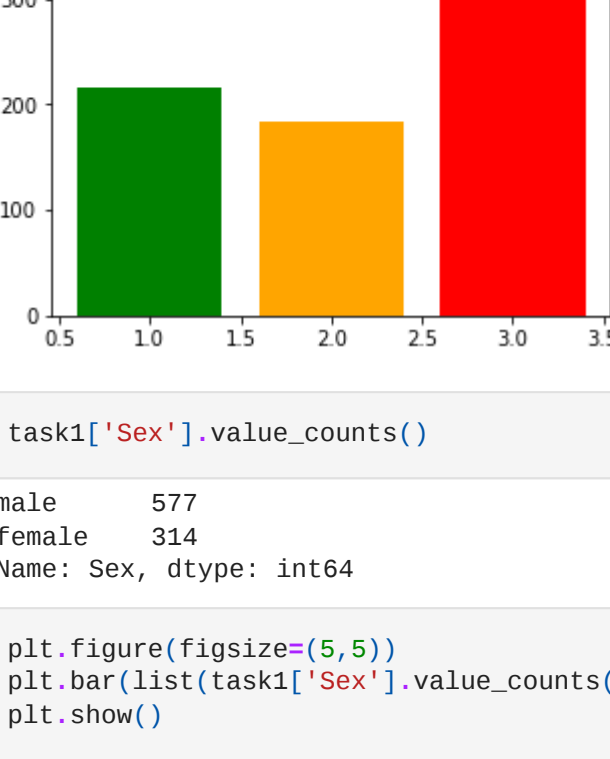
```
In [12]: plt.figure(figsize=(5,5))
plt.bar(list(task1['Survived'].value_counts().keys()),list(task1['Survived'].value_counts()))
plt.show()
```



```
In [13]: task1['Pclass'].value_counts()
```

```
Out[13]: 3      491
1       216
2       184
Name: Pclass, dtype: int64
```

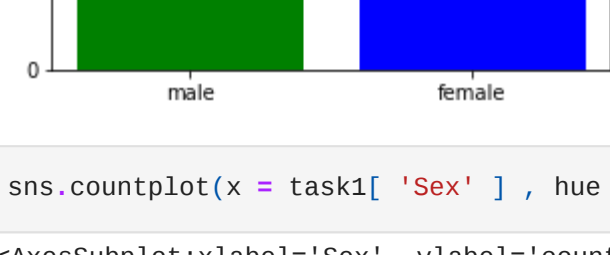
```
In [14]: plt.figure(figsize=(5,5))
plt.bar(list(task1['Pclass'].value_counts().keys()),list(task1['Pclass'].value_counts()))
plt.show()
```



```
In [15]: task1['Sex'].value_counts()
```

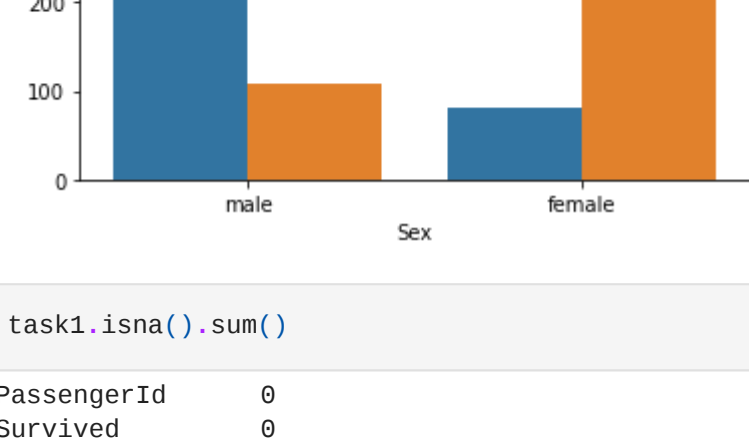
```
Out[15]: male      577
female    314
Name: Sex, dtype: int64
```

```
In [16]: plt.figure(figsize=(5,5))
plt.bar(list(task1['Sex'].value_counts().keys()),list(task1['Sex'].value_counts()))
plt.show()
```



```
In [17]: sns.countplot(x = task1[ 'Sex' ] , hue = task1['Survived'])
```

```
Out[17]: <AxesSubplot:xlabel='Sex', ylabel='count'>
```



```
In [18]: task1.isna().sum()
```

```
Out[18]: 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
```

```
In [19]: task1.replace({'Sex':{'male' : 0 , 'female' : 1 }}, inplace = True)
task1.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	NaN	

```
In [20]: x = task1[['Pclass','Sex']]
y = task1['Survived']
```

```
In [21]: x,y
```

```
Out[21]: (      Pclass  Sex
0           3     0
1           1     1
2           3     1
3           1     1
4           3     0
...
886         2     0
887         1     1
888         3     1
889         1     0
890         3     0

[891 rows x 2 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)
```

```
In [22]: 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 = 42)
```

```
In [23]: print(x.shape , x_train.shape , x_test.shape)
```

```
(891, 2) (712, 2) (179, 2)
```

```
In [24]: print(y.shape , y_train.shape , y_test.shape)
```

```
(891,) (712,) (179,)
```

```
In [25]: from sklearn.linear_model import LogisticRegression
ML = LogisticRegression()
ML.fit(x_train , y_train)
```

```
Out[25]: LogisticRegression()
```

```
In [26]: x_test_prediction = ML.predict(x_test)
print(x_test_prediction)
```

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

```
In [28]: 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
```

```
In [43]: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

model = LogisticRegression()
model.fit(x_train, y_train)

y_train_pred = model.predict(x_train)
y_test_pred = model.predict(x_test)

train_accuracy = accuracy_score(y_train, y_train_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)

print("Training accuracy:", train_accuracy)
print("Test accuracy:", test_accuracy)
```

```
Training accuracy: 0.7865168539325843
Test accuracy: 0.7877094972067039
```

```
In [44]: model.fit(x_train, y_train)

x_train_prediction = model.predict(x_train)

train_accuracy = accuracy_score(y_train, x_train_prediction)

print("Training accuracy:", train_accuracy)
```

```
Training accuracy: 0.7865168539325843
```

```
In [45]: from sklearn.metrics import accuracy_score

train_accuracy = accuracy_score(y_train , x_train_prediction)
test_accuracy = accuracy_score(y_test , x_test_prediction)

print("Accuracy scores of training and test data are " , train_accuracy , "and" , test_accuracy)
```

```
Accuracy scores of training and test data are 0.7865168539325843 and 0.7877094972067039 respectively
```

```
In [ ]:
```