

```
In [1]: import pandas as pd
```

```
In [2]: df=pd.read_csv('titanic.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C8
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Na
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C12
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Na

```
In [3]: df.columns
```

```
Out[3]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype='object')
```

```
In [4]: df.shape
```

```
Out[4]: (891, 12)
```

## Data Cleaning

```
In [5]: df.drop(['PassengerId', 'Name', 'SibSp', 'Parch', 'Ticket', 'Cabin', 'Embarked'], axis=
```

```
In [6]: df.head()
```

```
Out[6]:
```

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

### find out how many people are alive

```
In [7]: a =df[df.Survived==1]
```

```
In [8]: a.shape
```

```
Out[8]: (342, 5)
```

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

```
Out[9]: (891, 5)
```

```
In [10]: print('total unsurvive people is', 891-342)
```

```
total unsurvive people is 549
```

### find out how many of then are male and Female

```
In [11]: b=df[df.Sex=='male']  
b.shape
```

```
Out[11]: (577, 5)
```

```
In [12]: print(f'Male are 577 and female are {891-577}.')
```

```
Male are 577 and female are 314.
```

### find out how many of then are male and Female are alive

```
In [14]: # c=df([[df.Sex=='male',df.Survived=1]], 'inPlace=True)
```

## Naive Baves Classifier

```
In [15]: target=df.Survived
input=df.drop('Survived', axis = 'columns')
```

```
In [16]: dummies = pd.get_dummies(input.Sex)
dummies.head()
```

```
Out[16]:
```

	female	male
0	0	1
1	1	0
2	1	0
3	1	0
4	0	1

```
In [17]: input= pd.concat([input,dummies],axis = 'columns')
input.head()
```

```
Out[17]:
```

	Pclass	Sex	Age	Fare	female	male
0	3	male	22.0	7.2500	0	1
1	1	female	38.0	71.2833	1	0
2	3	female	26.0	7.9250	1	0
3	1	female	35.0	53.1000	1	0
4	3	male	35.0	8.0500	0	1

```
In [18]: input.drop(['Sex'], axis = 'columns', inplace = True)
input.head()
```

```
Out[18]:
```

	Pclass	Age	Fare	female	male
0	3	22.0	7.2500	0	1
1	1	38.0	71.2833	1	0
2	3	26.0	7.9250	1	0
3	1	35.0	53.1000	1	0
4	3	35.0	8.0500	0	1

```
In [19]: input.columns[input.isna().any()]
```

```
Out[19]: Index(['Age'], dtype='object')
```

```
In [20]: d=input[input.Age==20]
d.shape
```

```
Out[20]: (15, 5)
```

```
In [26]: input.Age[:10]
```

```
Out[26]: 0    22.000000
         1    38.000000
         2    26.000000
         3    35.000000
         4    35.000000
         5    29.699118
         6    54.000000
         7     2.000000
         8    27.000000
         9    14.000000
         Name: Age, dtype: float64
```

```
In [25]: input.Age = input.Age.fillna(input.Age.mean())
```

```
In [28]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(input,target,test_size=0.3)
```

```
In [29]: len(x_train)
```

```
Out[29]: 623
```

```
In [34]: from sklearn.naive_bayes import GaussianNB
         model = GaussianNB()
```

```
In [35]: model.fit(x_train,y_train)
```

```
Out[35]: GaussianNB()
```

```
In [36]: model.score(x_test,y_test)
```

```
Out[36]: 0.7835820895522388
```

```
In [ ]:
```