

```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import pandas as pd
```

```
salary_train=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SalaryData_Train.csv')
salary_train
```

	age	workclass	education	educationno	maritalstatus	occupation	relation
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-f
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Hus
2	38	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-f
3	53	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Hus
4	28	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	
...	
30156	27	Private	Assoc-acdm	12	Married-civ-spouse	Tech-support	
30157	40	Private	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Hus
30158	58	Private	HS-grad	9	Widowed	Adm-clerical	Unm
30159	22	Private	HS-grad	9	Never-married	Adm-clerical	Own
30160	52	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	

30161 rows × 14 columns

```
salary_test = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/SalaryData_Test.csv')
salary_test
```

	age	workclass	education	educationno	maritalstatus	occupation	relation
0	25	Private	11th	7	Never-married	Machine-op-inspct	Own
1	38	Private	HS-grad	9	Married-civ-spouse	Farming-fishing	Hus
2	28	Local-gov	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Hus
3	44	Private	Some-college	10	Married-civ-spouse	Machine-op-inspct	Hus
4	34	Private	10th	6	Never-married	Other-service	Not-in-f
...
15055	33	Private	Bachelors	13	Never-married	Prof-specialty	Own
15056	39	Private	Bachelors	13	Divorced	Prof-specialty	Not-in-f
15057	38	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Hus
15058	44	Private	Bachelors	13	Divorced	Adm-clerical	Own
15059	35	Self-emp-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Hus

15060 rows × 14 columns

salary_train.shape , salary_test.shape

((30161, 14), (15060, 14))

salary_train.isna().sum()

```
age          0
workclass    0
education    0
educationno  0
maritalstatus 0
occupation   0
relationship 0
race         0
sex          0
capitalgain  0
capitalloss  0
hoursperweek 0
native       0
```

```
Salary      0
dtype: int64
```

```
salary_test.isna().sum()
```

```
age          0
workclass    0
education    0
educationno   0
maritalstatus 0
occupation   0
relationship 0
race         0
sex          0
capitalgain   0
capitalloss   0
hoursperweek 0
native       0
Salary       0
dtype: int64
```

```
salary_train.dtypes
```

```
age          int64
workclass    object
education    object
educationno   int64
maritalstatus object
occupation   object
relationship  object
race         object
sex          object
capitalgain   int64
capitalloss   int64
hoursperweek  int64
native       object
Salary       object
dtype: object
```

```
salary_test.dtypes
```

```
age          int64
workclass    object
education    object
educationno   int64
maritalstatus object
occupation   object
relationship  object
race         object
sex          object
capitalgain   int64
capitalloss   int64
hoursperweek  int64
native       object
```

```
Salary      object
dtype: object
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
salary_train['workclass']=le.fit_transform(salary_train['workclass'])
salary_train['education']=le.fit_transform(salary_train['education'])
salary_train['maritalstatus']=le.fit_transform(salary_train['maritalstatus'])
salary_train['occupation']=le.fit_transform(salary_train['occupation'])
salary_train['relationship']=le.fit_transform(salary_train['relationship'])
salary_train['race']=le.fit_transform(salary_train['race'])
salary_train['sex']=le.fit_transform(salary_train['sex'])
salary_train['native']=le.fit_transform(salary_train['native'])
salary_train['Salary']=le.fit_transform(salary_train['Salary'])
salary_train
```

	age	workclass	education	educationno	maritalstatus	occupation	relatior
0	39	5	9	13	4	0	
1	50	4	9	13	2	3	
2	38	2	11	9	0	5	
3	53	2	1	7	2	5	
4	28	2	9	13	2	9	
...	
30156	27	2	7	12	2	12	
30157	40	2	11	9	2	6	
30158	58	2	11	9	6	0	
30159	22	2	11	9	4	0	
30160	52	3	11	9	2	3	

30161 rows × 14 columns

```
salary_train.dtypes
```

```
age          int64
workclass    int64
education    int64
educationno  int64
maritalstatus int64
occupation   int64
relationship int64
race         int64
```

```

sex            int64
capitalgain    int64
capitalloss    int64
hoursperweek   int64
native         int64
Salary         int64
dtype: object

```

```

salary_test['workclass']=le.fit_transform(salary_test['workclass'])
salary_test['education']=le.fit_transform(salary_test['education'])
salary_test['maritalstatus']=le.fit_transform(salary_test['maritalstatus'])
salary_test['occupation']=le.fit_transform(salary_test['occupation'])
salary_test['relationship']=le.fit_transform(salary_test['relationship'])
salary_test['race']=le.fit_transform(salary_test['race'])
salary_test['sex']=le.fit_transform(salary_test['sex'])
salary_test['native']=le.fit_transform(salary_test['native'])
salary_test['Salary']=le.fit_transform(salary_test['Salary'])
salary_test

```

	age	workclass	education	educationno	maritalstatus	occupation	relatior
0	25	2	1	7	4	6	
1	38	2	11	9	2	4	
2	28	1	7	12	2	10	
3	44	2	15	10	2	6	
4	34	2	0	6	4	7	
...	
15055	33	2	9	13	4	9	
15056	39	2	9	13	0	9	
15057	38	2	9	13	2	9	
15058	44	2	9	13	0	0	
15059	35	3	9	13	2	3	

15060 rows × 14 columns

```

X=salary_train.drop(['Salary'], axis=1)
y=salary_train['Salary']

```

```

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,plot_confusion_matrix,accuracy_score

```

```

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20, random_state=12)

```

```
from sklearn.naive_bayes import MultinomialNB
nb_classifier = MultinomialNB()
```

```
nb_classifier.fit(X_train,y_train)
```

```
    MultinomialNB()
```

```
y_pred = nb_classifier.predict(X_test)
y_pred
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
accuracy = accuracy_score(y_test, y_pred)
confusion_matrix = confusion_matrix(y_test,y_pred)
```

```
accuracy
```

```
0.7720868556273828
```

```
confusion_matrix
```

```
array([[4358, 199],
       [1176, 300]])
```

```
X1=salary_test.drop(['Salary'], axis=1)
y1=salary_test['Salary']
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix,plot_confusion_matrix,accuracy_score
```

```
X_train,X_test,y_train,y_test=train_test_split(X1,y1,test_size=0.20, random_state=12)
```

```
from sklearn.naive_bayes import MultinomialNB
nb_classifier = MultinomialNB()
```

```
nb_classifier.fit(X_train,y_train)
```

```
    MultinomialNB()
```

```
y_pred = nb_classifier.predict(X_test)
```

y_pred

```
array([0, 0, 1, ..., 0, 0, 0])
```

```
accuracy = accuracy_score(y_test, y_pred)  
confusion_matrix = confusion_matrix(y_test,y_pred)
```

accuracy

```
0.7606241699867198
```

confusion_matrix

```
array([[2125, 109],  
       [ 612, 166]])
```

✓ 0s completed at 4:05 PM

● ✕