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

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

```
     Mounted at /content/drive
```

```
import pandas as pd
```

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

	age	workclass	education	educationno	maritalstatus	occupation	relatio
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.shape
```

```
     (30161, 14)
```

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

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

30161 rows × 14 columns

```
salary.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
```

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

✓ 0s completed at 8:13 PM ● ✕