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

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

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

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
sal_train = pd.read_csv('/content/drive/MyDrive/SalaryData_Train(1).csv')
sal_train
```

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

30161 rows × 14 columns

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

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

15060 rows × 14 columns

```
sal_train.shape , sal_test.shape
```

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

```
sal_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
```

```
sal_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
```

```
sal_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
```

```
sal_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()
```

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

	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

```
sal_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
```

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

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

15060 rows × 14 columns

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

```
X
```

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

15060 rows × 13 columns

```
y
```

```
0        0
1        0
2        1
3        1
4        0
        ..
15055    0
15056    0
15057    0
15058    0
15059    1
Name: Salary, Length: 15060, dtype: int64
```

```
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=0.30,random_state=1)


from sklearn.svm import SVC


rbf_classifier = SVC(kernel='rbf', C=0.01, gamma=0.1)
rbf_classifier
rbf_classifier.fit(X_train,y_train)
y_test_pred=rbf_classifier.predict(X_test)


y_test_pred
```

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

```
accuracy_score(y_test,y_test_pred)
```

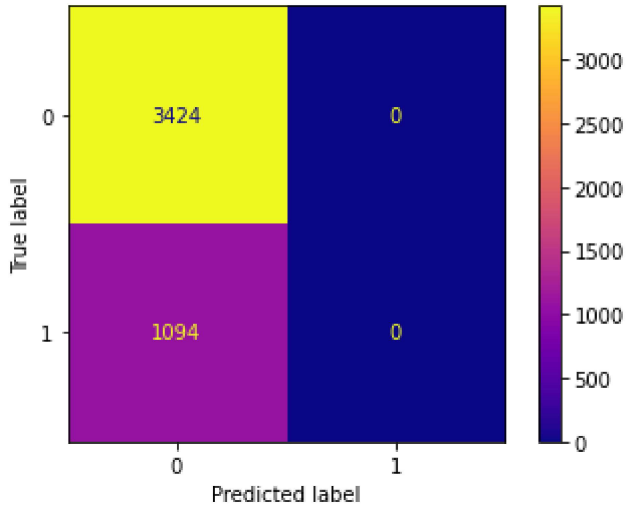
```
0.7578574590526782
```

```
confusion_matrix(y_test, y_test_pred)
```

```
array([[3424,    0],
       [1094,    0]])
```

```
plot_confusion_matrix(rbf_classifier,X_test,y_test, cmap='plasma')
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: F
  warnings.warn(msg, category=FutureWarning)
```



```
classifier=SVC(kernel='linear', C=0.01, gamma=0.1)
classifier.fit(X_train, y_train)
y_test_pred=classifier.predict(X_test)
```

```
accuracy_score(y_test,y_test_pred)
```

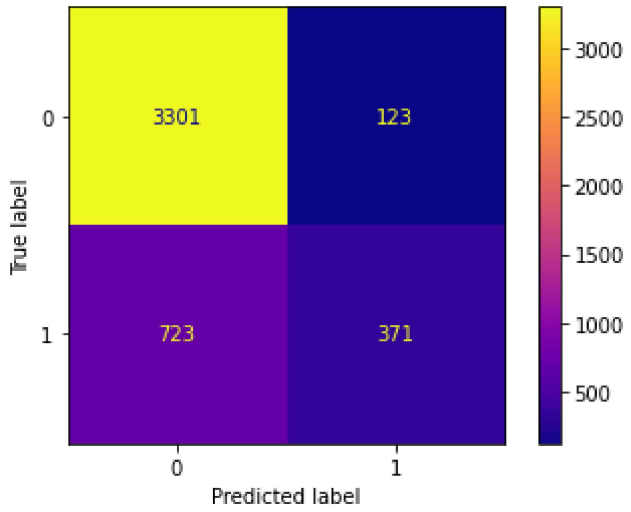
```
0.8127490039840638
```

```
confusion_matrix(y_test,y_test_pred)
```

```
array([[3301,  123],
       [ 723,  371]])
```

```
plot_confusion_matrix(classifier,X_test,y_test, cmap='plasma')
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: F
  warnings.warn(msg, category=FutureWarning)
```



✓ 0s completed at 4:14 PM