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

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

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

```
import pandas as pd
```

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

	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

```
data.shape
```

```
(30161, 14)
```

```
data.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
```

```
data.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()
```

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

	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

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

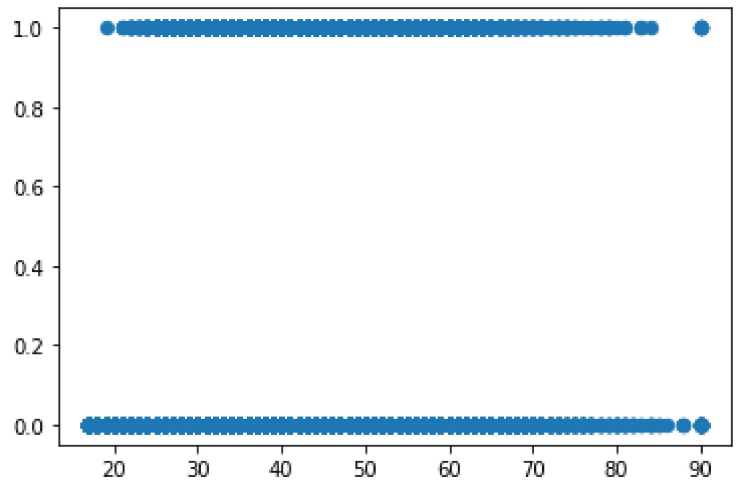
```
X
```

	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	

```
y
```

```
0        0
1        0
2        0
3        0
4        0
        ..
30156    0
30157    1
30158    0
30159    0
30160    1
Name: Salary, Length: 30161, dtype: int64
```

```
import matplotlib.pyplot as plt
plt.scatter(data['age'], y, s=40, alpha=1)
plt.show()
```




```
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=20)


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)
```

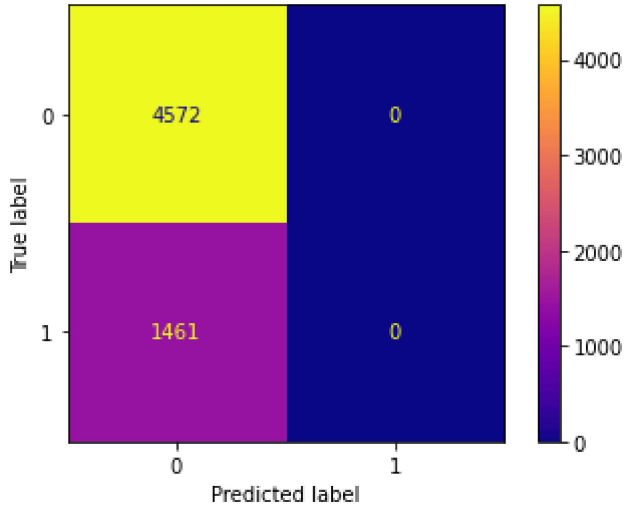
```
0.7578319244157136
```

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

```
array([[4572,    0],
       [1461,    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: FutureWarn
  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)
```

```
0.8145201392342118
```

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

```
array([[4455,  117],
       [1002,  459]])
```

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

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