```
# Support Vector Machine (SVM)

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv(r'C:\Users\Admin\Desktop\NIT\1. NIT_Batches\2. EVENING BATCH\N_Batch -- 5.30PM -- Mar26\3. Jan\29th - SVM\Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Training the SVM model on the Training set
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', degree = 5)
classifier.fit(X_train, y_train)
```

```
# knn classifier algorithm
#from sklearn.neighbors import KNeighborsClassifier
#classifier_knn = KNeighborsClassifier
#classifier_knn.fit(X_train,y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)

# This is to get the Models Accuracy
from sklearn.metrics import accuracy_score
ac = accuracy_score(y_test, y_pred)
print(ac)

bias = classifier.score(X_train,y_train)
print(bias)

variance = classifier.score(X_test,y_test)
print(variance)

# This is to get the Classification Report
from sklearn.metrics import classification_report
cr = classification_report(y_test, y_pred)
cr
```