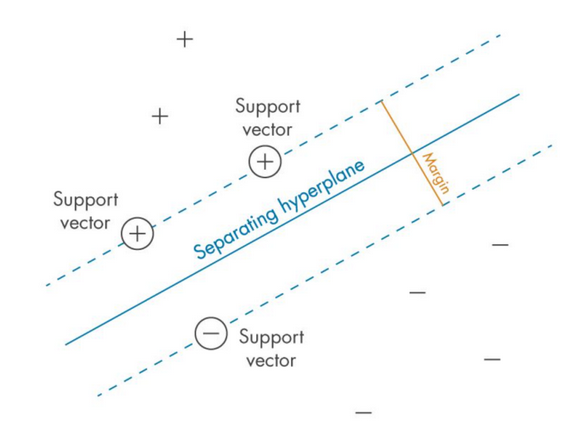
**Cycle-7**

**Aim:**Train, Validate and TestSupport Vector Machines (SVM)Model

**Solution:**

**Support Vector Machines (SVM) Classifier**

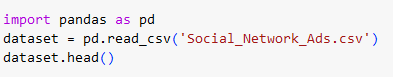
* Support vector machine (SVM) is a supervised machine learning algorithm that can be used for **both classification and regression** tasks.
* SVM for classification is termed as **support vector classification (SVC)** and SVM for regression is termed as **support vector regression (SVR)**
* The main idea behind SVM classifier is to find a **hyperplane** that maximally separates the data points of different classes.
* The hyperplane is selected such that it **maximizes the distance between the hyperplane and the closest data points from each class**, which are called **support vectors.**



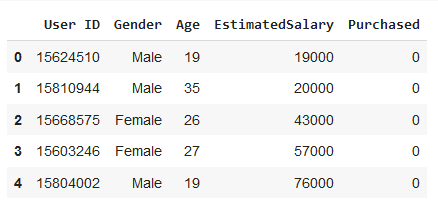
* When the data is not linearly separable, SVMs use a technique called **kernel trick**, which maps the original features into a higher-dimensional feature space where the data is linearly separable.
* Three hyperparameters that results in different SVM models : C, gamma and kernel function.
  + **C** is typically used as a **regularization parameter** allows the algorithm to make more accurate predictions on new data points.A **smaller value of C** will result in a larger margin, but may **lead to more misclassifications**.**A larger value of C** will result in a smaller margin, but **may lead to better classification accuracy**. A smaller value of C is preferred when the dataset is noisy. A larger value of C is preferred when the dataset is clean.
  + **gammahyperparameter** can take on two special values: ‘scale’, ‘auto’ or ‘float’. When gamma is set to ‘scale’, it means that the value of gamma is calculated as **1 / (number of features \* variance in data)**. This ensures that all features are given equal importance in the model and produces consistent results no matter how many features are used.
  + **SVM kernel** is a mathematical function that is used to map the data points from one space into another, usually higher dimensional space. It can take values ‘**linear’, ‘poly’, ‘rbf’ and ‘sigmoid’**.
    - **When kernel is set to ‘linear’**, the model will use a linear boundary for classification and regression. This is the simplest type of SVM and works best when **data are linearly separable**.
    - **When kernel is set to ‘poly’,** the model will use polynomial functions of degree higher than 1 for classification. This type of SVM is more suitable for **complex non-linear datasets**.
    - **When kernel is set to ‘rbf’,** the model will use radial basis funcitons for classification or regression. RBF kernels are capable ofdealing with **complex multi-class datasets and have good generalization performance with noisy data points.**
    - **When kernel is set to ‘sigmoid’**, that the model will apply sigmoid functions instead of RBFs for classification or regression tasks. Sigmoid kernels tend to be less sensitive than RBFs with respect to outliers in data but may not generalize as well unless their parameters are tuned properly.

**Source Code:**

### Step 1: Load the dataset

****

**Sample Output:**

****

### Step 2: Check for null values

### 

**Sample Output:**

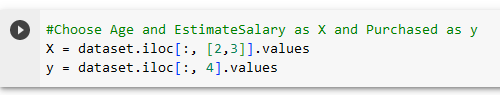
### 

### 

**Sample Output:**

### 

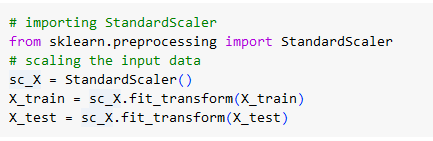
### Step 3: Choose input and output features

****

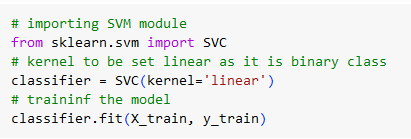
### Step 4: Splitting Data into Training and Testing Datasets

****

**Step 5: Scaling the input data**



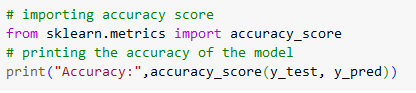
**Step 6: Training the model using Linear Kernel**



**Step 7: Testing the model**



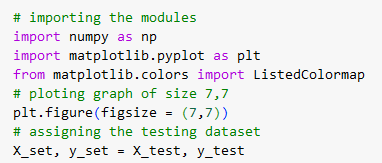
**Step 8: Find the Accuracy of the model**

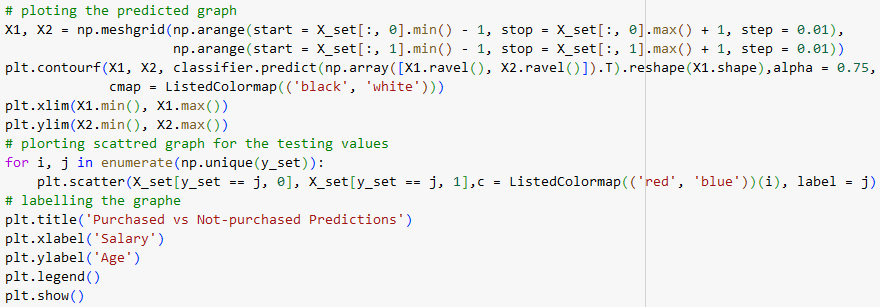


**Sample Output:**

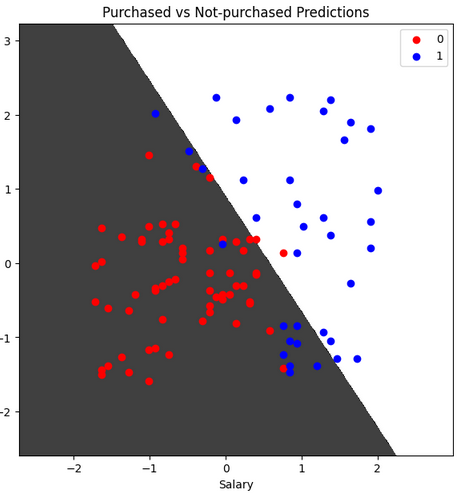


**Step 9:Visualization of testing data**

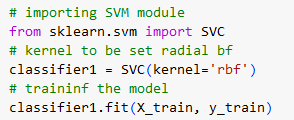




**Sample Output:**



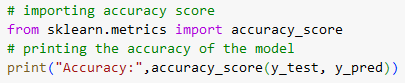
**Step 10: Training the model using Radial Base Function (rbf) kernel**



**Step 11: Testing the model**



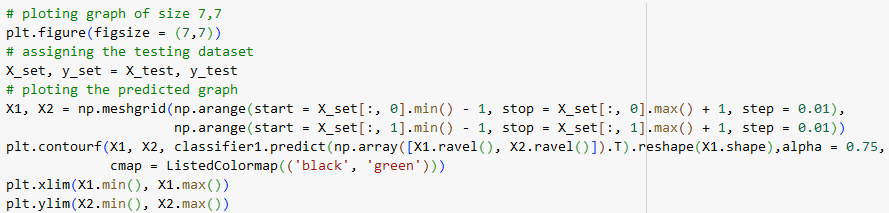
**Step 12: Find the Accuracy of the model**

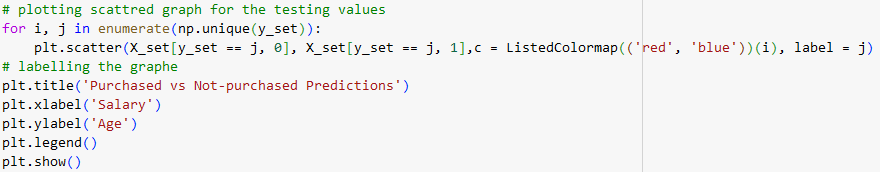


**Sample Output:**

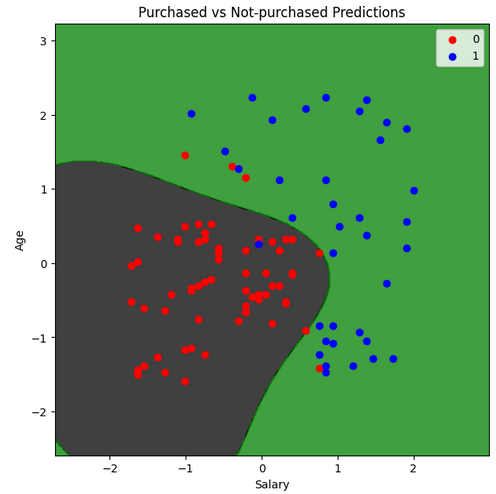


**Step 13:Visualization of testing data**

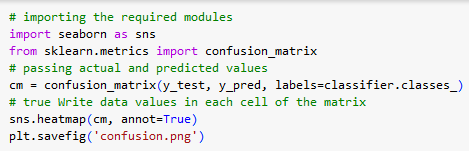


****

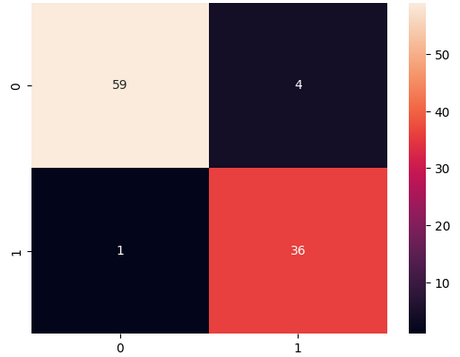
**Sample Output:**



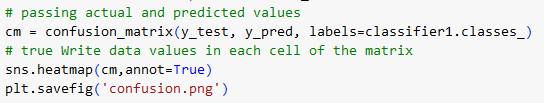
**Step 14:Displaying the confusion matrix for kernel=linear**



**Sample Output:**



**Step 14:Displaying the confusion matrix for kernel=rbf**



**Sample Output:**

