In [ ]:

```
import numpy as np
import pandas as pd
from sklearn.svm import SVC,SVR
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,mean_squared_error
from sklearn.metrics import roc_curve,roc_auc_score
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import KFold
import seaborn as sns
import matplotlib.pyplot as plt
```

Support Vector Clasification

In [ ]:

```
data=pd.read_csv("D:\\Academics\\SLIIT\\MLOM\\data\\Bank.CSV")
data.head()
```

In [ ]:

```
x=data.iloc[:,:7]
y=data.iloc[:,7]
```

In [ ]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

Creating the model

In [ ]:

```
svcl=SVC(kernel="linear",C=1,probability=True)
```

Training the model

In [ ]:

```
svcl.fit(x_train,y_train)
```

Predictions and accuracy

In [ ]:

```
y_pred=svcl.predict(x_test)
accuracy_score(y_test,y_pred)
```

Confusion Matrix

```
In [ ]: confusion_matrix(y_test,y_pred)
```

```
In [ ]: sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,fmt="g")
        plt.xlabel("Predicted")
        plt.ylabel("Actual")
        plt.show()
```

Classification Report

```
In [ ]: print(classification_report(y_test,y_pred))
```

ROC & AUC

```
In [ ]: y_pred_probs=svcl.predict_proba(x_test)
```

```
In [ ]: fpr, tpr, _ = roc_curve(y_test,  y_pred_probs[:,1])
        plt.plot(fpr,tpr)
        plt.title("ROC Curve")
        plt.show()
```

```
In [ ]: auc = roc_auc_score(y_test, y_pred_probs[:,1])
        auc
```

Selecting parameters with Hyper Parameter Optimization

```
In [ ]: params={"C":[0.5,1],"kernel":["linear","poly"]}
        model=SVC()
        cval=KFold(n_splits=2)
```

```
In [ ]: gsearch = GridSearchCV(model, params,cv=cval)
```

```
In [ ]: results = gsearch.fit(x_train, y_train)
```

```
In [ ]: results.best_params_
```

Support Vector Regression

```
In [ ]: data=pd.read_csv("D:\\Academics\\SLIIT\\MLOM\\data\\Boston.CSV")
        data.head()
```

```
In [ ]: x=data.iloc[:,:12]
        y=data.iloc[:,12]
```

```
In [ ]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=
        0)
```

```
In [ ]: svrg=SVR(kernel="linear",C=1)
```

```
In [ ]: svrg.fit(x_train,y_train)
```

```
In [ ]: y_pred=svrg.predict(x_test)
        mean_squared_error(y_test,y_pred)
```

```
In [ ]: np.sqrt(mean_squared_error(y_test,y_pred))
```