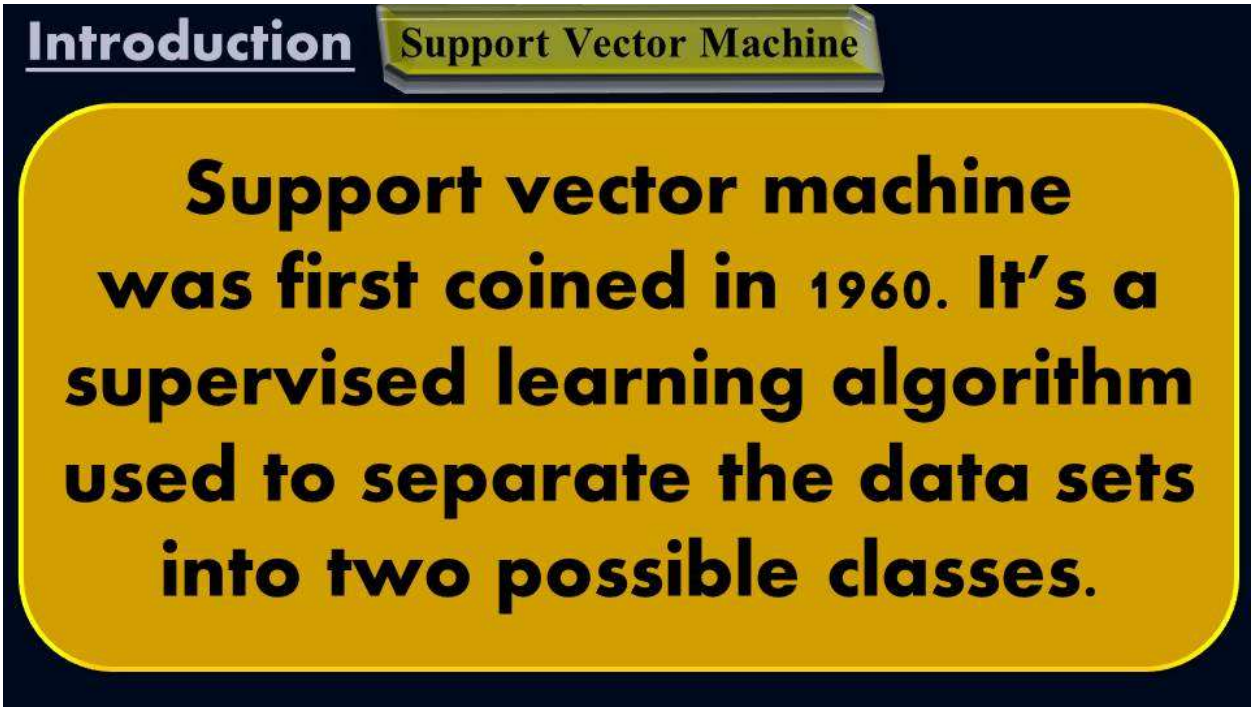
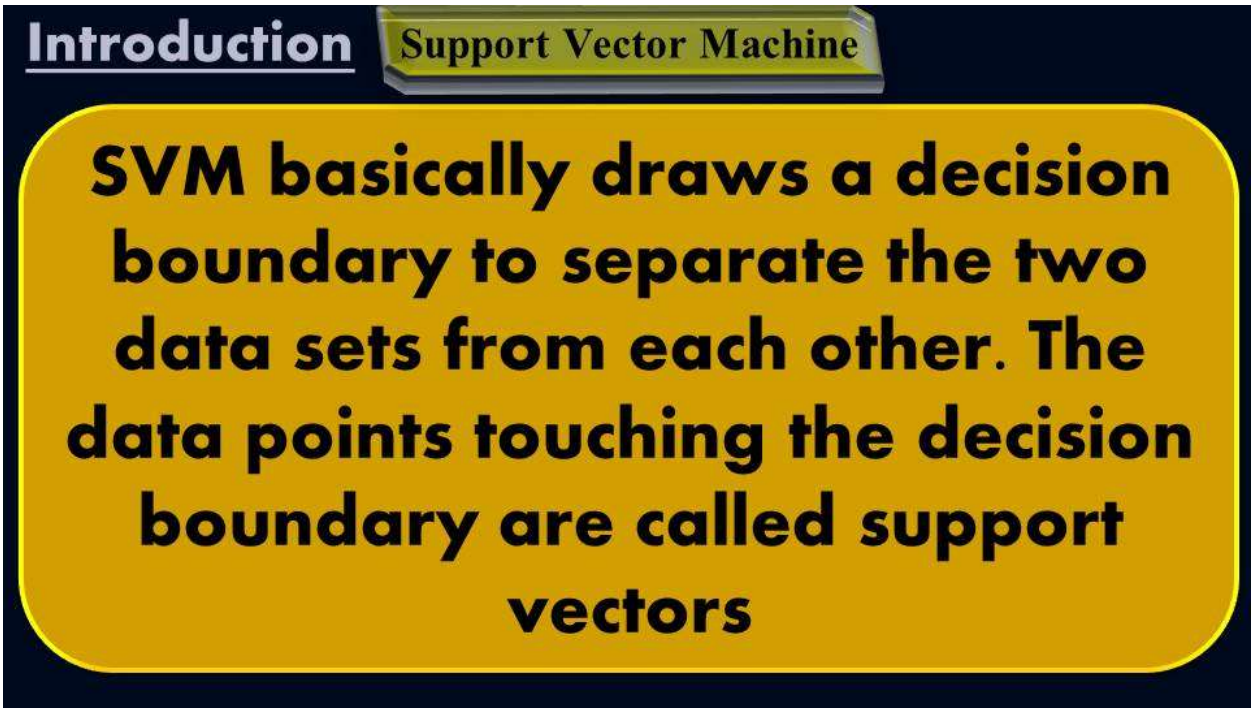
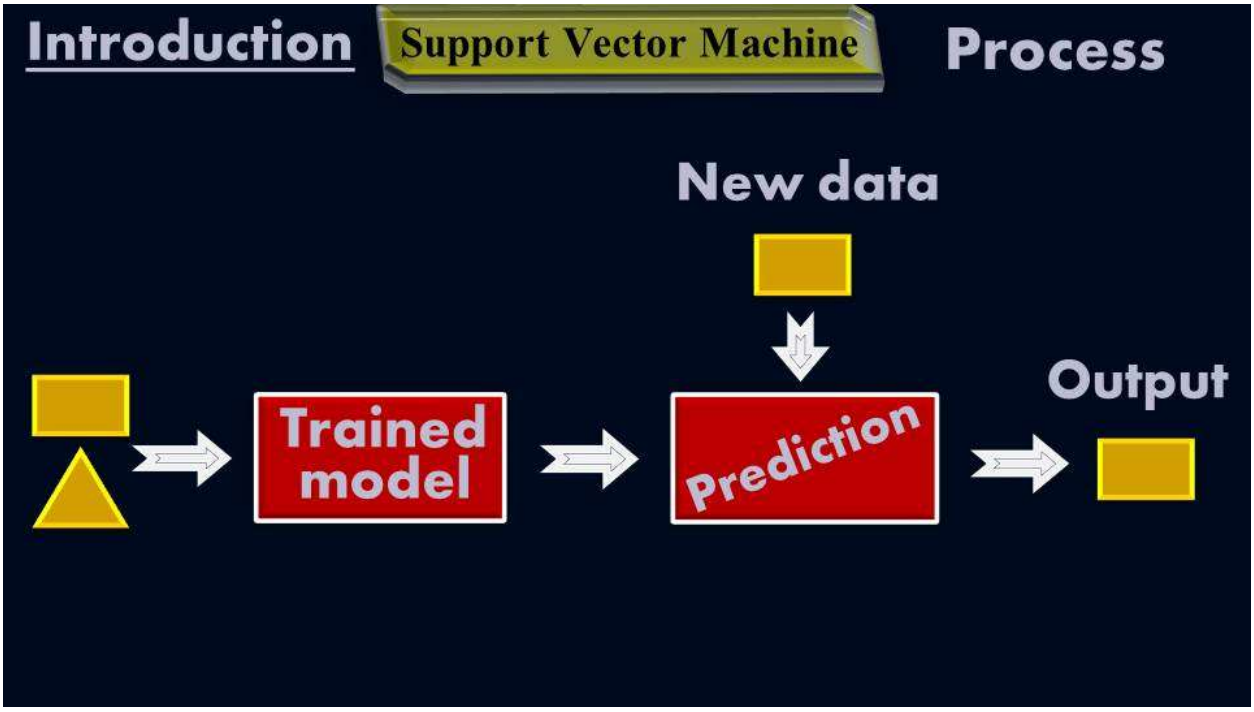
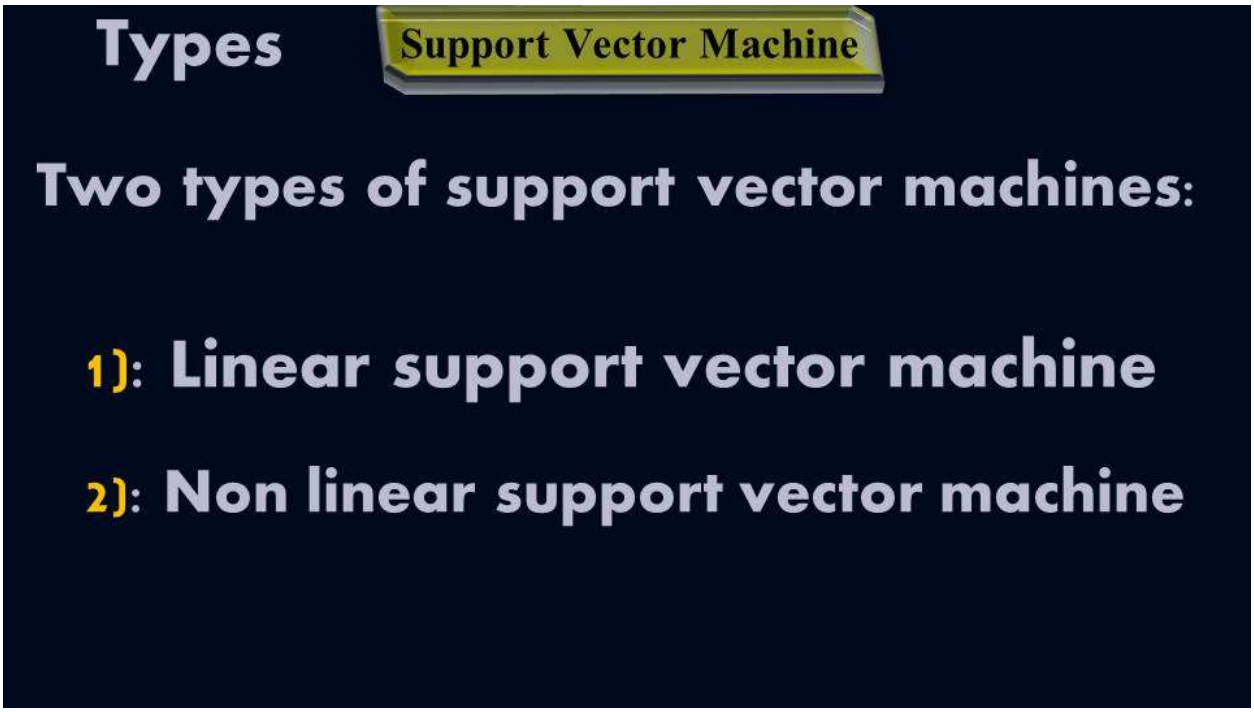
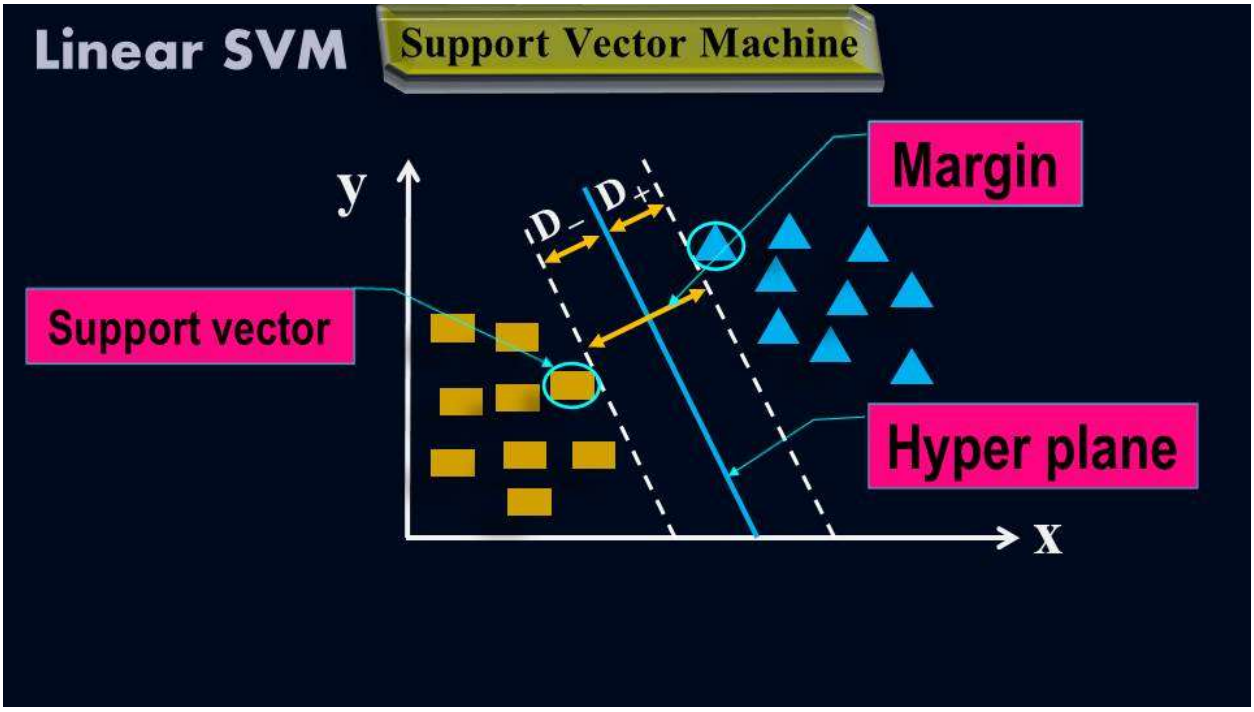
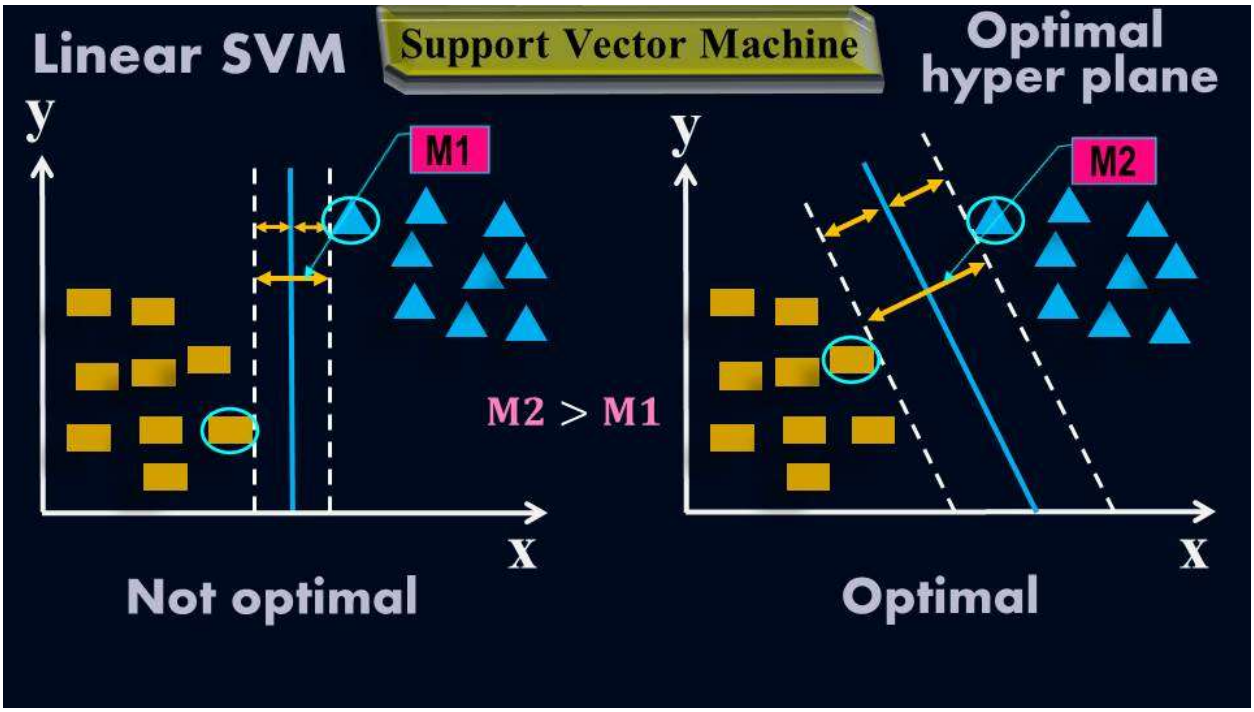

Introduction
Support Vector Machine
Support vector machine was first coined in 1960. It's a supervised learning algorithm used to separate the data sets into two possible classes.

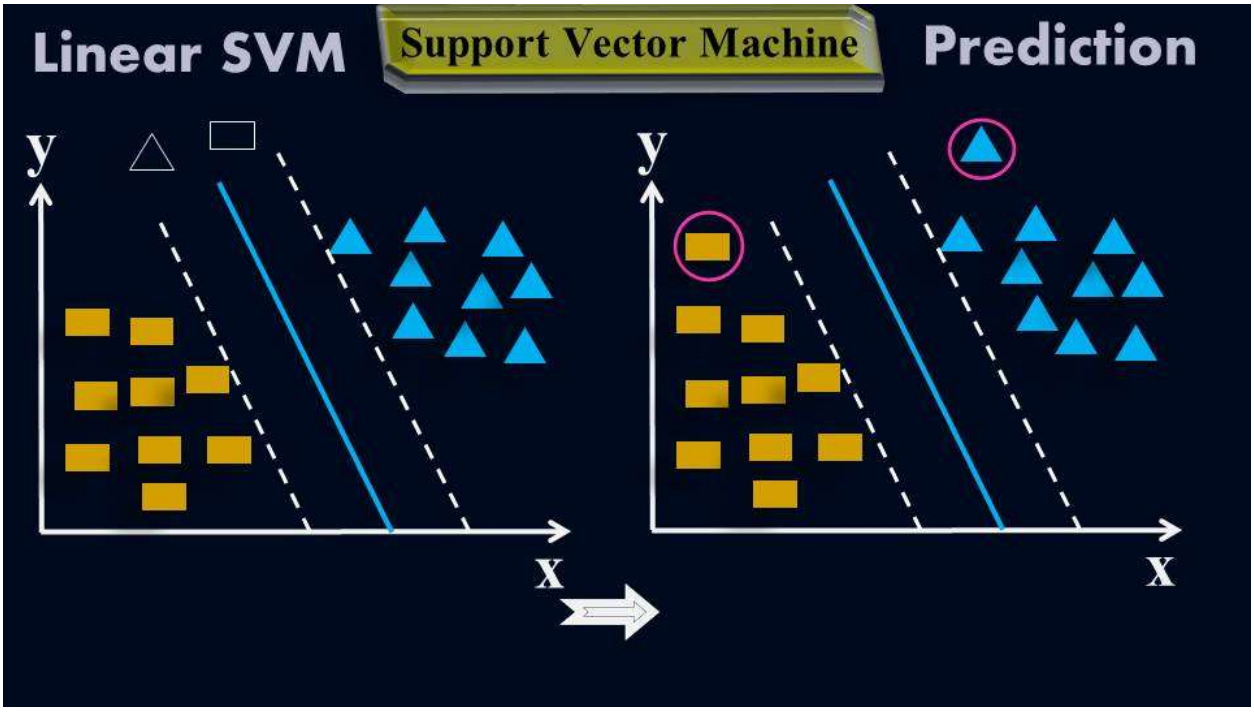

Introduction
Support Vector Machine
SVM basically draws a decision boundary to separate the two data sets from each other. The data points touching the decision boundary are called support vectors

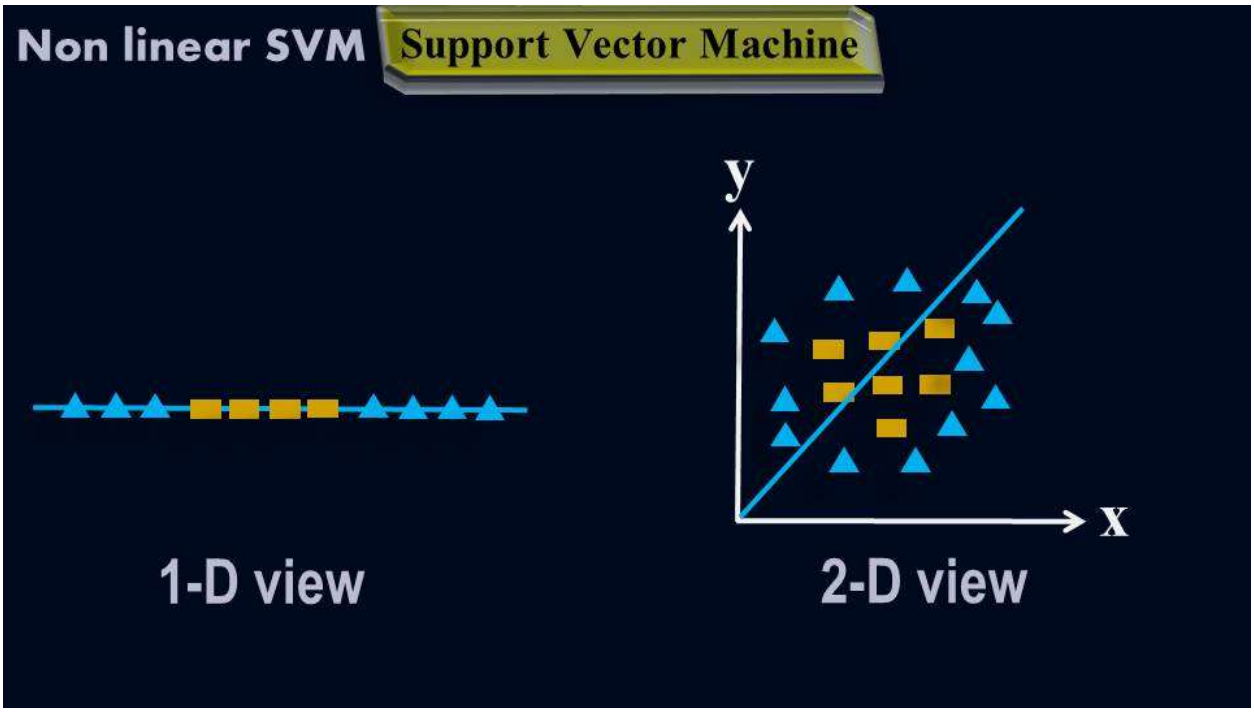

Introduction
Support Vector Machine
Process
New data
Trained model
Prediction
Output

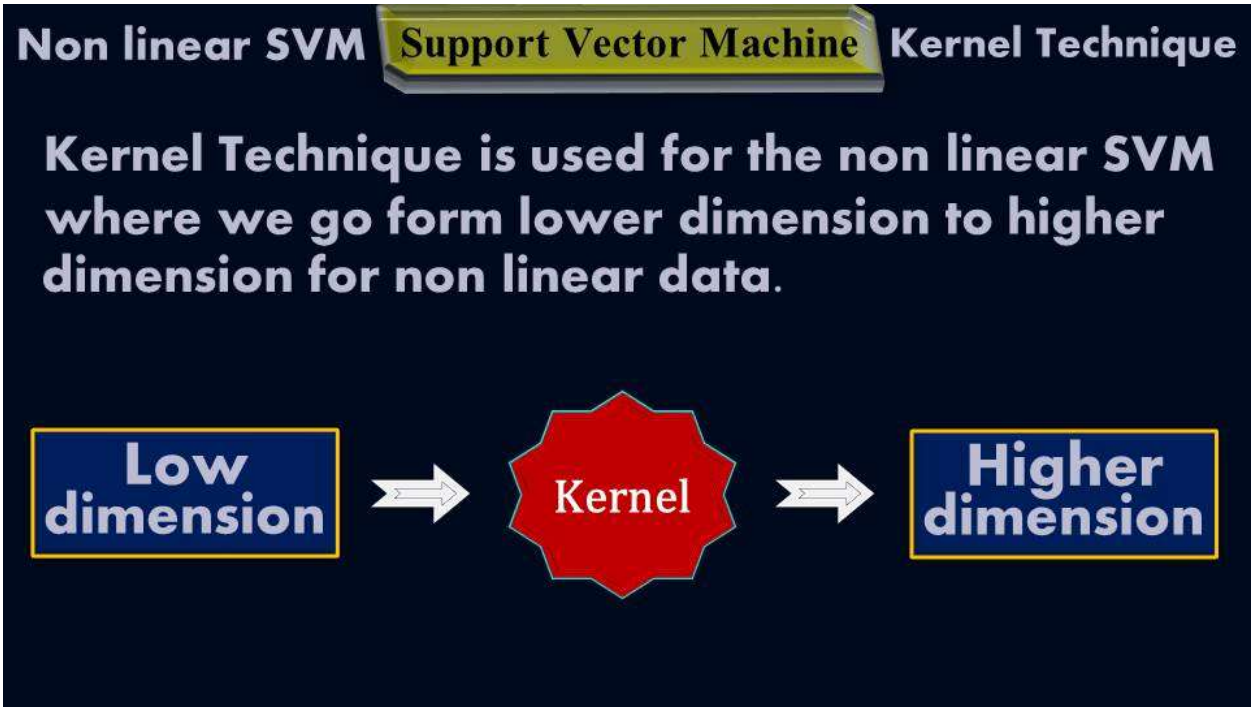

Types
Support Vector Machine
Two types of support vector machines:
1): Linear support vector machine
2): Non linear support vector machine

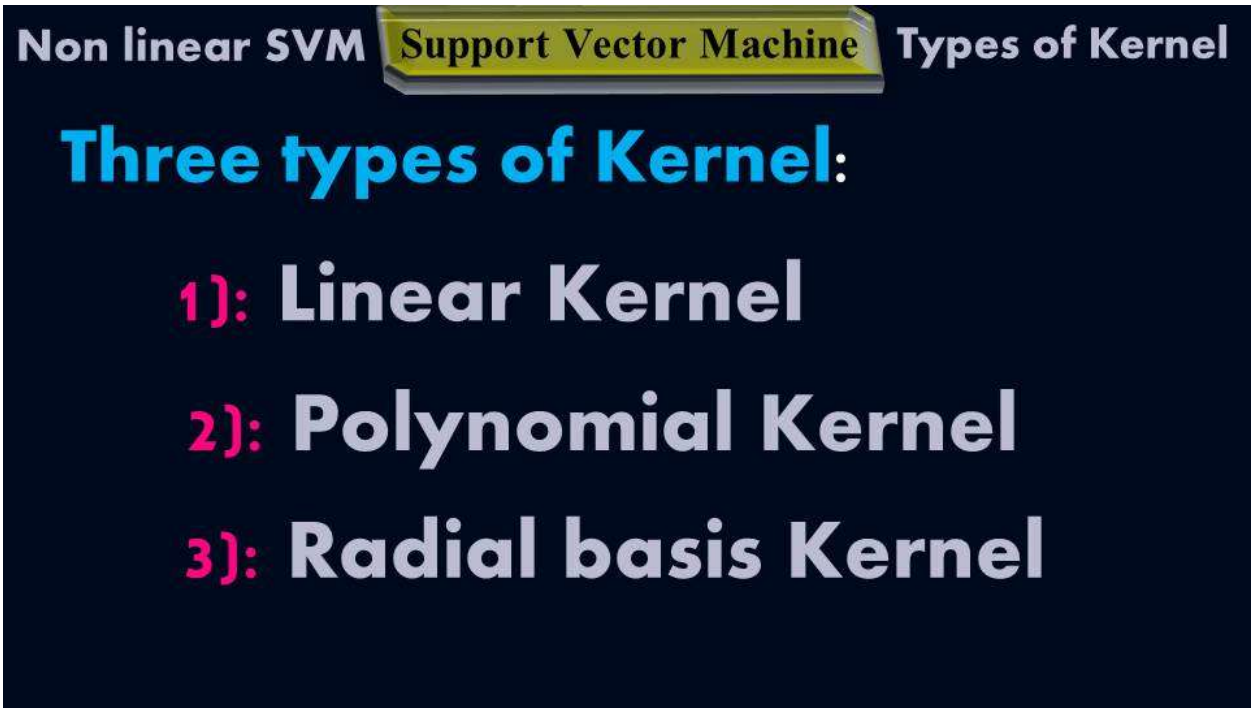

Linear SVM
Support Vector Machine
Margin
y
D−
D+
Support vector
Hyper plane
x

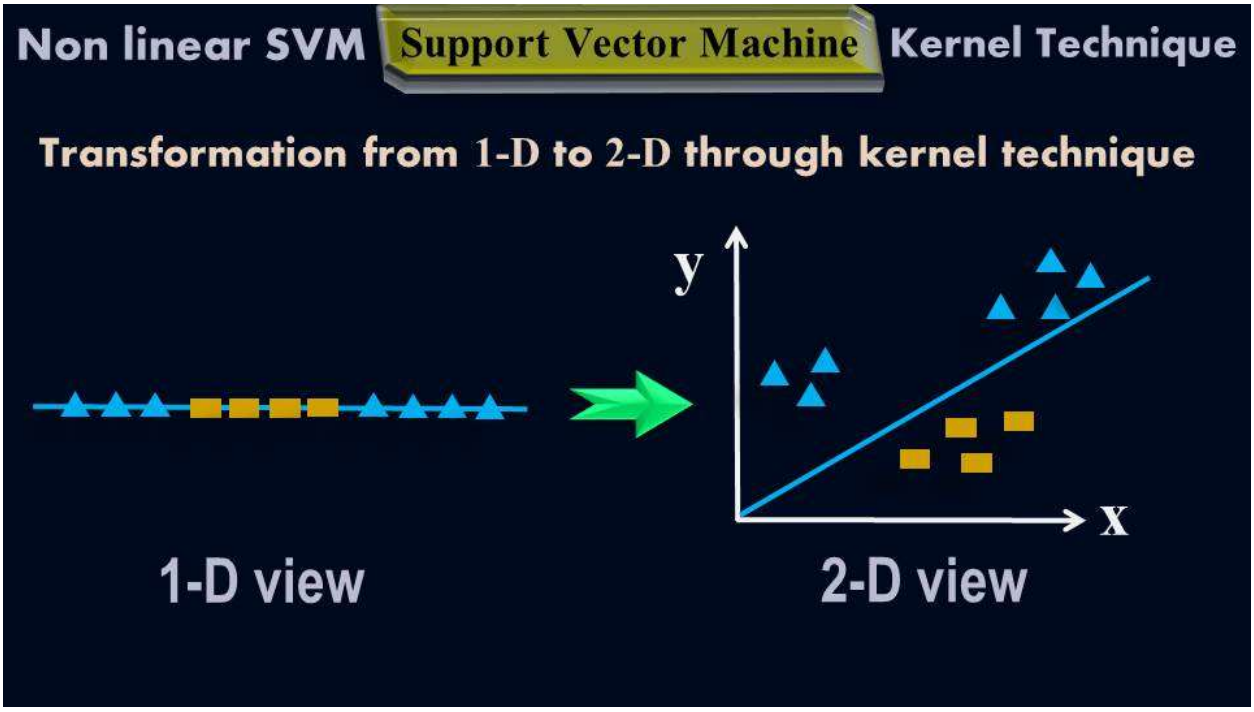

Linear SVM
Support Vector Machine
Optimal hyper plane
y
M1
M2 > M1
x
Not optimal
y
M2
x
Optimal

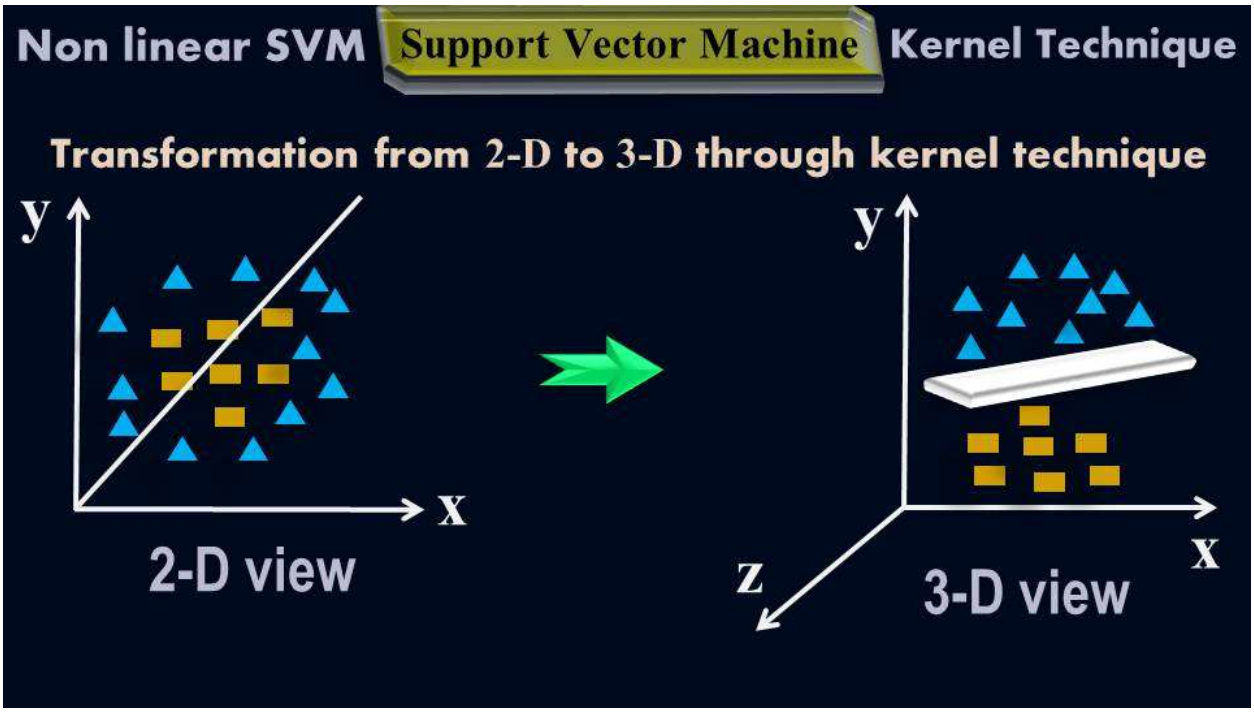

Linear SVM
Support Vector Machine
Prediction
y
x
y
x

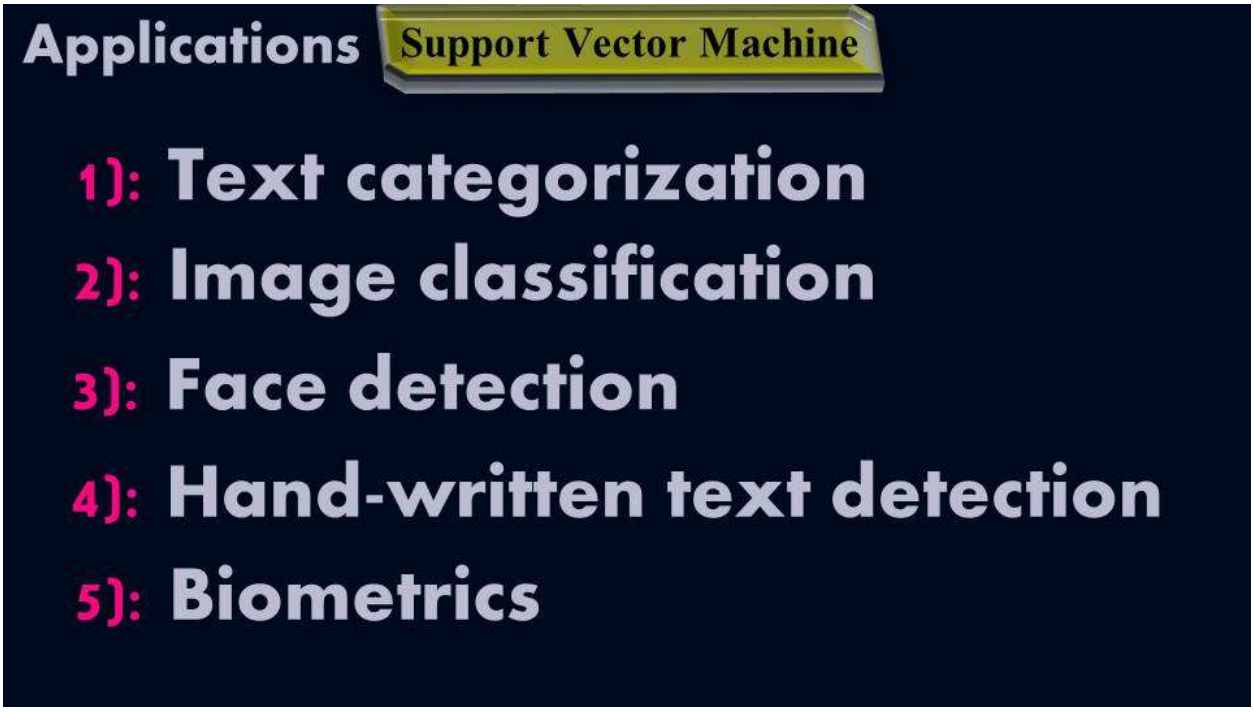

Non linear SVM
Support Vector Machine
y
x
1-D view
2-D view

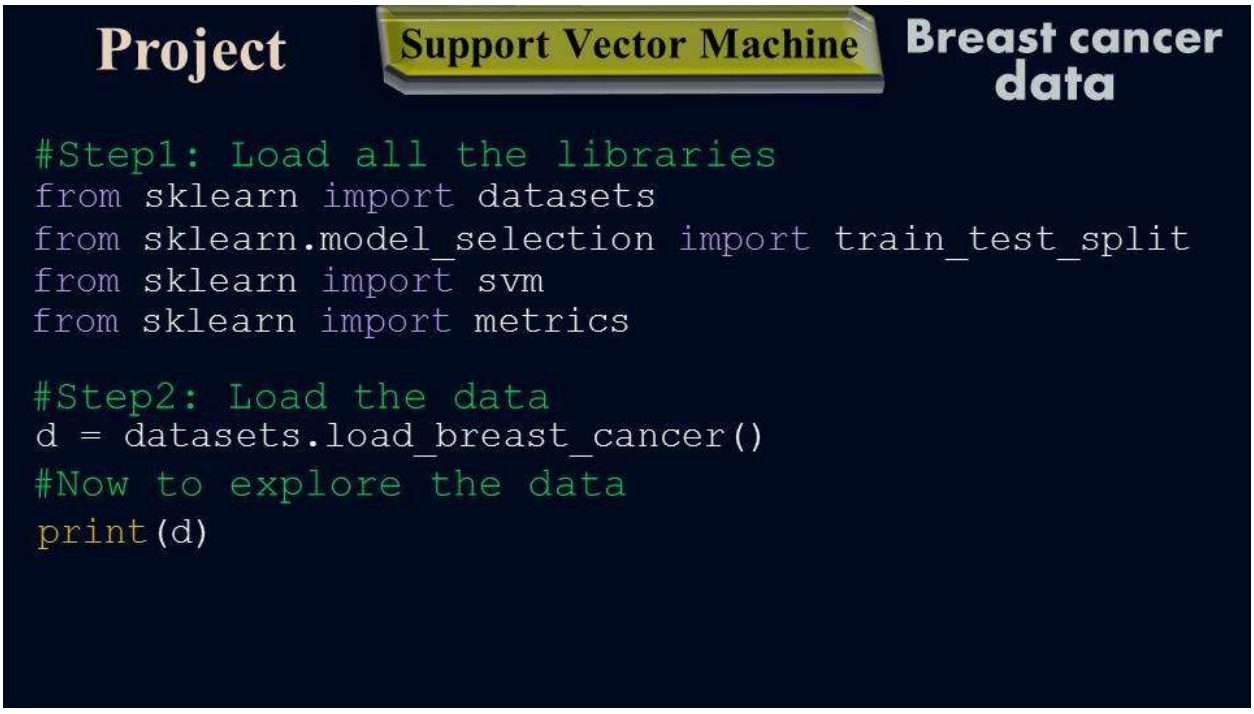

Non linear SVM
Support Vector Machine
Kernel Technique
Kernel Technique is used for the non linear SVM where we go form lower dimension to higher dimension for non linear data.
Low dimension
Kernel
Higher dimension

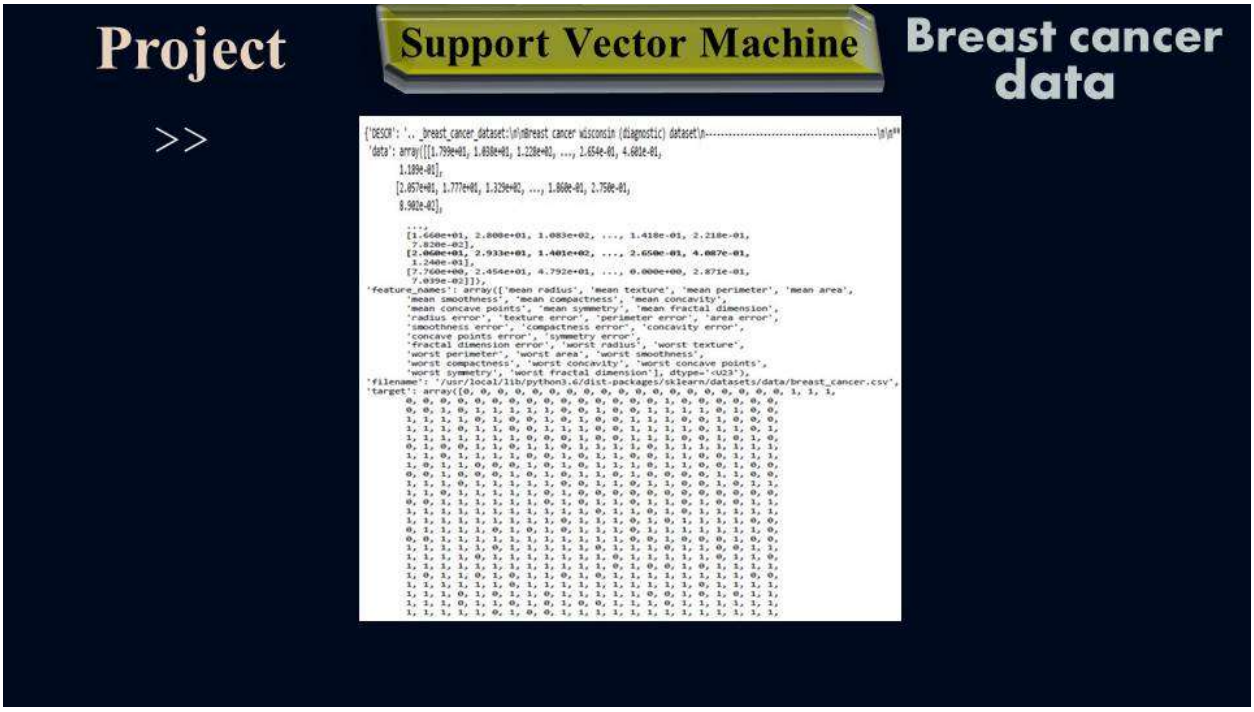

Non linear SVM
Support Vector Machine
Types of Kernel
Three types of Kernel:
1): Linear Kernel
2): Polynomial Kernel
3): Radial basis Kernel

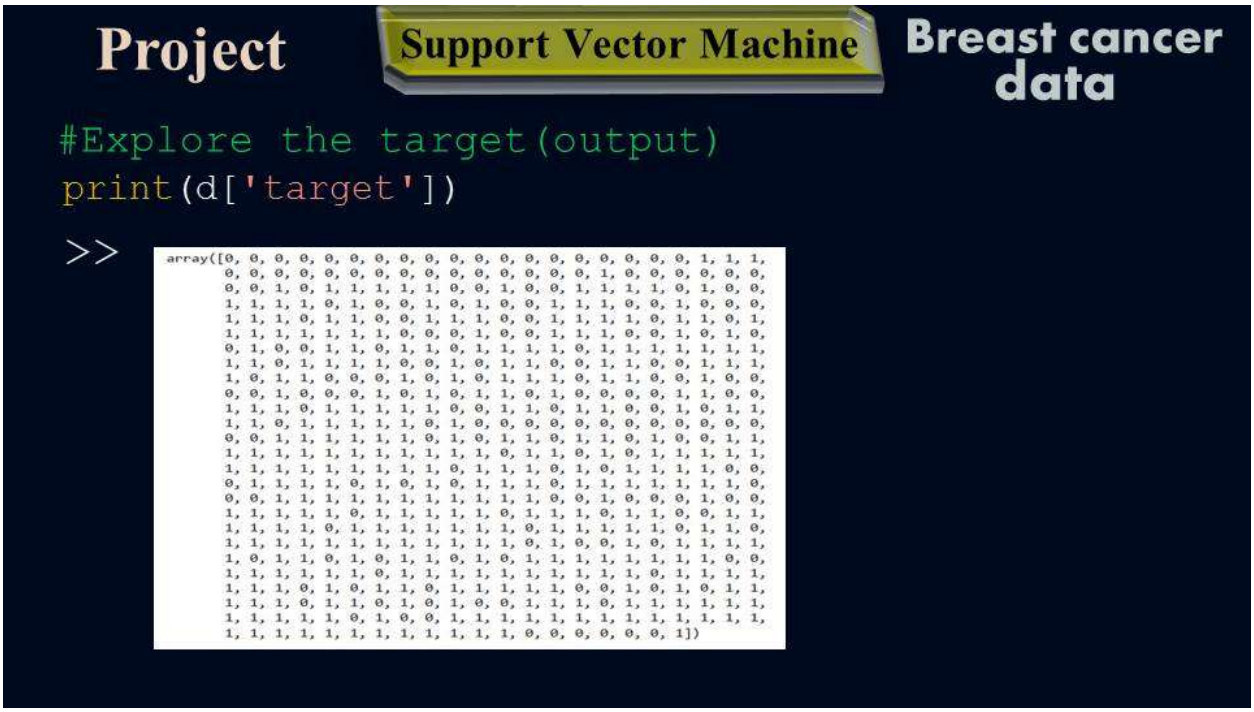

Non linear SVM
Support Vector Machine
Kernel Technique
Transformation from 1-D to 2-D through kernel technique
y
x
1-D view
2-D view


Non linear SVM
Support Vector Machine
Kernel Technique
Transformation from 2-D to 3-D through kernel technique
y
x
2-D view
y
x
z
3-D view


Applications
Support Vector Machine
1): Text categorization
2): Image classification
3): Face detection
4): Hand-written text detection
5): Biometrics


Project
Support Vector Machine
Breast cancer data
#Step1: Load all the libraries
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
#Step2: Load the data
d = datasets.load_breast_cancer()
#Now to explore the data
print(d)


Project
Support Vector Machine
Breast cancer data
>>


Project
Support Vector Machine
Breast cancer data
#Explore the target(output)
print(d['target'])
>>

Project **Support Vector Machine** **Breast cancer data**

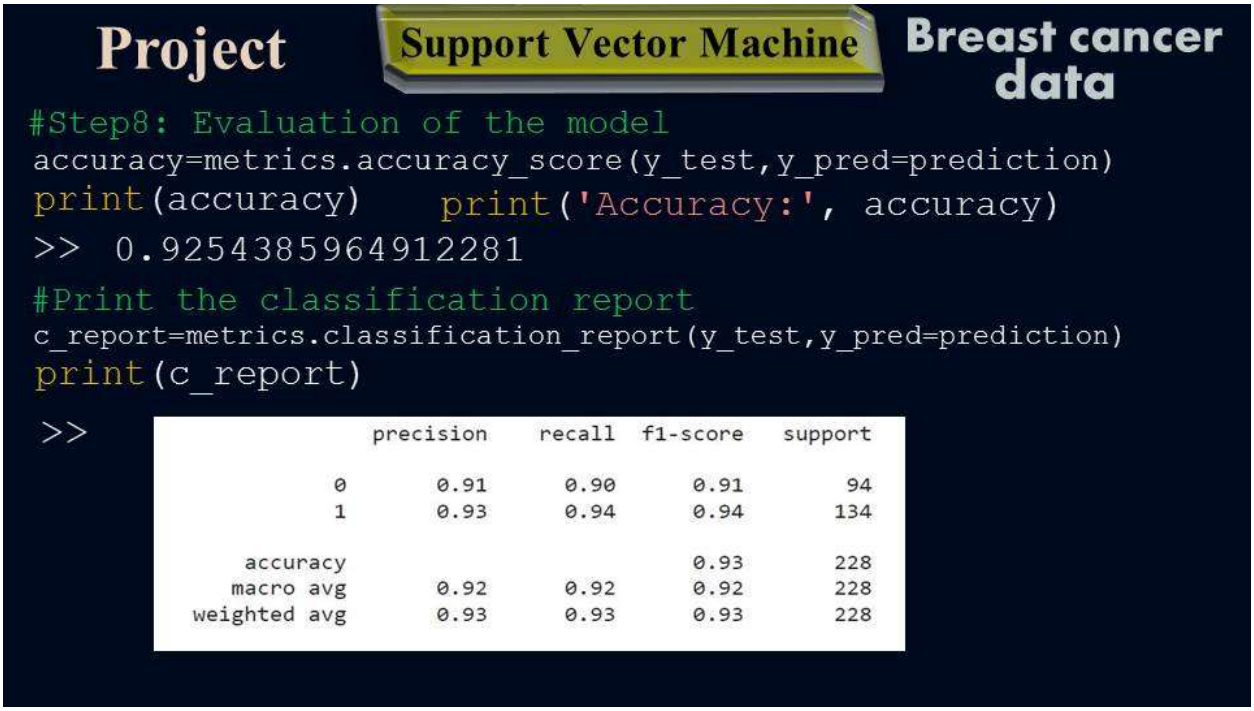
```
#Step3: Clean the data: Data is already clean
#Step4: Split the data into train and test
x_train, x_test, y_train,y_test=train_test_split(d.data,d.target,test_size=0.4,random_state=209)
#Step5: Create the machine model
model = svm.SVC(kernel='linear')
#Step6: Train the machine model using fit method
model.fit(x_train,y_train)
```

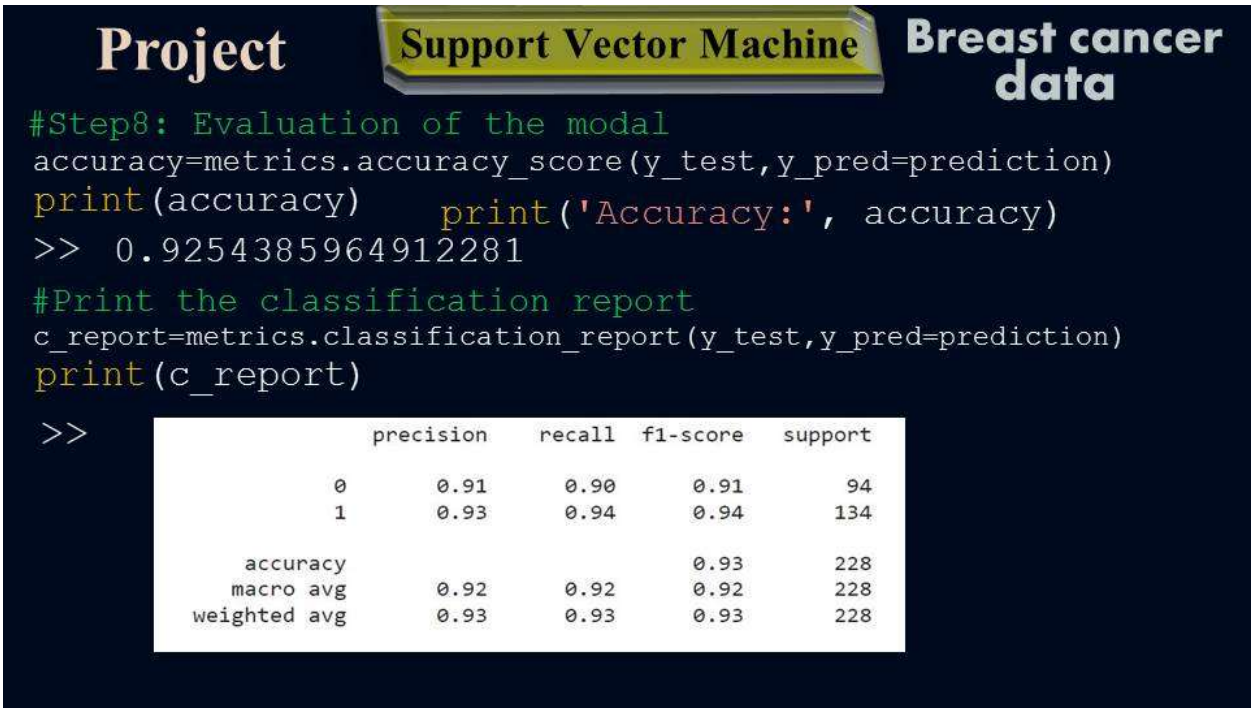
```
>> SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Project **Support Vector Machine** **Breast cancer data**

```
#Step7: Prediction of the model
prediction=model.predict(x_test)
print(prediction)
```

```
>> array([0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0,
       0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0,
       0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
       1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 1, 0])
```


Project
Support Vector Machine
Breast cancer data
#Step8: Evaluation of the model
accuracy=metrics.accuracy_score(y_test,y_pred=prediction)
print(accuracy)
print('Accuracy:', accuracy)
>> 0.9254385964912281
#Print the classification report
c_report=metrics.classification_report(y_test,y_pred=prediction)
print(c_report)
>>
precision recall f1-score support
0 0.91 0.90 0.91 94
1 0.93 0.94 0.94 134
accuracy 0.93 228
macro avg 0.92 0.92 0.92 228
weighted avg 0.93 0.93 0.93 228


Project
Support Vector Machine
Breast cancer data
#Step8: Evaluation of the modal
accuracy=metrics.accuracy_score(y_test,y_pred=prediction)
print(accuracy)
print('Accuracy:', accuracy)
>> 0.9254385964912281
#Print the classification report
c_report=metrics.classification_report(y_test,y_pred=prediction)
print(c_report)
>>
precision recall f1-score support
0 0.91 0.90 0.91 94
1 0.93 0.94 0.94 134
accuracy 0.93 228
macro avg 0.92 0.92 0.92 228
weighted avg 0.93 0.93 0.93 228

COMPLETE CODES ON ONE PAGE