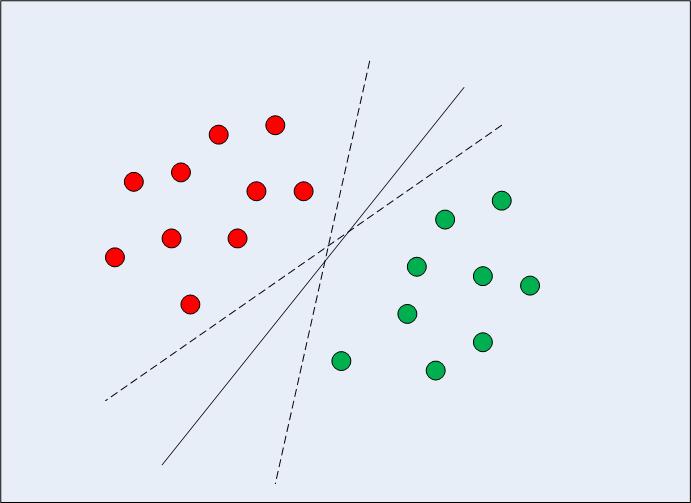
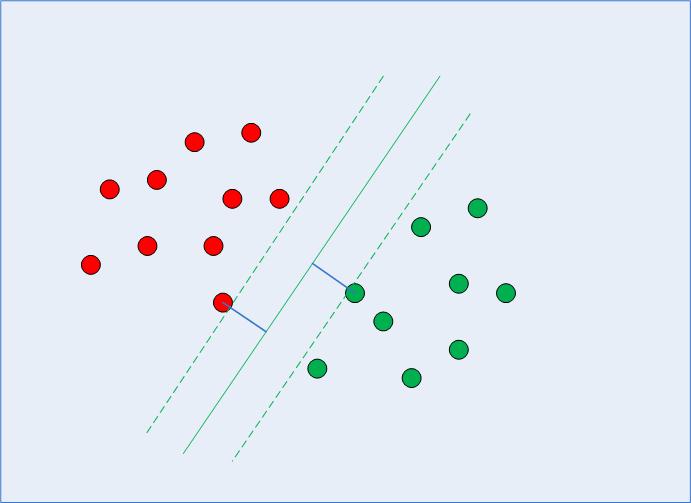
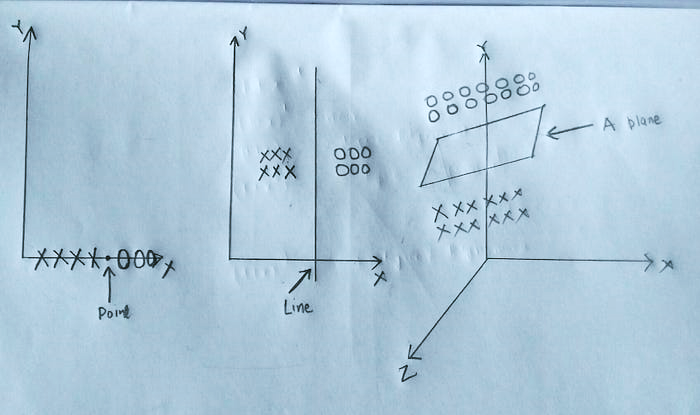
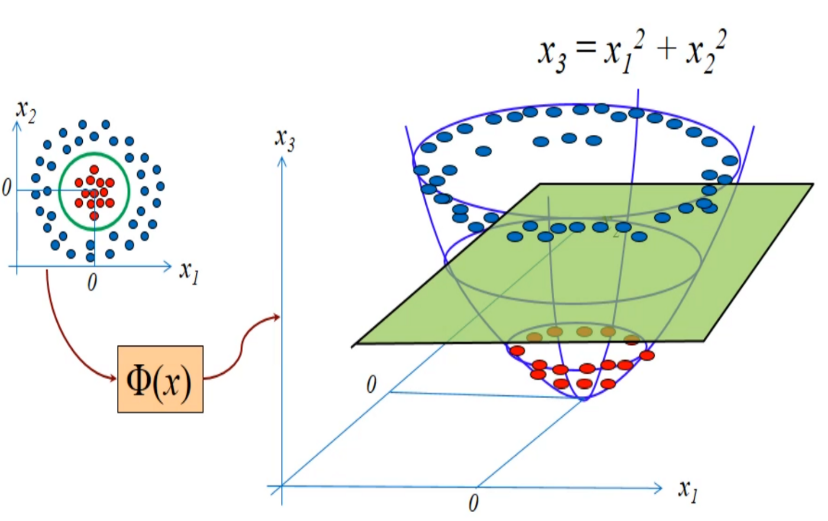
**Support Vector Machine**





**# Implementing SVM with Scikit-Learn**

import pandas as pd

bankdata = pd.read\_csv("/content/bill\_authentication.csv")

X = bankdata.drop('Class', axis=1)

y = bankdata['Class']

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=42)

from sklearn.svm import SVC

svclassifier = SVC(kernel='linear')

svclassifier.fit(X\_train, y\_train)

y\_pred = svclassifier.predict(X\_test)

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

print(accuracy\_score(y\_test,y\_pred))

print(confusion\_matrix(y\_test,y\_pred))

print(classification\_report(y\_test,y\_pred))

**#Decision tree algorithm**

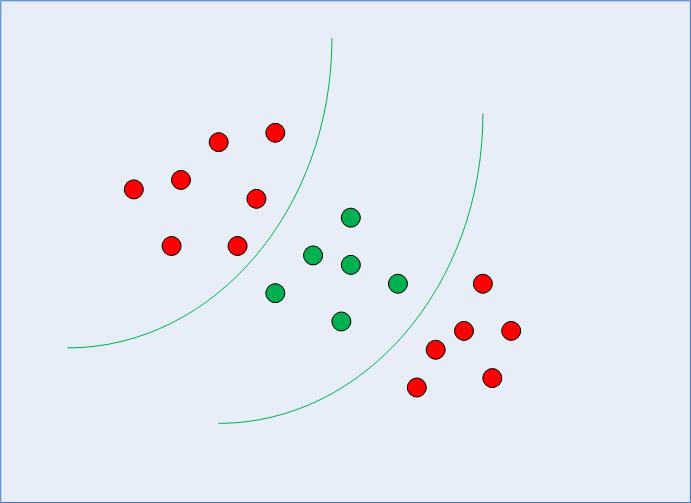
from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier()

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

**#Kernel SVM**



import numpy as np

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

***# Assign column names to the dataset***

colnames = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

***# Read dataset to pandas dataframe***

irisdata = pd.read\_csv(url, names=colnames)

X = irisdata.drop('Class', axis=1)

y = irisdata['Class']

***#Train Test Split***

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)

***#1. Polynomial Kernel***

from sklearn.svm import SVC

svclassifier = SVC(kernel='poly', degree=8)

svclassifier.fit(X\_train, y\_train)

***#Making Predictions***

y\_pred = svclassifier.predict(X\_test)

***#Evaluating the Algorithm***

from sklearn.metrics import classification\_report, confusion\_matrix print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

***#2. Gaussian Kernel***

from sklearn.svm import SVC

svclassifier = SVC(kernel='rbf')

svclassifier.fit(X\_train, y\_train)

***#Prediction and Evaluation***

y\_pred = svclassifier.predict(X\_test)

from sklearn.metrics import classification\_report, confusion\_matrix

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

***#3. Sigmoid Kernel***

from sklearn.svm import SVC

svclassifier = SVC(kernel='sigmoid')

svclassifier.fit(X\_train, y\_train)

***#Prediction and Evaluation***

…

**Train SVM classifier using sklearn digts dataset (e. from sklearn.datasets import load\_digits) and then**,

1. Measure accuracy of your model using different kernels such as rbf,poly and linear.

2. Tune your model further using regularization and gamma parameters and try to come up with highest accuracy score.

3. Use 80% of samples as training data size.