SVM

- Support vector machine is nothing but supervised machine learning classification techinique.
- It is helps to classify the classes with the help of the hyperplan.
- The hyperplan is the nothing a line bisect the two classed or more classes with minimum squared error.
- In the surrounding of the hyperplan we having the margine which is equidistantly plan on both side of the plan.
- Whichever datapoint goes through the that margin we suppose to called as vector them.
- By using the that vector we can classify the two classes with least errors.

- But not always we find the claases those can bisect very easily by using straight line.
- Suppose if we have the polynominal or the circle datapoint that a simple straiht line will not properly.
- So to get the best classification we use the kernal tricks.

- Kernal tricks help the alorthm to classify the such complex distribution of the data by enhancing the dimensions of the data.

- Let's us consider we havin the liner circle type of the data to classify that we have to use the kernal technique which add the dimensinality of the model and help to classify the datapoint with least squared error.

- **Kernal Type:-**
  - Linear Kernal.
  - Polynomial Kernal.
  - RBF Kernal.
  - sigmoid kernal.

At the polynomial degree of the polynomial kernal works as linear kernal.

In [70]:
```
import numpy as np
import pandas as pd
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [71]:
```
from sklearn.metrics import classification_report,confusion_matrix,mean_squared_error,accuracy_score,recall_sc
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import datasets
```

In [72]:
```
cancer= datasets.load_iris()
```

In [73]:
```
cancer.target_names
```

Out[73]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [74]:
```
cancer.feature_names
```

Out[74]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [75]:
```
x = cancer.data
y = cancer.target
```

In [76]:
```
x.shape,y.shape
```

Out[76]:
```
((150, 4), (150,))
```

In [77]:
```
scaler = StandardScaler()
x = scaler.fit_transform(x)
```

In [78]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size=0.80,random_state=42,stratify=y)
```

In [79]:
```
clf = SVC(kernel='linear')
clf.fit(x_train,y_train)
y_pred=clf.predict(x_test)
print('Accuarcy :',accuracy_score(y_test,y_pred))
```

```
Accuarcy : 1.0
```

In [83]:
```
print("Precision Score : ",precision_score(y_test, y_pred,
                                           pos_label='positive',
                                           average='micro'))
print("Precision Score : ",recall_score(y_test, y_pred,
                                           pos_label='positive',
                                           average='micro'))
```

```
Precision Score :  1.0
Precision Score :  1.0
```

In [ ]:
```

```