

Register No: 21MIS1044

Name: Srinivasan JP

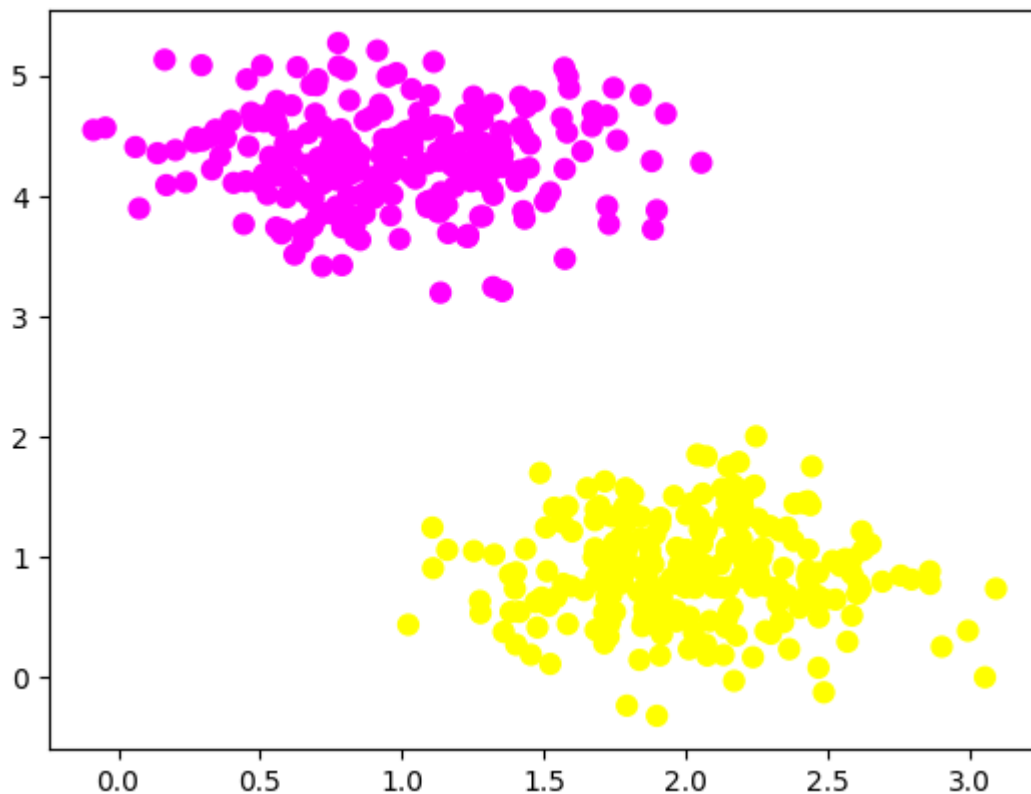
## Support Vector Mechanism

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
#imports the required modules
```

```
In [ ]: # importing scikit learn with make_blobs

# creating datasets X containing n_samples
# Y containing two classes
X, Y = make_blobs(n_samples=500, centers=2,
                  random_state=0, cluster_std=0.40)

import matplotlib.pyplot as plt
# plotting scatters
plt.scatter(X[:, 0], X[:, 1], c=Y, s=50, cmap='spring');
plt.show()
```

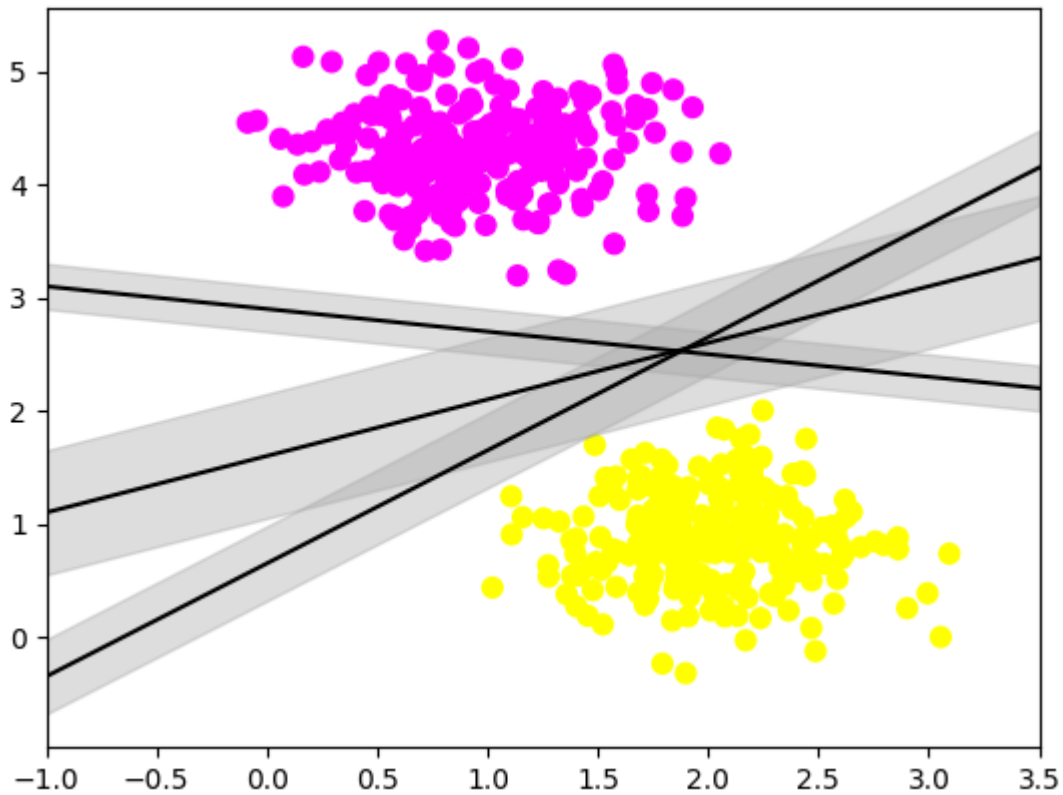


```
In [ ]: # creating linspace between -1 to 3.5
xfit = np.linspace(-1, 3.5)

# plotting scatter
plt.scatter(X[:, 0], X[:, 1], c=Y, s=50, cmap='spring')
```

```
# plot a line between the different sets of data
for m, b, d in [(1, 0.65, 0.33), (0.5, 1.6, 0.55), (-0.2, 2.9, 0.2)]:
    yfit = m * xfit + b
    plt.plot(xfit, yfit, '-k')
    plt.fill_between(xfit, yfit - d, yfit + d, edgecolor='none',
                    color='#AAAAAA', alpha=0.4)

plt.xlim(-1, 3.5)
plt.show()
```



Importing the dataset and converting the Categorical data to 0 and 1

```
In [ ]: x = pd.read_csv("breast+cancer+wisconsin+diagnostic\\wdbc.csv")
y = x.iloc[:, 1].map({'M': 0, 'B': 1}) # Mapping 'M' to 0 and 'B' to 1

# Extracting two features
x = x.iloc[:, [4, 5]].values

# 569 samples and 2 features
x.shape

print (x),(y)

[[ 132.9  1326. ]
 [ 130.   1203. ]
 [  77.58  386.1 ]
 ...
 [ 108.3   858.1 ]
 [ 140.1  1265. ]
 [  47.92   181. ]]
```

```
Out[ ]: (None,
0      0
1      0
2      0
3      0
4      0
..
563    0
564    0
565    0
566    0
567    1
Name: M, Length: 568, dtype: int64)
```

Using the SKLearn's prebuild algorithms the model have been fit

```
In [ ]: # import support vector classifier
# "Support Vector Classifier"
from sklearn.svm import SVC
clf = SVC(kernel='linear')

# fitting x samples and y classes
clf.fit(x, y)
```

```
Out[ ]: SVC
SVC(kernel='linear')
```

Predicting the value which is belongs positive or negative

```
In [ ]: def getName(value):
        print("Predicted category is :",end=" ")
        if(value==1):
            print("malignant")
        else:
            print("benign")
```

```
In [ ]: getName(clf.predict([[120, 590]]))
getName(clf.predict([[85, 550]]))
```

```
Predicted category is : benign
Predicted category is : malignant
```