# Support Vector Machines

1. Consider following two featured dataset.

```
X1  = [1, 5, 1.5, 8, 1, 9]
X2  = [2, 8, 1.8, 8, 0.6, 11]
```

Plot the dataset using scatter plot.
Create a matrix with above two features.

| x1 | x2 | y |
|---|---|---|
| 1 | 2 | 0 |
| 5 | 8 | 1 |
| 1.5 | 1.8 | 0 |
| 8 | 8 | 1 |
| 1 | 0.6 | 0 |
| 9 | 11 | 1 |

Here y is class label

Define svm classifier as follows

```
clf = svm.SVC(kernel='linear', C = 1.0)
```

In order to call this function, you need to import svm from sklearn

*from sklearn import svm*

Here kernel is linear function, and C is equal to 1.0. C is a valuation of "how badly" you want to properly classify, or fit, everything. There exist many debates about the value of C, as well as how to calculate the value for C. We're going to just stick with 1.0 for now, which is a nice default parameter.

Next, we call:

```
clf.fit(X,y)
```

For predicting label of a sample (*0.58,0.76)*

```
print(clf.predict([0.58,0.76]))

print(clf.predict([10.58,10.76]))
```

2. Now, to visualize your data use the following code:

```
w = clf.coef_[0]
print(w)
a = -w[0] / w[1]

xx = np.linspace(0,12)
yy = a * xx - clf.intercept_[0] / w[1]

h0 = plt.plot(xx, yy, 'k-', label="non weighted div")
plt.scatter(X[:, 0], X[:, 1], c = y)
plt.show()
```

3. Following is a simple python code that uses inbuilt function for classification using the concept of Support vector Machine

```
# Support Vector Machine
from sklearn import datasets
from sklearn import metrics
from sklearn.svm import SVC
# load the iris datasets
dataset = datasets.load_iris()
# fit a SVM model to the data
model = SVC()
model.fit(dataset.data, dataset.target)
print(model)
# make predictions
expected = dataset.target
predicted = model.predict(dataset.data)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

Execute the code given to you and understand the working of svm classifier.

4. Compare SVM with kNN and Logistic Regression.

5. In SVC function you may change the parameter C and plot the accuracy for different values of C with a kernel - radial basis function (RBF) and polynomial kernel with degree 3.

```
svm.SVC(kernel='rbf', gamma=0.7, C=C),
svm.SVC(kernel='poly', degree=3, C=C))
```

6. Does the accuracy change if the kernel function changes?