Subject : AML

Subject Code : 3CS1111

Roll No : 20MCED08

KNN CLASSIFIER

In [1]:
```
from sklearn import datasets, metrics
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt
```

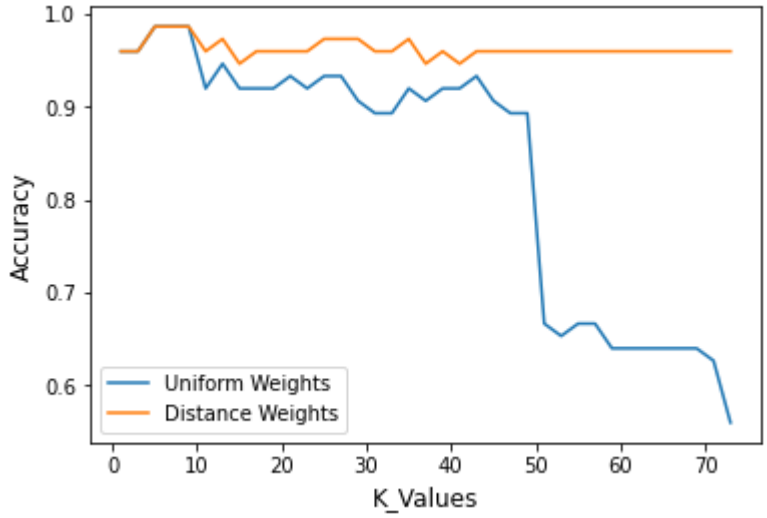
In [2]:
```
#Loading the Iris Dataset
X,y = datasets.load_iris(return_X_y=True)
```

In [3]:
```
X_train=X[range(0,150,2),:]
y_train=y[range(0,150,2)]
X_test=X[range(1,150,2),:]
y_test=y[range(1,150,2)]
```

In [4]:
```
n=[]
acc = []
for i in range(1,75,2):
    n.append(i)
    n_neighbors=i
    weights='uniform' #or weights='distance'
    #creating instance of a classifier
    clf = KNeighborsClassifier(n_neighbors, weights='uniform')
    # other argument is metric (default is 'minkowski'. There are many other p
ossibilities including 'euclidean')
    #Other useful argument is p which controls the power of minkowski distanc
e. Default is 2
    clf.fit(X_train, y_train)
    prediction = clf.predict(X_test) #predict using the learnt classifier
    acc.append(metrics.accuracy_score(y_test, prediction, normalize=True))
```

In [5]:
```
n1=[]
acc1 = []
for i in range(1,75,2):
    n1.append(i)
    n_neighbors=i
    weights='distance' #or weights='distance'
    #creating instance of a classifier
    clf = KNeighborsClassifier(n_neighbors, weights='distance')
    clf.fit(X_train, y_train)
    prediction = clf.predict(X_test) #predict using the learnt classifier
    acc1.append(metrics.accuracy_score(y_test, prediction, normalize=True))
```

In [6]:
```
plt.figure()
plt.plot(n, acc, label='Uniform Weights')
plt.plot(n1, acc1, label='Distance Weights')
plt.xlabel('K_Values', fontsize=12)
plt.ylabel('Accuracy', fontsize=12)
plt.legend()
plt.show()
```



In [ ]:
```

```