1 Nearest Neighbor

1.1 Describe Implementation

The general idea is to compute the L2 Norm to the nearest neighbor, K point, and return the most recurring label in those neighbors.

1.1.1 KNN_TEST

This function genrates a label based on the points given to it, computes the distance, to each of those points (training points) and puts them in a list. This list is then sorted based on the distance that was calculated before. The subset of the K nearest neighbor or point will have their labels summed and then we will output the sign of the summation as the label, so if our predicted label is less than 0 we return -1 and if it is greater than 0 we return 1. In case the predicted value is 0 we will choose the result based on the ordering of the training data and from the set of neighbors in the sorted list.

```python
def KNN_test(X_train, Y_train, X_test, Y_test,K):
  NumberOfCorrect = 0
  for i, j in enumerate(X_test):
    radius = []
    for n, m in enumerate(X_train):
      radius.append(((math.pow((m[0] - j[0]), 2) + math.pow((m[1] - j[1]), 2)), Y_train[n
  ][0]))
  List_sorted = sorted(radius, key=lambda member: member[0], reverse=False)[:K]
  predictedvalue = sum([member[1] for member in List_sorted])
  if (not predictedvalue):
    predictedvalue = List_sorted[0][1]
  elif (predictedvalue < 0):
    predictedvalue = -1
  elif (predictedvalue > 0):
    predictedvalue = 1
  if predictedvalue == Y_test[i][0]:
    NumberOfCorrect = NumberOfCorrect+1
  return NumberOfCorrect / len(X_test)

```

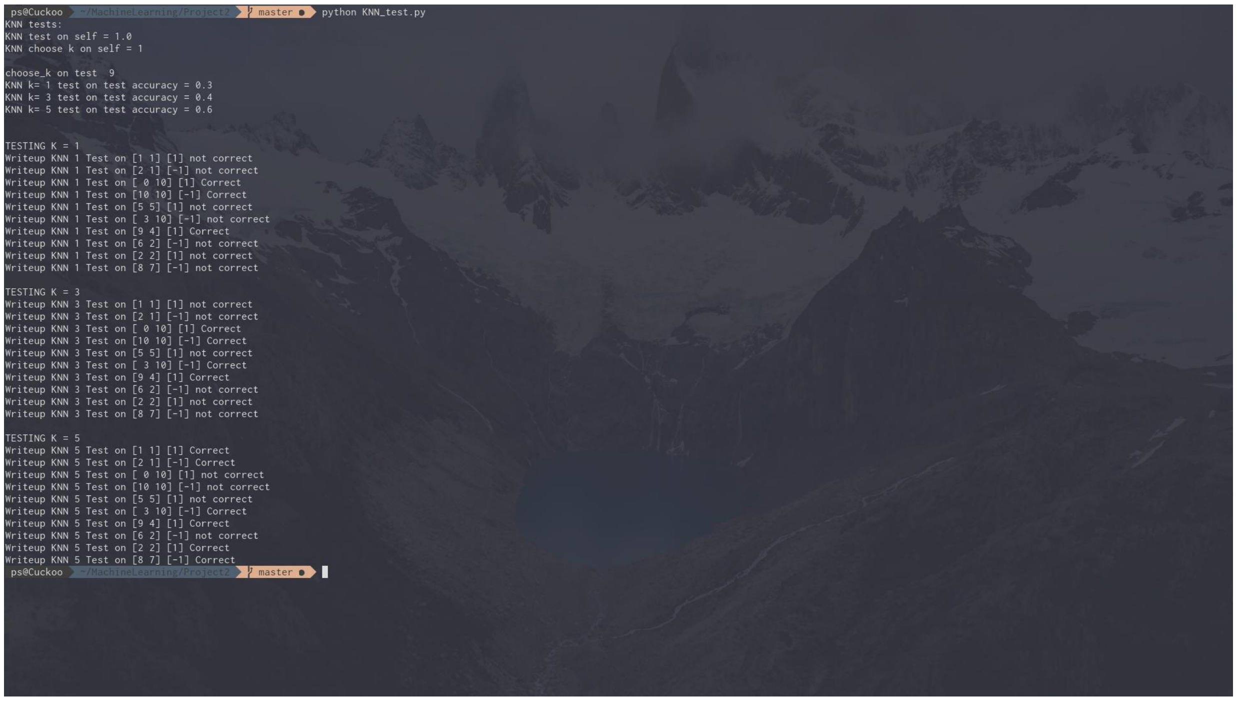
1.1.2 Choose_K

This function is simply checking all possible accuracies on the given test data and returns the index with the greatest accuracy which will be our K.

```python
  OldAccuracy = 0
  NewAccuracy = 0
  Predicted_K = 1
  K = len(X_train)
  for i in range(1, K):
    OldAccuracy = KNN_test(X_train,Y_train,X_val,Y_val, i)
    if(OldAccuracy>NewAccuracy):
      NewAccuracy = OldAccuracy
      Predicted_K = i
  return Predicted_K

```

1.2 Algorithm Classification

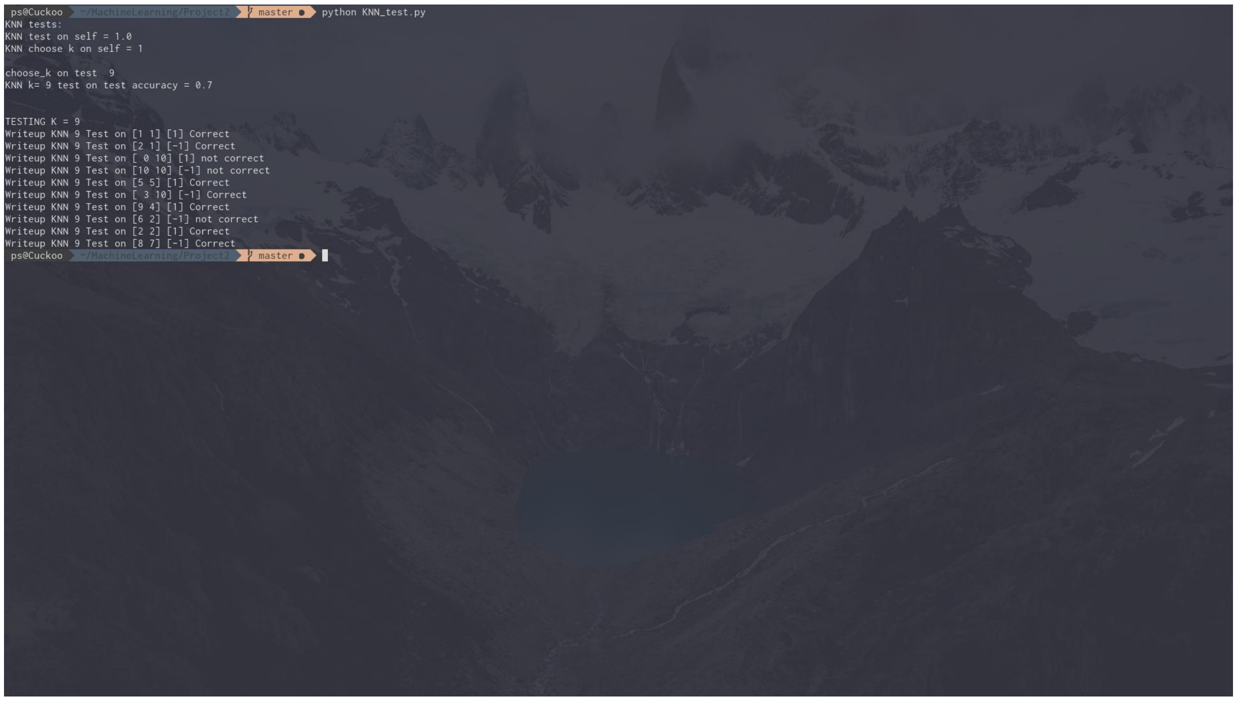
Using the testing data given to us from the project description we will get the following results



```
ps@Cuckoo  ~/MachineLearning/Project2  master ●  python KNN_test.py
KNN tests:
KNN test on self = 1.0
KNN choose k on self = 1

choose_k on test  9
KNN k= 1 test on test accuracy = 0.3
KNN k= 3 test on test accuracy = 0.4
KNN k= 5 test on test accuracy = 0.6


TESTING K = 1
Writeup KNN 1 Test on [1 1] [1] not correct
Writeup KNN 1 Test on [2 1] [-1] not correct
Writeup KNN 1 Test on [ 0 10] [1] Correct
Writeup KNN 1 Test on [10 10] [-1] Correct
Writeup KNN 1 Test on [5 5] [1] not correct
Writeup KNN 1 Test on [ 3 10] [-1] not correct
Writeup KNN 1 Test on [9 4] [1] Correct
Writeup KNN 1 Test on [6 2] [-1] not correct
Writeup KNN 1 Test on [2 2] [1] not correct
Writeup KNN 1 Test on [8 7] [-1] not correct

TESTING K = 3
Writeup KNN 3 Test on [1 1] [1] not correct
Writeup KNN 3 Test on [2 1] [-1] not correct
Writeup KNN 3 Test on [ 0 10] [1] Correct
Writeup KNN 3 Test on [10 10] [-1] Correct
Writeup KNN 3 Test on [5 5] [1] not correct
Writeup KNN 3 Test on [ 3 10] [-1] Correct
Writeup KNN 3 Test on [9 4] [1] Correct
Writeup KNN 3 Test on [6 2] [-1] not correct
Writeup KNN 3 Test on [2 2] [1] not correct
Writeup KNN 3 Test on [8 7] [-1] not correct

TESTING K = 5
Writeup KNN 5 Test on [1 1] [1] Correct
Writeup KNN 5 Test on [2 1] [-1] Correct
Writeup KNN 5 Test on [ 0 10] [1] not correct
Writeup KNN 5 Test on [10 10] [-1] not correct
Writeup KNN 5 Test on [5 5] [1] not correct
Writeup KNN 5 Test on [ 3 10] [-1] Correct
Writeup KNN 5 Test on [9 4] [1] Correct
Writeup KNN 5 Test on [6 2] [-1] not correct
Writeup KNN 5 Test on [2 2] [1] Correct
Writeup KNN 5 Test on [8 7] [-1] Correct
ps@Cuckoo  ~/MachineLearning/Project2  master ●
```

1

The best K for our Algorithm would be 9 with the accuracy of 0.7



```
ps@Cuckoo  ~/MachineLearning/Project2  master ●  python KNN_test.py
KNN tests:
KNN test on self = 1.0
KNN choose k on self = 1

choose_k on test  9
KNN k= 9 test on test accuracy = 0.7


TESTING K = 9
Writeup KNN 9 Test on [1 1] [1] Correct
Writeup KNN 9 Test on [2 1] [-1] Correct
Writeup KNN 9 Test on [ 0 10] [1] not correct
Writeup KNN 9 Test on [10 10] [-1] not correct
Writeup KNN 9 Test on [5 5] [1] Correct
Writeup KNN 9 Test on [ 3 10] [-1] Correct
Writeup KNN 9 Test on [9 4] [1] Correct
Writeup KNN 9 Test on [6 2] [-1] not correct
Writeup KNN 9 Test on [2 2] [1] Correct
Writeup KNN 9 Test on [8 7] [-1] Correct
ps@Cuckoo  ~/MachineLearning/Project2  master ●
```

2

[1] The picture has a high resolution so you should be able to zoom in and see the detail, but the picure has also been included in the zip file so please refer to that if there is any issues

[2] The picture has a high resolution so you should be able to zoom in and see the detail, but the picure has also been included in the zip file so please refer to that if there is any issues

2 Perceptron

2.1 Describe Implementation

2.2 Decision Boundary