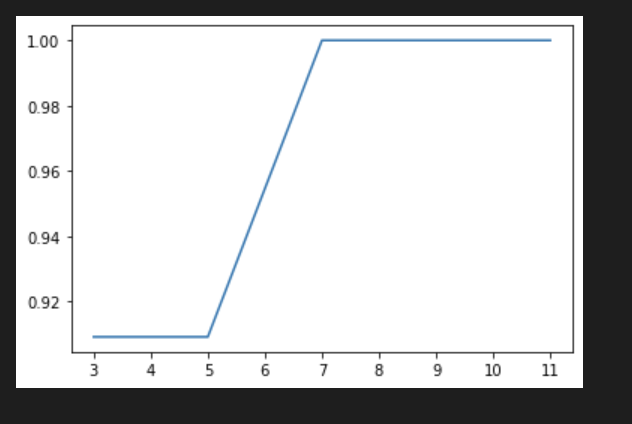
**Question 1 ( Manhattan distance )**

1.

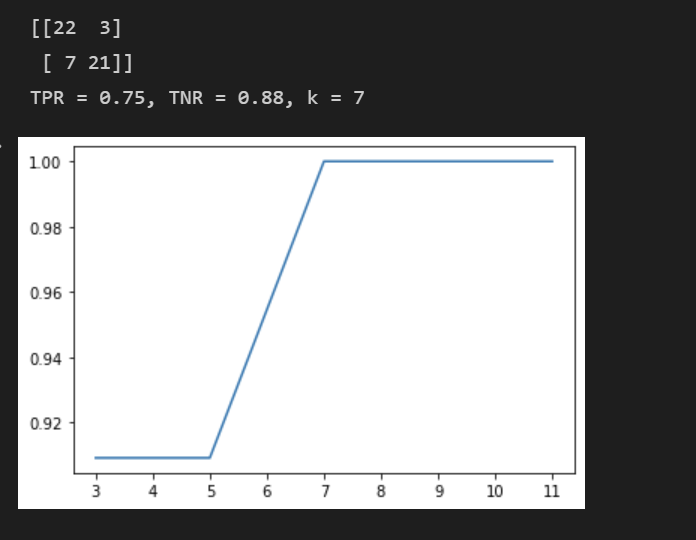
7 is the optimal value of k for year1



2.

81% is the accuracy

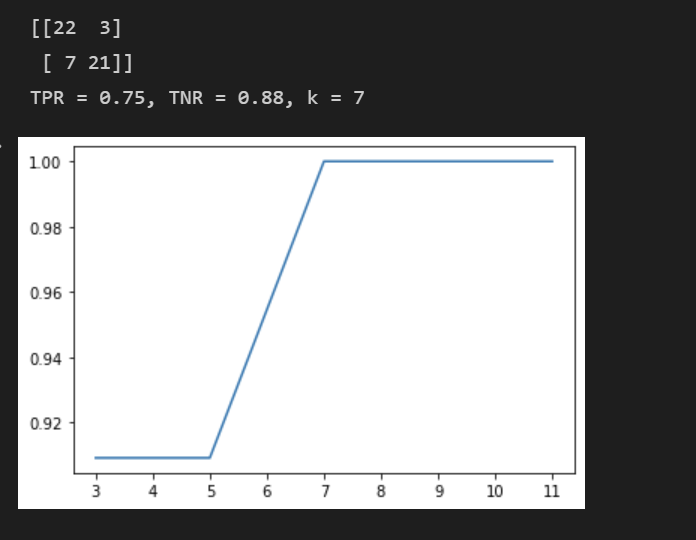
3.



4.

The k\* in regular KNN is 11. Therefor, k\* is different in Manhattan distance.

5.



6.

BH = 121

Manhattan KNN = 233

7.

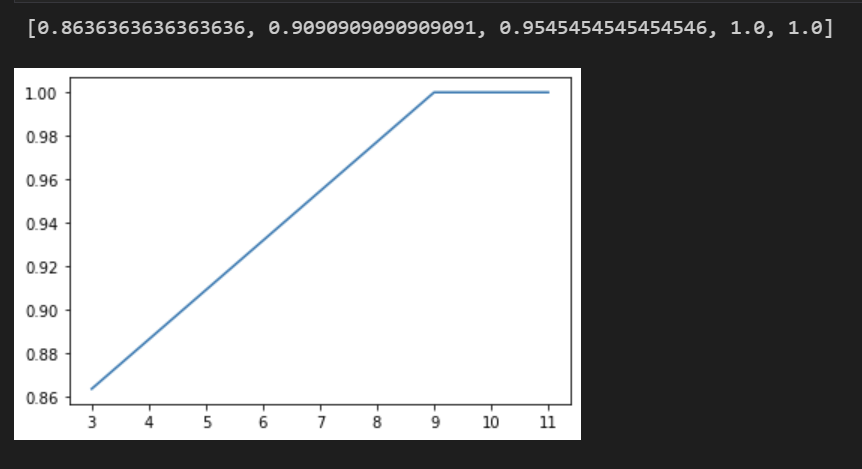
Regular KNN = 229

Manhattan KNN do improve a little.

**Question2(Minkowski)**

1.

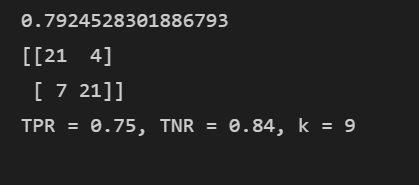
I choose k = 9 as the optimal k value



2.

79%

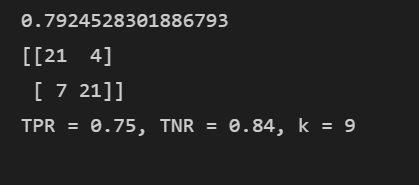
3.



4.

K in regular KNN is 11. Therefore, it is different

5.



6.

BH = 121

Minkowski = 235

7.

Regular KNN = 229

Minkowski do improve a bit.

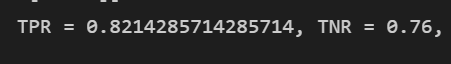
**Question3(Centroid)**

1.

|  |  |  |
| --- | --- | --- |
|  | Average | Median |
| Green Centroid | 0.72 | 0.51 |
| Red Centroid | 0.77 | 0.61 |

Red has a bigger sphere.

2.



3.

BH = 121

Centroid = 239

4.

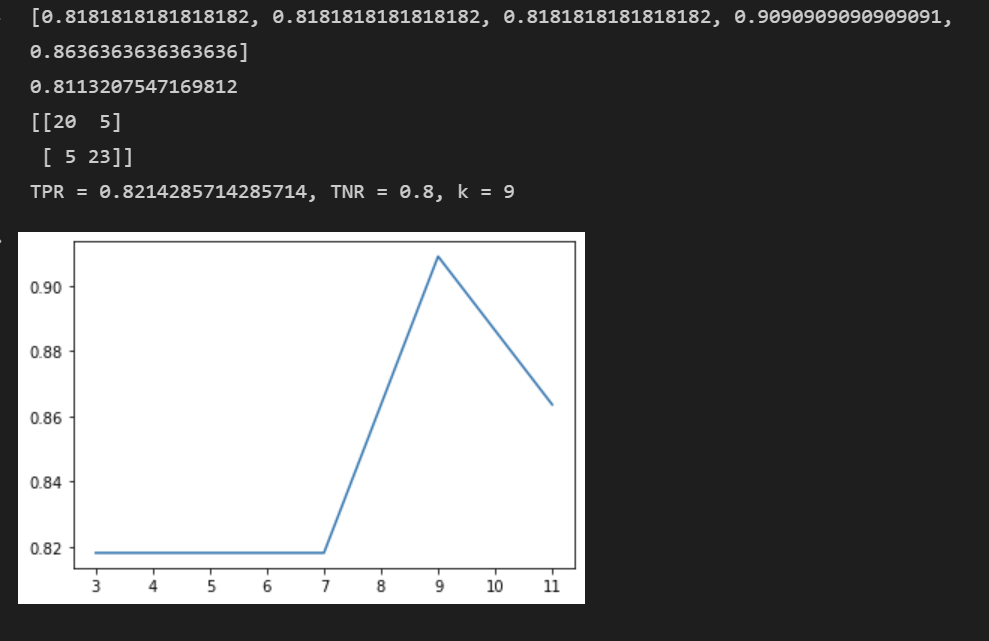
Regular KNN = 229

Nearest centroid method do improve a bit

**Question4(domain transformation)**

1.

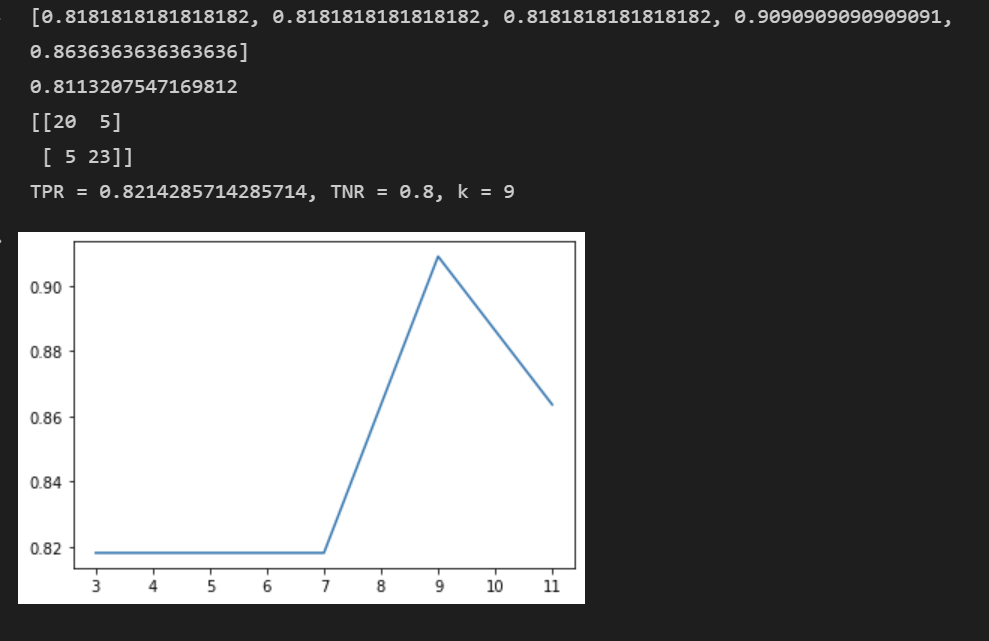
I choose k = 9 as the optimal k value



2.

81%

3.

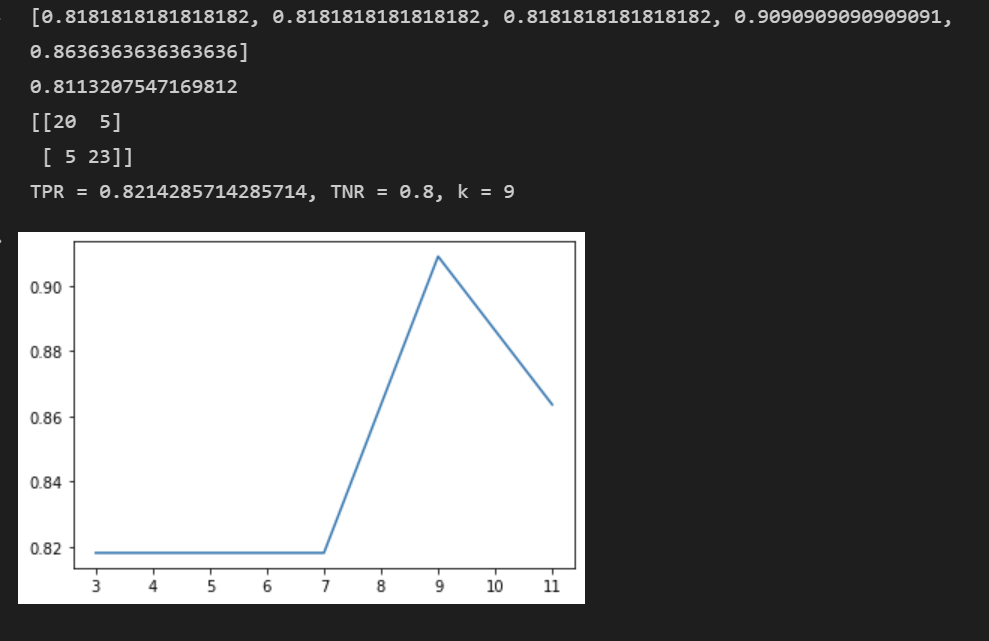


4.

K\* in regular KNN is 11.

It is different from k in domain transformation

5.



6.

BH = 121

Domain transformation = 238

7.

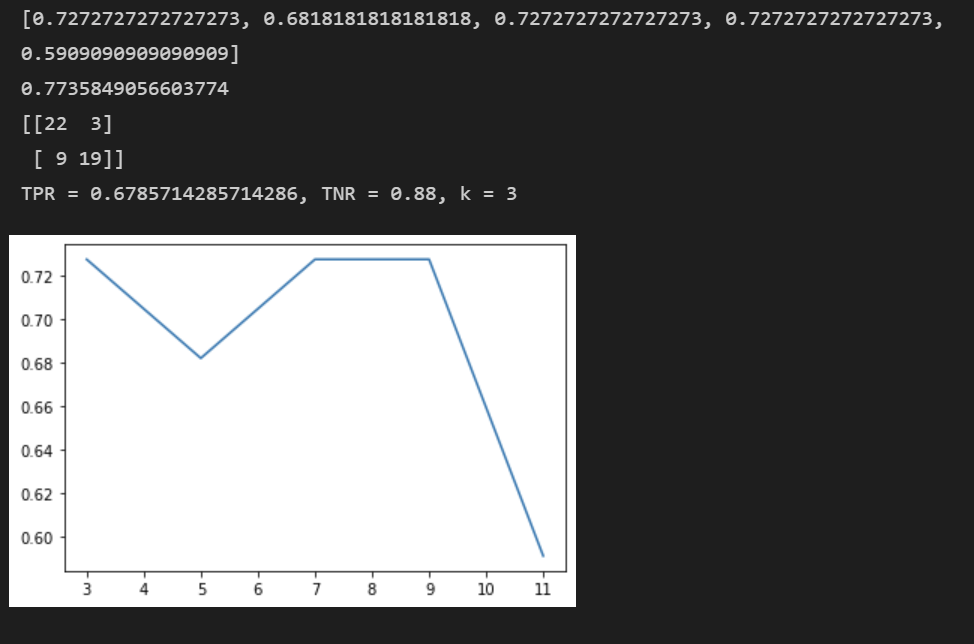
Regular KNN = 229

Domain transformation do improve a bit

**Question5(k Predicted Neighbors)**

1.

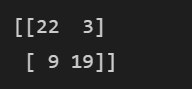
I choose k = 3 as the optimal k value



2.

77%

3.



4.

K\* = 11 in regular KNN.

Yes, it is different in k-predicted neighbors

5.



6.

BH = 121

K-predicted neighbors = 200

7.

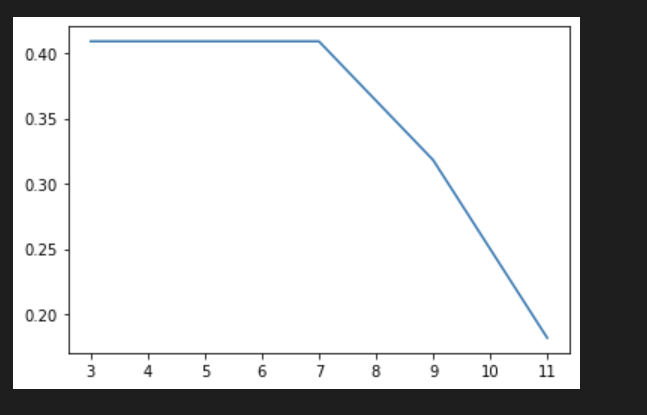
Regular KNN = 229

k-predicted neighbors does not improve at all

**Quesiton6( k-hyperplanes)**

1.

I choose k = 3 as the optimal k value



2.



3.

BH = 121

K hyperplanes = 60

4.

Regular KNN = 229

k-hyperplanes is not a good classifying method

**Question7(Summary)**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Best K | Accuracy% | Amount$ |
| BH | NA | NA | 121 |
| KNN (Euclidean) | 11 | 79 | 229 |
| KNN (Manhattan) | 7 | 81 | 233 |
| KNN (Minkowski) | 9 | 79 | 235 |
| Nearest Centroid | NA | 79 | 239 |
| Domain Transformation | 9 | 81 | 238 |
| k-predicted neighbors | 3 | 77 | 200 |
| k-hyperplanes | 3 | 41 | 60 |

First thing I would like to point out is that k-hyperplanes is not a good classifier algorithm for my case. The accuracy is below than 50%; therefore, the trading strategy with this method tends to lose money a lot.

For accuracy, all of the KNN variations, except k-hyperplanes, have accuracy about 80%, which is a good news. Moreover, it is only a slice difference when using different method to train.

For trading outcome, Nearest Centroid and Domain Transformation could provide the highest return.

**Question8(my own variation – KyleCircleLabel)**

For each black point from testing data, calculates the distance between this black point with each training data point, and make a list.

Then find the average distance of this black point to each training data points.

Assign r = 0.5\*average distance

A screenshot of a computer

Description automatically generated with low confidence

Create a circle with radius = r.

A screenshot of a computer

Description automatically generated with low confidence

Assign the majority of label to this new black point.

A screenshot of a computer

Description automatically generated with low confidence

|  |  |  |
| --- | --- | --- |
| Accuracy | TPR | TNR |
| 77% | 82% | 72% |

The performance of **KyleCircleLabel** algorithm, predict year2 from year1 data

For trading amount.

BH = 121

KyleCircleLabel = 223