HARIS RAMZAN
19I-2118

Section: A

Report

Machine Learning

February 7, 2020

Question 1: How is run-time effected by increase in number of training examples?

No of training examples	Manual prediction time(sec)	SKlearn Time(sec)
200	0.01157	0.0004
1000	0.1778	0.0014
2000	0.6658	0.0017
20000	64.4150	0.0078

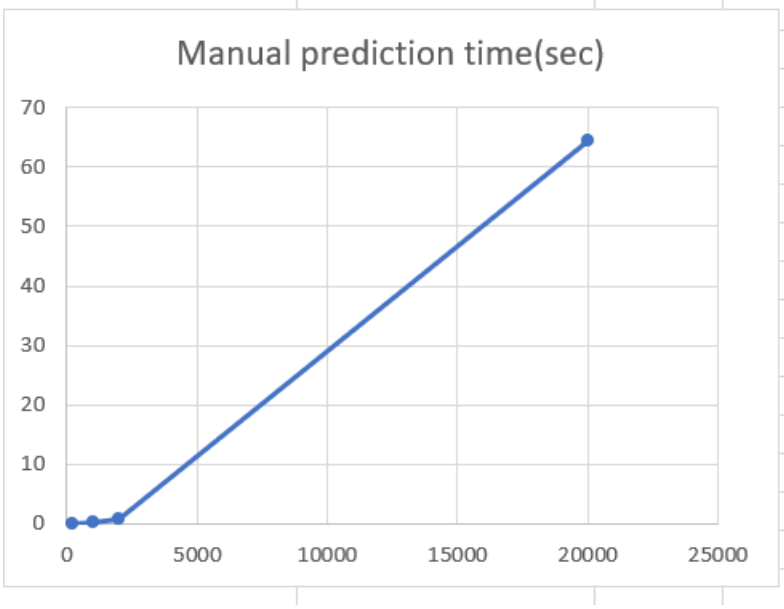


Figure 1 Time taken by manual implementation w.r.t example

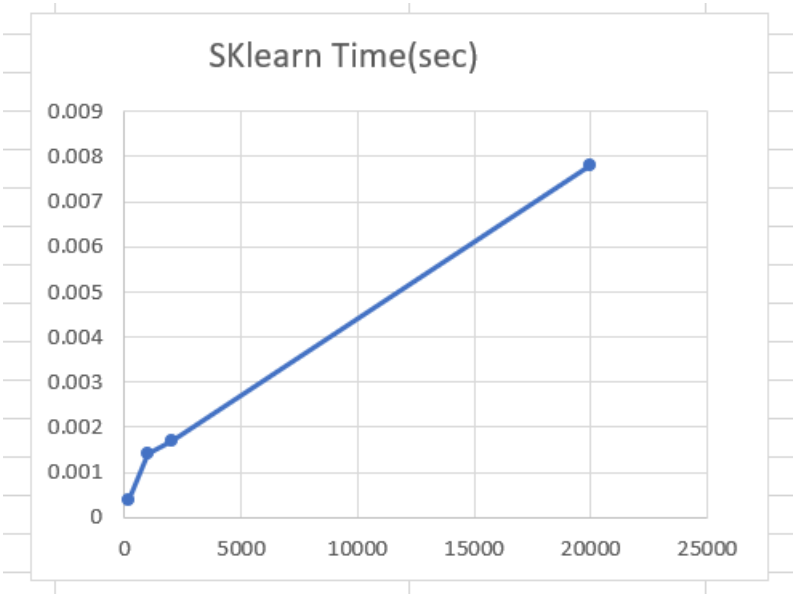


Figure 2 Time taken by SKLearn w.r.t training example

From the above figures it is clearly demonstrated that by increasing the number of training example there is a huge impact on time taken by the manual implementation as compared to skLearn implementation which taken lesser time. Both memory and time are consumed as well.

Question 2: How is run-time effected by increase in dimensionality?

N=5000,K=29

```
Dimensions of the data:  2
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 16.23113000000012
Accuracy of manual code is: 0.9193
Accuracy of sklearn is: 0.9193
Time taken to execute sklearn predict function is: 0.003070900002057897
```

```
Dimensions of the data:  3
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 20.01102499999979
Accuracy of manual code is: 0.9589
Accuracy of sklearn is: 0.9589
Time taken to execute sklearn predict function is: 0.0033614999993005767
```

```
Dimensions of the data:  4
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 24.572261600002093
Accuracy of manual code is: 0.9771
Accuracy of sklearn is: 0.9771
Time taken to execute sklearn predict function is: 0.003861900000629248
```

```
Dimensions of the data:  200
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 925.5894059999991
Accuracy of manual code is: 1.0
Accuracy of sklearn is: 1.0
Time taken to execute sklearn predict function is: 0.09705899999971734
```

As shown above by increasing the dimension accuracy is also increasing may be at some large value of dimension its accuracy is disturbed when curse of dimensionality occurs. When numbers of dimensions are increased the size of data is also increased exponentially and at some level apart points came close like nearest points which is called as curse of dimensionality.

Question 3: How is training accuracy effected by change in k?

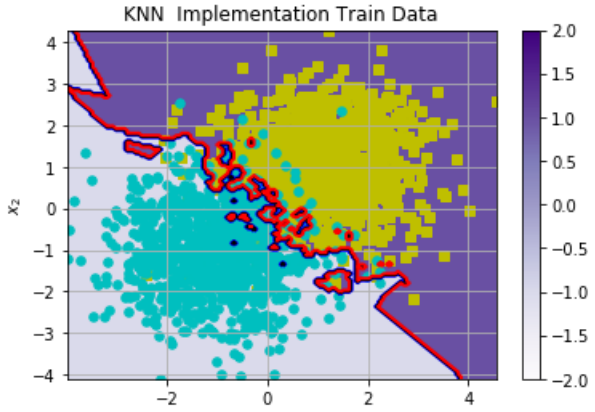


Figure 3 when k=3 and n=1000

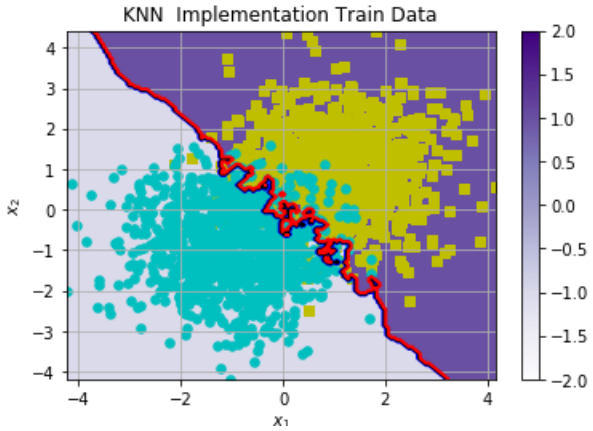


Figure 4 When k=5 and n=1000

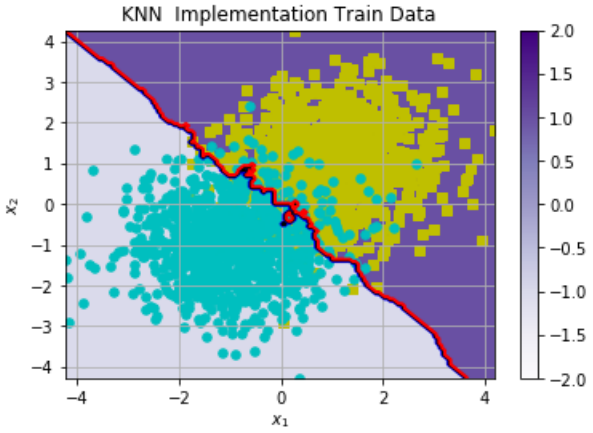


Figure 5 When k=9 and n=1000

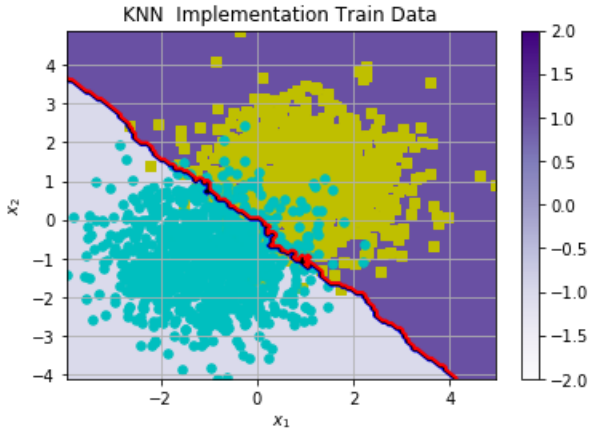
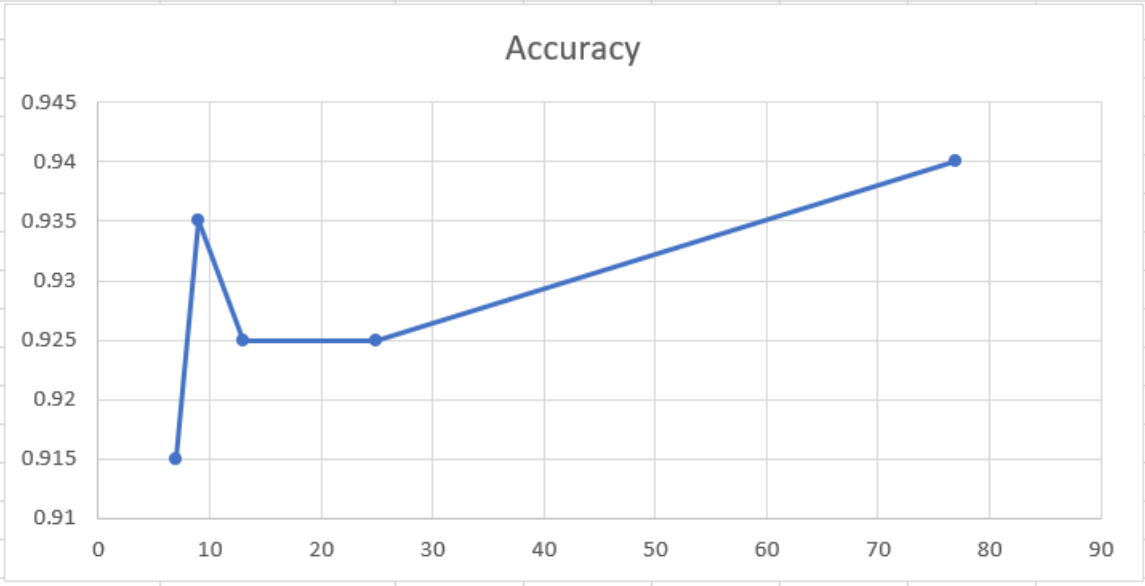


Figure 6 When k=21 and n=1000

By increase the value of k the line is getting straighter showing the line of best fit for the training data. But for the very large value of k more data points are wrongly classified as data points are crossing the red line. Coming closer to the best classified line on unseen data.

value of K	Accuracy
7	0.915
9	0.935
13	0.925
25	0.925
77	0.94



Question 4: How is test accuracy effected by change in k?

```
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 2.6326955999975326
Accuracy of manual code is: 0.9195
Accuracy of sklearn is: 0.9195
Time taken to execute sklearn predict function is: 0.0016523000012966804
```

Figure 7 n=2000 k=13

```
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 8.414749999999913
Accuracy of manual code is: 0.9223333333333333
Accuracy of sklearn is: 0.9223333333333333
Time taken to execute sklearn predict function is: 0.00214369999957853
```

Figure 8 n=3000 k=661

By changing the value of k test accuracy also changes at certain level and best accuracy is achieved when value of k is increased as shown above.

Question 5: How can you make your implementation more efficient?

- Implementation of the algorithm can be improved by using modern approach known as locality sensitive hashing technique to solve the problem of nearest neighbor problem in higher dimension space. In this approach the points lie close to each other are hashed to the same bucket.
- Code optimization can also be done to reduce execution time.
- Using effective data structures like dictionaries, as most of the python libraries use dictionaries.

Question 6: Compare the performance of your implementation with SKlearn implementation?

No of training examples	Manual prediction time(sec)	SKlearn Time(sec)	Accuracy (Sklearn=Manual Implementation)
200	0.01157	0.0004	0.895
1000	0.1778	0.0014	0.882
2000	0.6658	0.0017	0.9045
20000	64.4150	0.0078	0.9033
100000	2176.218	0.1023	0.9075

```
**********- Nearest Neighbor Implementation**********
Time taken to execute manual predict function is: 2176.218222700001
Accuracy of manual code is: 0.90751
Accuracy of sklearn is: 0.90751
Time taken to execute sklearn predict function is: 0.10231620000195107
```

There is a large difference between the sklearn implementation and manual implementation in terms of time specially. Although accuracy of both implementations is same.