Assigment # 1

2.Using your implementation of K-NN perform the following analysis for toy dataset. Also, Compare the performance of your implementation with SKlearn implementation.

a. How is run-time affected by increase in number of training examples?

Ans. Run- time will be increased as the number of training examples increased as shown in **Fig 1.1.** In the fig 1.1 we made a graph between Sample size on the x-axis and Time on the y-axis which clearly confirm that the Time take by the model will be increased as we increased the number of training examples in the fig there are two line Blue (shows the Sklearn Classifier) and the line Red (shows the My classifier) and between both of them Sklearn perform pretty much better than My Classifier.

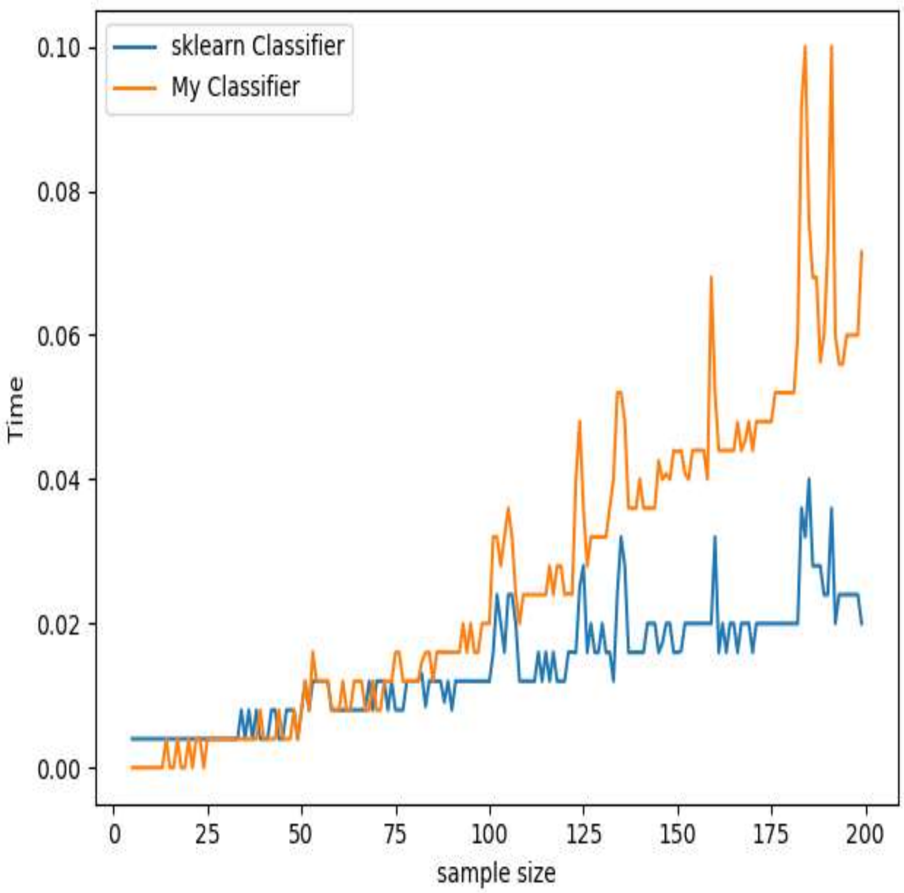


Fig. 1.1

b. How is run-time affected by increase in dimensionality?

Ans. Run-time will be increased as the number of dimensions increased as shown in **Fig 1.2.** In the fig 1.2 we made a graph between Dimension size on the x-axis and Time on the y-axis which clearly confirm that the Time take by the model will be increased as we increased the number of training examples in the fig there are two line Blue (shows the Sklearn Classifier) and the line Red (shows the My classifier) and between both of them Sklearn perform pretty much better than My Classifier.

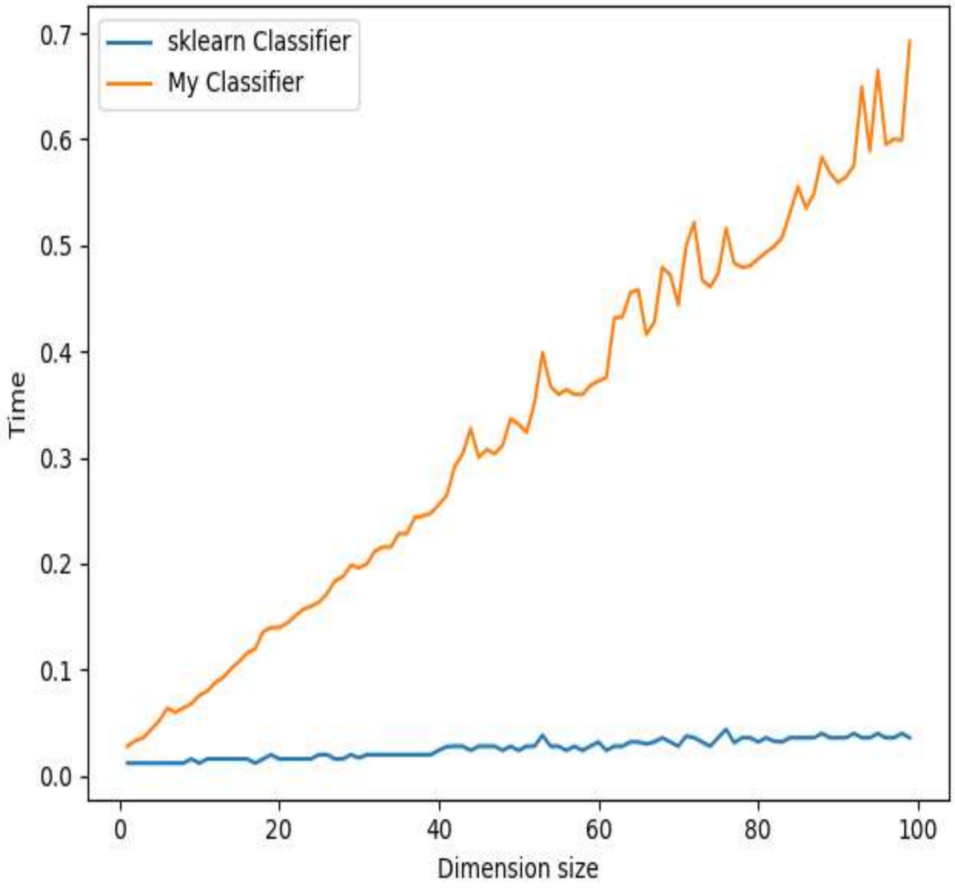
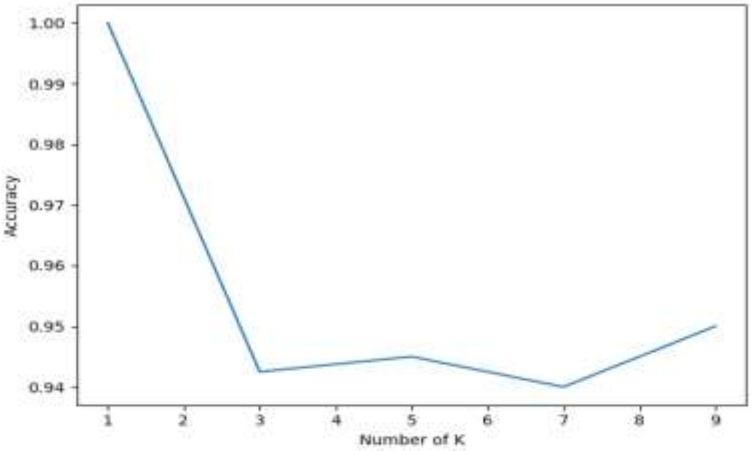


Fig 1.1

c. How is training accuracy affected by change in *K?*

Ans. As the number of K increased the training accuracy first decreased and then fluctuated as the shown in **Fig 1.3.** As we check this accuracy on training data so it shows 100 per for K=1 and as the number of K increased it starting fluctuating. If Fig 1,3 we made a graph between Number of K (x-axis) and Accuracy (y-axis) and a Blue line is the comparison between them which clearly shows the upper discussed results. As Sklearn and MY classifier have same trends so in below fig both graphs some the same result .

SKLearn Classifier



My Classifier

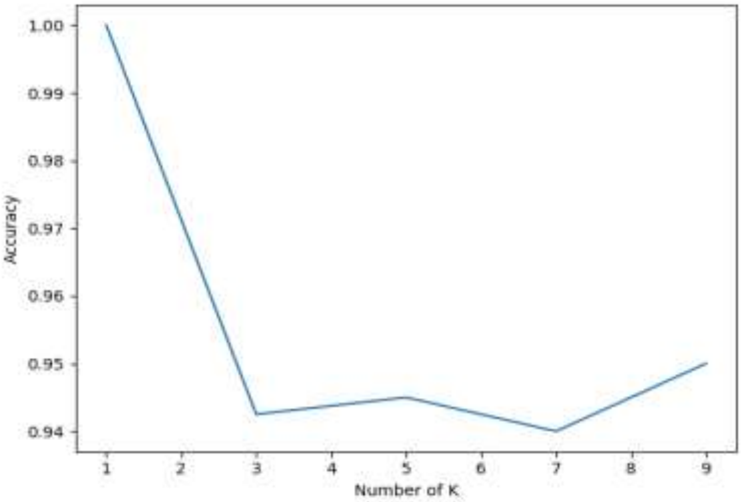
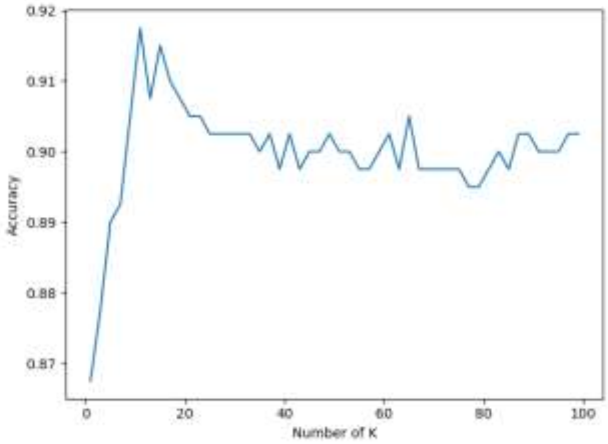


Fig 1.3

d. How is test accuracy affected by change in *K*?

Ans. As the number of K increased the training accuracy first increased and then fluctuated as the shown in **Fig 1.4.** As we check this accuracy on testing data so it shows near to zero accuracy for K=1 and as the number of K increased it starting fluctuating. If Fig 1.4 we made a graph between Number of K (x-axis) and Accuracy (y-axis) and a Blue line is the comparison between them which clearly shows the upper discussed results. As Sklearn and MY classifier have same trends so in below fig both graphs some the same result .

<u>SKLearn Classifier</u>



<u>My Classifeir</u>

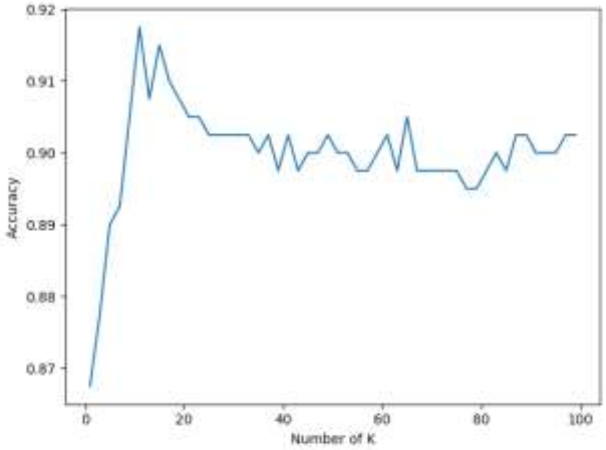


Fig 1.4

3.Give some ideas on how you think you can make your implementation more efficient?

Ans. We can make our implementation more efficient by using some of the following techniques

1. Distance Technique:

   In K-NN right now we are using Euclidian to find the distance as this technique will not work properly in case of higher dimension even by using some other techniques let's say Manhattan or the like technique.

2. Cross Validation:

   Cross-validation is a method that involves copying a set of data sets for which you are not going to model. You will then try to use the model before completing this sample. This method helps in the improvement of accuracy definitely when you already check the Model before testing it on the test data then it will most probably perform excellent on the test data

3. Weight Assign:

   Nearest K points assign a weight using a function "Kernel". Function value decreases as the distance increases and vice versa. This plays a great role in the accuracy of the algorithm which directly effects the efficiency of the algorithm.