

1. KNN algorithm is very fast as compared to linear classifiers such as logistic regression in this sense

*reason is that KNN only stores the training datagen and only will learn from it only during making real time predictions since it doesnt require training of the train dataset like the other linear classifiers thus no training period is required for it in this case

2.Logistic regression in this case will work perfectly when dealing with large datasets on high dimensions as compared to KNN

*reason being that it becomes very difficult for the KNN to calculate the distances in all dimensions which degrades the performance o the algorithm ,something which is easily achivable with Logistic regression

3. There are only 2 parameters that are required to implement the nearest neighbor i.e

*the distance function such as Euclidean, Manhattan or Minkowski and the K value

4.KNN is a memory intensive algorithm and as a result is classified as memory-based algorithm. This is because k-NN is a lazy classifier which works by memorizing all of the training set big O of n ($O(n)$) without learning time (running time is constant $O(1)$)

*as a result Knn algorithm uses relatively much memory to store all the memorized training datagen big $O(n)$ complexity

So in precise for a 64 bit system with a total of 4 blocks and number of attributes set as 9 ,this algorithm is calculated to occopy a memory of 24,588 bytes which is 71.4% in percentage

Num bytes = $683 * 9 * 4 = 24,588$ bytes

5. The computational cost complexity for our KNN model is of a big O complexity of order n times $\log n$ given as $O(n * \log(n))$ where n is the dimensionality value

6.The distribution of the Hamming distance of the data points implies that based on the size of the dataset which is growing exponentially on high dimensions it must be set to maintain the same space complexity ,so inorder to maintain the same space complexity then the same KNN density must be kept similar

