Assignment 1

Foundations of Machine Learning

IIT-Hyderabad

Aug-Dec 2021

Max Marks: 30

Due: 3rd Oct 2021 11:59 pm

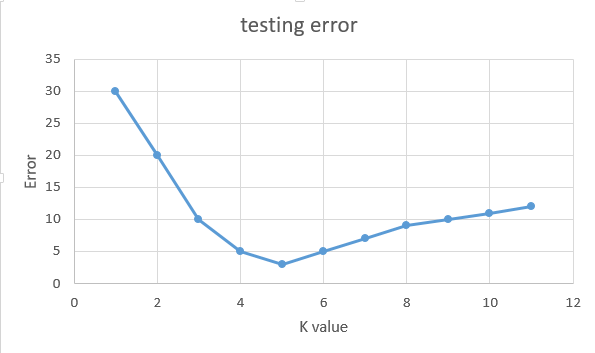
Answer 1(a): If K=n then it will start to take vote considering all rest of training points as neighbors and since the training set has equal distribution of the two classes the predicted outcome would be with tied in votes and the result set generated would be completely belonging to one of the class depending whichever class breaks the tie. The training error will be at its highest at this point.

As we start reducing k from k=n to k=n-1, n-3 and so on, the training error starts to reduce as points farthest from the current point will stop being part of the voting process and points in vicinity starts determining the predicted class of current point and training error goes down gradually

As we approach k=1, we will seeing a sharp dip in the training error as we approach an overfitting situation as the point closest to the current point will only be considered in voting and since the overlap between classes isn’t very high.

At k=1, the training error will become 0 as the only neighbor it will consider in the voting process will be the point itself.

Answer 1(b). The plot of testing error and k values would look something like this



At k=1, since it will be an over fitted model, almost all of the overlapped class points will be predicted incorrectly in test since it will only considering the most closest point as determining the class and the testing error will be at a high point

As we go on increasing, value of k, there will be a point of minima we will reach for testing error where a good enough balance of choosing the right number of neighbors for determining class of the testing points and the prediction will be more accurate. This will become our optimum value of K.

If we further go on increasing the value K, the error rate will keep on increasing again and reach it’s maxima at k=n, as it will start predicting based on basis of majority class in entire dataset. It will be an underfit model

Answer 1(c). Reason1: Distance calculation becomes meaningless in higher dimensions

Explanation: In a higher dimension space, the distances between points becomes too large for Knn to converge on nearest neighbors as almost all the points in the space will be equidistant from a particular point and lying on the edge of hyperspace and hence the nearest point becomes as far as an average distance between all points, the algorithm won’t be able to figure the right set of neighbors to pick for prediction

Reason 2: Would require exponentially large training data to overcome the higher dimensions issue mentioned above

Explanation: Since higher dimensions is basically hollow and most points will effectively lie at the surface of the space, in order to maintain a good data density inside the space, one would need an exponentially large number of observations in the data for KNN to identify nearest neighbors. This brings more constraint as now one would need a powerful computer and a large storage to run KNN as time complexity of KNN is O(nkd), where ‘n’ is no of observations, ‘k’ is the nearest neighbor constant and ‘d’ is dimension. As ‘d’ increases so will ‘n’ in order to maintain the data density and will result in exponential increase in time complexity

(d) *(3 marks)* Is it possible to build a univariate decision tree (with decisions at each node of the form “is *x > a*”, “is *x < b*”, “is *y > c*”, or “is *y < d*” for any real constants *a, b, c, d*) which classifies exactly similar to a 1-NN using the Euclidean distance measure? If so, explain how. If not, explain why not.

Answer 1(d). No, it is not possible for any real constant of rule set “x>a” or “x<b” or “y>a” or “y<d”. 1NN will basically be a cell boundary itself and does not follow lines parallel to coordinate axes and Decision Tree basically forms multiples parallel lines to enclose a decision in the intersecting area. For the decision tree to predict each point would require such an uncountable number of lines to predict the each and every point

Let’s us take an example of below data

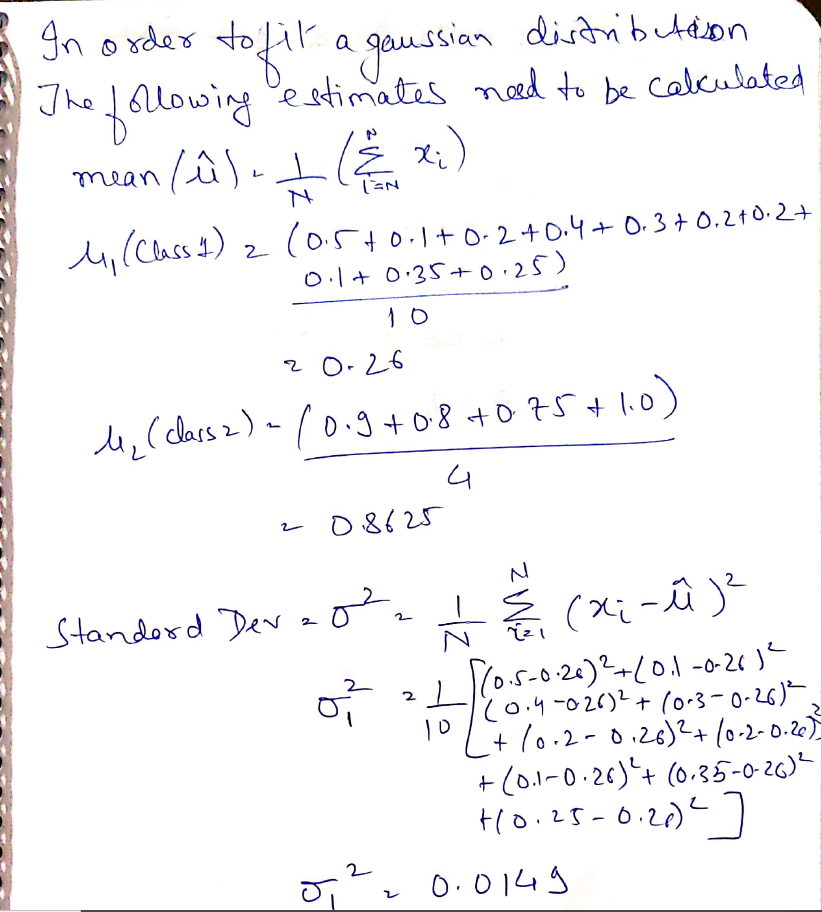
|  |  |  |  |
| --- | --- | --- | --- |
|  | x | y | class |
| 1 | 1 | 2 | 0 |
| 2 | 2 | 1 | 1 |
| 3 | 0.5 | 2 | 0 |
| 4 | 2 | 0.5 | 1 |

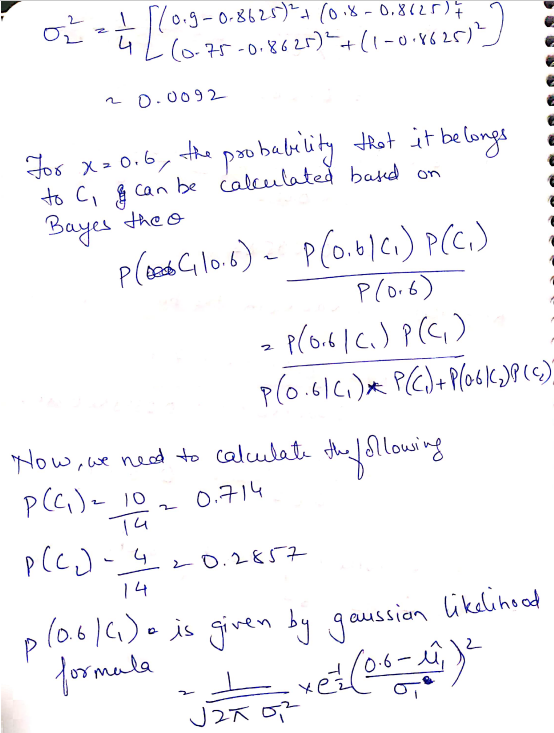
For 1-NN, there are no overlapping class points and will perfectly predict the classes

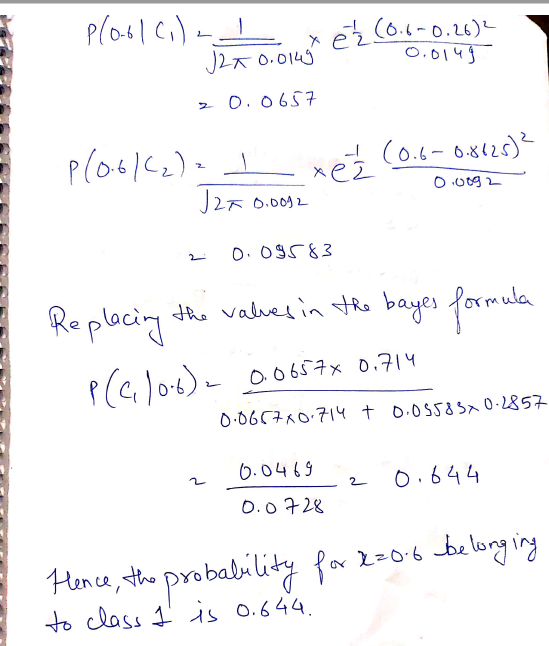
But in case of decision trees, only a few rule sets would actually provide the same output as 1NN that being one of the following univariate rules (“X<2”) ,(“y>1”),(“x>1”),(“y<2”)

But if the constants were chosen in a range outside of the above rule set, it would result in both observation getting predicted under same class and would be give a different result other than the one provided by 1NN. Hence Decision tree won’t be able to reduce its rules to a particular set to determine all the points in a given space

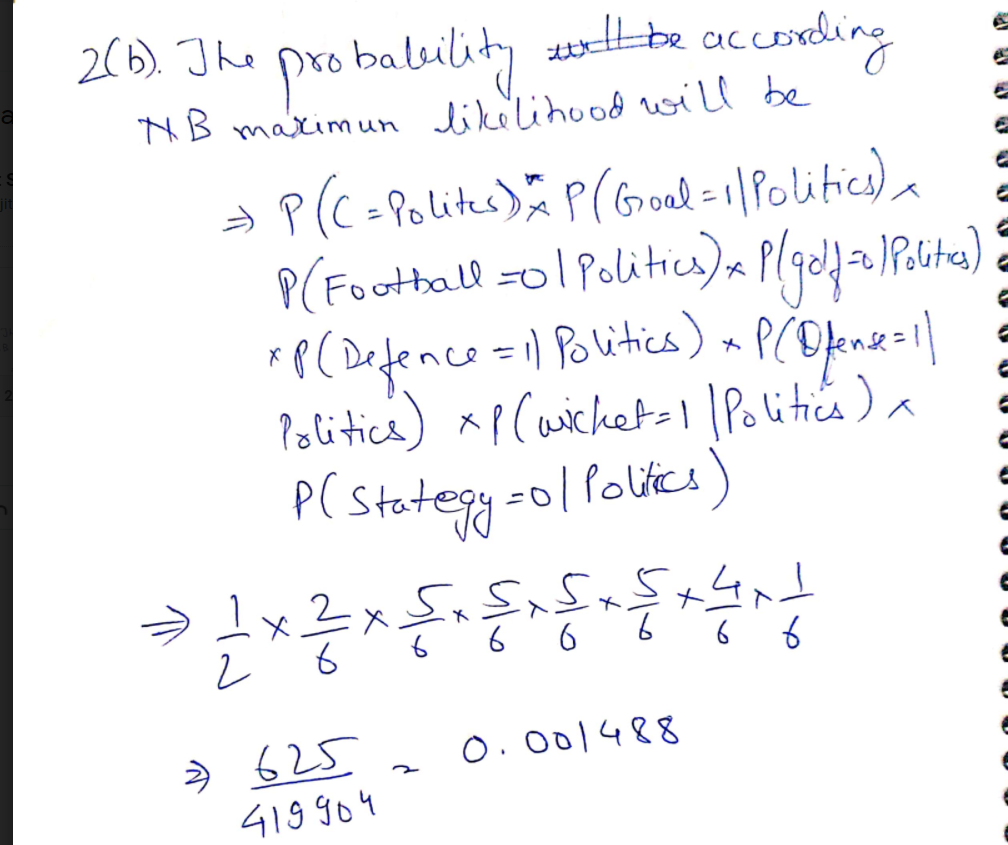
2(a). I have done it on paper below are the paper scans:







2(b).



3 (a,b,c). Deliverables:

• Code – 3 codes

1). Initial implementation with entropy, univariate, binary and full depth tree

2). Improvement with introduction of pruning

3). Improvement with introduction of gini index

4). Brief report (PDF) explaining the code and implementation