SMAI Assignment-2 Report

Name: Akshat Maheshwari (20161024)

Question-1:

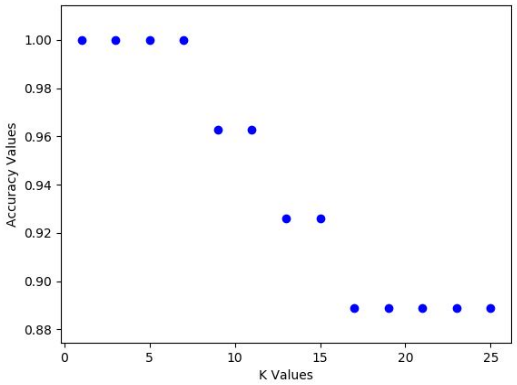
Iris Dataset:

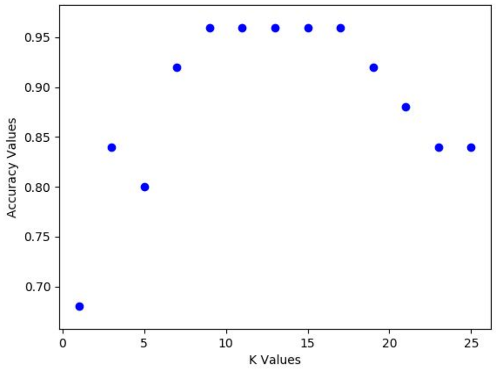
- Best Accuracy: 1.0
- Best Precision: 1.0
- Best Recall: 1.0
- Best F1-Score: 1.0

Robot Dataset:

- Best Accuracy: 0.98
- Best Precision: 1.0
- Best Recall: 0.8
- Best F1-Score: 0.55

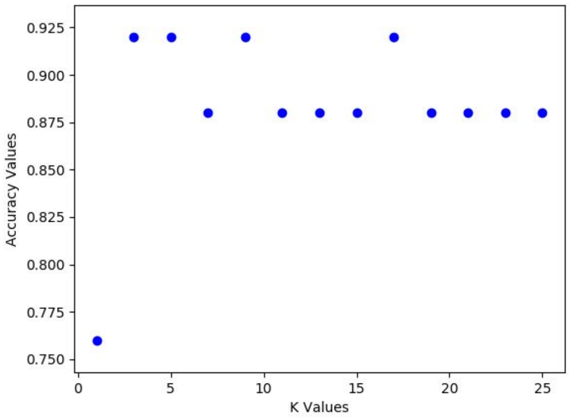
These values might change on running code different times, because I have randomly shuffled the dataset before splitting it into train and validation.





For iris dataset

For robot1 dataset



For robot2 dataset

I have mainly used the L2 distance as the distance function here, because of the reason that it will handle the negative values if they come in the difference.
Also, one more advantage of this over the L1 distance is that L2 is the smallest distance between any 2 possible points in the space.

I am always taking the value as odd numbers like sir told in the class.
One more observation is that, generally on smaller values of k, better accuracies are obtained in my model.

<u>Question-2</u>:

Bayes' Theorem:

P(h|d) = (P(d|h) * P(h)) / P(d)

- P(h|d) is the probability of hypothesis h given the data d. This is called the posterior probability.
- P(d|h) is the probability of data d given that the hypothesis h was true.
- P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

- P(d) is the probability of the data (regardless of the hypothesis).

$$P(Y|X_1, X_2, …., X_n) = P(Y) \prod^{k}_{i=1} P(X_i \mid Y)$$

This is the formula that is used for calculating the probabilities.

General steps used for implementing the classifier:

1. **Load Data**: Load the data from CSV file and split it into training and test datasets.
2. **Summarize Data**: Summarize the properties in the training dataset so that we can calculate probabilities and make predictions (Basically calculate the mean and variance corresponding to each column of the dataset).
3. **Make a Prediction**: Use the summaries of the dataset to generate a single prediction.
4. **Make Predictions**: Generate predictions given a test dataset and a summarized training dataset.
5. **Evaluate Accuracy**: Evaluate the accuracy of predictions made for a test dataset as the percentage correct out of all predictions made.
6. **Tie it Together**: Use all of the code elements to present a complete and standalone implementation of the Naive Bayes algorithm.

As per my implementation, I have used 0.5 as the threshold value. It gives the best possible result among some of the values I tried.

Best accuracy attained = 90%

This accuracy might vary in different run of the code as the dataset is randomly shuffled before splitting into the train and validation sets.

Question-3:

Steps used for Linear regression:

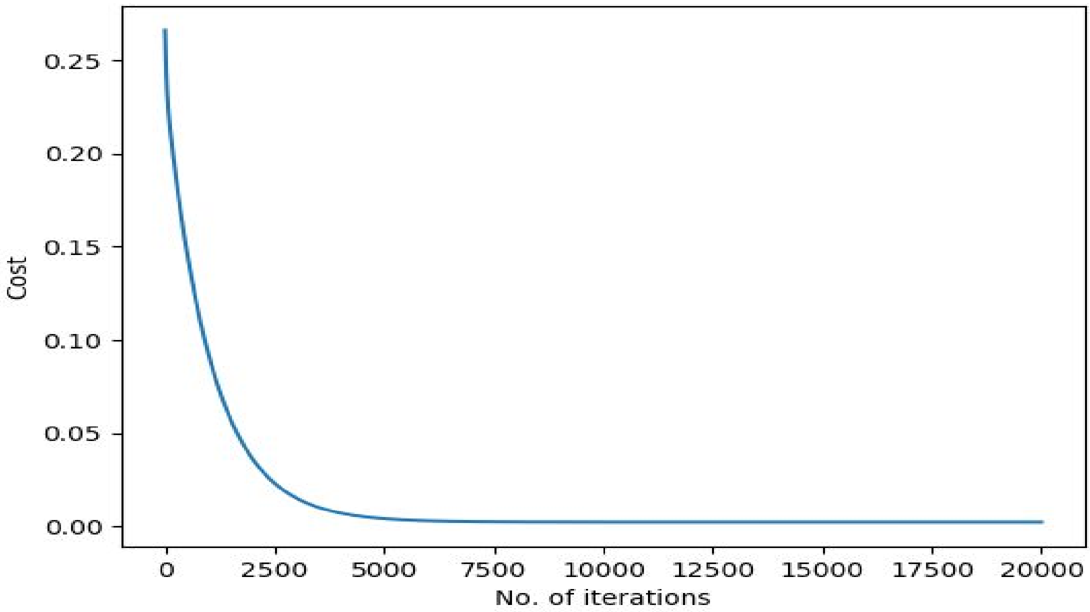
1. **Load Data**: Load the data from CSV file and split it into training and test datasets.
2. **Normalize the Data**: Normalize the data in the columns in which the elements are not in the range of 0-1.
3. **Creating matrices and set Hyperparameters**: A column vector of all ones is added to the start of the X matrix. Alpha, number of iterations and theta is set.
4. **Cost Function**:
    a. $h_{\Theta}(x) = \Theta_0 + \Theta_1 x + \Theta_2 x^2 +$ ……..
    b. $MSE = (1/n) * \Sigma^n_{i=1}(Y_i - Y\hat{}_i)^2$
    c. Gradient Descent:
        i. $\Theta_j = \Theta_j - \alpha * d/d\Theta$ costFunc(Θ), where,
        □ = learning rate
5. **Gradient Descent**

I have used the matrix method to solve the problem of linear regression.

MSE = 0.0020848194393353896
MAE = 0.02480113652017857
MPE = -0.016873440579037457

The above values may change on different iterations, because I have randomly shuffled the dataset before splitting it into train and validation tests.

Graph:



The cost here is actually the error obtained in that iteration. Although the error values may vary in different runs of the code, but the graph remains the same. The graph has the same curve always, that is required for good efficiency of the model, that the error should decrease with the increase in the number of iterations.