Ummay Habiba

CISC 5800: Machine Learning

Final Project

**Ultrasonic Flow Meter Diagnostics Data Set: Experimenting with Naive Bayes and Logistic Classifier to Classify MeterA**

**Background**

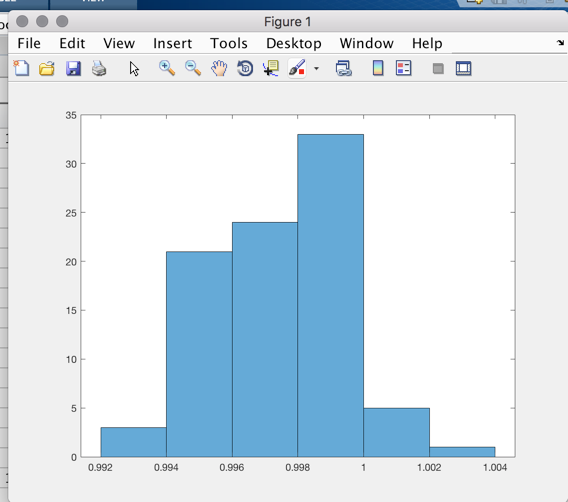
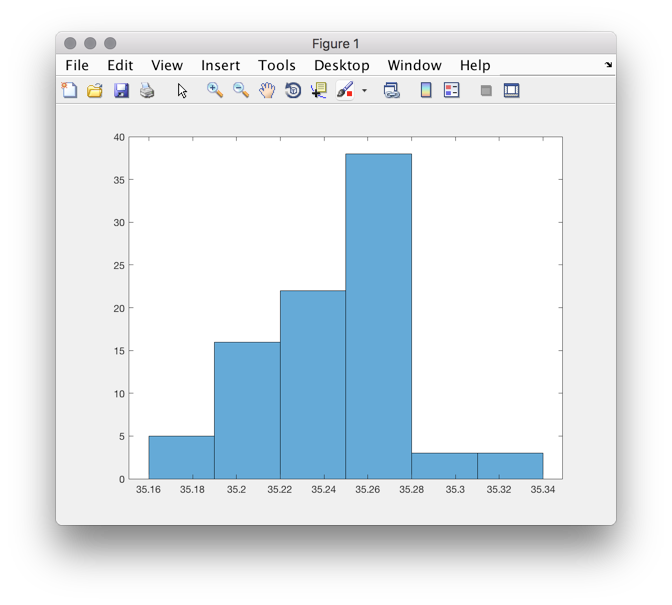
The Ultrasonic flowmeter diagnostic dataset aims to fault diagnosis of four liquid ultrasonic flowmeters labeled Mater A, Meter B, Meter C, and Meter D. This machine learning classification is done only on Meter A which contains 87 instances of diagnostic parameters for an 8-path liquid ultrasonic flow meter. This dataset has 37 attributes and 2 classes. The data is broken down as follows: (1) -- Flatness ratio (2) -- Symmetry (3) -- Crossflow (4)-(11) -- Flow velocity in each of the eight paths (12)-(19) -- Speed of sound in each of the eight paths (20) -- Average speed of sound in all eight paths (21)-(36) -- Gain at both ends of each of the eight paths (37) -- Class attribute or health state of meter: 1,2. In this project although some data points fall under the same feature, for example, Crossflow is associated with columns 4-11 on the dataset, the classifications done recognizes these as separate features that all fall under Crossflow. More specifically because it is an 8 path flow meter, feature 4 would be associated with Crossflow for the first path, feature 5 with the second path, 6 with the third path and so on.

**Preparation of Data**

This project aimed to identify which classifier did a better job of being able to classify the data using two machine learning classifiers the Naive Bayes Classification and Logistic Regression Classification. In order to apply classification methods, the data needed to be prepared. The data from MeterA was taken and normalized. Then the normalized Meter A data set was divided into TrainMeterA and MeterATest. Where 80% of the data points were put into the Training Set, exactly 70 points of data, and 20% into the Testing Set, exactly 17 points of data. This was in order to provide a good number of points for our classifier to learn from. This division was data was done in Excel.

**Naive Bayes Classification**

In order to do the Naive Bayes classification, more specifically the Gaussian likelihood, it was important to understand the distribution of numeric values for each feature. After using the histogram function on Matlab it was evident from the graphs that the Gaussian distribution would be the best fit.



|  |
| --- |
| The figure on the left represents the bar graph of the Feature Symmetry while the the figure on the right represent the bar graph of the average speed of sound in the 5th path. |

The next thing done in order to begin classification was the mean and standard deviation needed to be learned from the Normalized Data. Once that was completed those values were put into excel sheets in order to be used for calculating the Gaussian and Naive Bayes. In order to calculate Naive Bayes, the Priors were also needed which when using the entire Normalized Dataset and not the training tet came out to .40 for C1 and .60 for C2. The first time the classification was done it was computed without multiplying priors. This created a 58% accuracy when used in the test data. However, when the priors were included, a change in the hyperparameter the accuracy decreased to 52%.

**Logistic Regression**

For the Logistic Regression Classification, the classes in the learn and test data files had to be changed in order to fit the logistic classification which classifies based on 0 and 1. So all classes with health C1 were changed to 0 and all classes with health C2 were changed 1. The modified data sets are then used to perform logistic regression. The number of weights is generated randomly based on the number of features in the training set, this is done in learnW where the sigmoid function is used to calculate the weights. For logistic regression, weight learning is performed for a specified number of loops with the training data via iterative gradient ascent with no regularization. Afterward, a classification algorithm is performed on the testing data, returning the class labels for each data point. These labels are compared against the actual labels, leading to the calculation of an accuracy of correctly labeled data points. This accuracy number ranges from 0 to 1, with larger numbers indicating more accurate results. The logistic classifier experiments with three hyperparameters training loops, the value of the step size, and the value of b.

The first hyper-parameter was the number of training loops performed on the weights. The ranges were 1, 50, 100, 250, and 500 testing loops in order to reflect the possibility of too much testing or too little testing. The second hyper-parameter was the epsilon step size. The epsilon should not be too large or too small since however larger epsilons were used to see the effect. The range of epsilon was 0.1, 0.5, 1, 5, and 10. The last hyper-parameter was b, a value used alongside weights to help define a boundary of separation between classes. The b ranges were set to see its effects on classification when b is both positive, negative, or 0, and this resulted in a range of -10, -5, 0, 5, and 10.

|  |  |  |  |
| --- | --- | --- | --- |
| Number of Loops | Epsilon (Step Size) | b | Percent Accuracy |
| 1 | 1 | 0 | 0.58824 |
| 50 | 1 | 0 | 0.41176 |
| 100 | 1 | 0 | 0.47059 |
| 250 | 1 | 0 | 0.41176 |
| 500 | 1 | 0 | 0.6348 |
| 100 | 0.1 | 0 | 0.41176 |
| 100 | 0.5 | 0 | 0.58824 |
| 100 | 5 | 0 | 0.47059 |
| 100 | 10 | 0 | 0.41176 |
| 100 | 0.5 | -5 | 0.11748 |
| 100 | 0.5 | -10 | 0.35294 |
| 100 | 0.5 | 10 | 0.94118 |

For logistic regression, it is evident that a high number of loops or high b values produce a higher percent correct while lower b values produce a lower percentage of accuracy.

**Conclusion**

When comparing the results is it evident that the logistic classifier did a better job of classifying specifically when the hyperparameter b was high and the number of loops. While the Naïve Bayes Classifier computed with 58% accuracy the Logistic Classifier was able to reach an accuracy of 94%. For future tests, higher b’s will be used in order to get a higher accuracy. For future experiments, I would test out more varied hyper-parameters while focusing on the effects of b. A focus would be placed on logistic regression, as it showcased the most promising results. I would also be interested in understanding how the functions overfit for the data by learning to well and how this affects my percent accuracy.

The results from this expeirment are not fully acurate. It is imporant to take into account that the data sets were fairly small and overleanring may occurred. Also important to acknowledge that a cross-validation data set would have improved the results of the LogisticRegression as overfitting became evident when running test results. The more a specific line call for numCorrectLogistic was run the higher the percent accuracy became because the functions were better able to learn from the data which also caused overfitting. Even with these results, it is important to note that the result of these experiments should not be considered conclusive as there were many limitations present. The randomization of the weights values could have also caused poor results. The logistic regression algorithms did not feature any form of regularization, which could have impacted the value placed on the weights.