**PROJECT REPORT**

**INSTANCE-BASED LEARNING AND FEATURE SELECTION**

**CSCE-633: MACHINE LEARNING**

**Submitted By:**

**Amandeep Singh Bhal**

**UIN: 724003595**

Implementation:

I have implemented the following algorithms:

* KNN Algorithm
* Distance Weighting Algorithm
* Feature Selection using Stepwise Backward Elemination
* NT Growth Algorithm

Implementation Details:

**Pre-Processing of Data:**

The first step of preprocessing was converting the data to be compatible with the system it will run on. This involved converting it from comma-separated values to list of string values. In the pre-processing the data records have been read from the .txt file and have stored it in a list format.

**Data Normalization:**

To prevent irrelevant attributes from overpowering the KNN algorithm we need to normalize the dataset so that each value is between [0,1]. One drawback of normalizing datasets is that it can discard the information and make it hard to analyze the data. For the classifiers used, preprocessing was done before presenting the inputs to the learning model by normalizing the variances of the input values to a range of [0,1] using following equation:

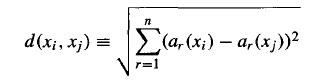
x = x – xmin/xmax – xmin

**Handling Missing Data:**

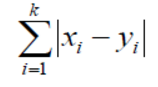
For Discrete attributes I have replaced the missing attribute value with the most frequent value in the dataset for that attribute and for the Continuous attributes I have replaced the missing attribute value with the mean of the values in the dataset.

**KNN Algorithm:**

I have implemented the KNN Algorithm as an Instance based algorithm. In this algorithm we assume that all instances correspond to points in the n-dimensional space. Here n is the number of fields/attributes. The nearest neighbor is defined in terms of the Euclidean Distance for continuous attributes. Thus the distance between 2 instances xi and xj is given by the following formula:



In case of discrete attributes, I have calculated the distance in terms of Hamming Distance which is given by the following formula:



The pseudocode of the algorithm is as follows:

KNN-Algorithm(tData, TraininingData, K)

1. Iterate over tData length taking one tRecord t at a time

Now iterate over all the TrainingData one instance ti at a time

Find the top-k element closest to t.

After finding the top-k element closest to t. Iterate over these elements

Find the majority classifier among these top elements.

Match the predicted class with the class of record t.

Update the accuracy count accordingly.

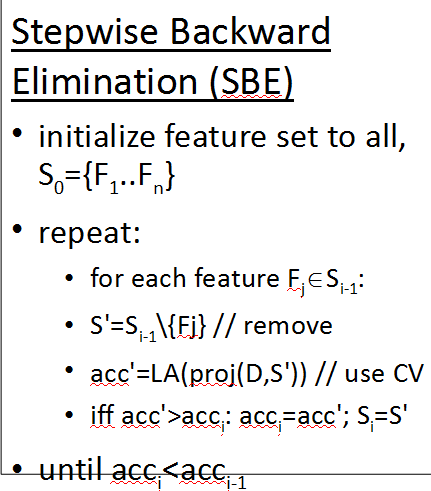
**Distance Weighting Algorithm:**

I have also implemented distance weighting algorithm. It allows all the training examples to have the influence on the classification of the instance x. There is no harm in allowing all the training examples to influence the classification, because the examples which are very far will have very little contribution towards the weight. The weight influences the classification by following formula:

https://lh5.googleusercontent.com/XBCJ_k4mXf9MCGksJnb-CMY2TyzWtf62AoHE97FJ2a7z6Bz9NZd5LrzKAbTeDH4ccxFZALCHKP9TciKa5nz9YB3TSLP1XBZD5_GoTawuxO7xaH7PoKySCQRvAzhFWzI9Yrgz7VxT

**Feature Selection Algorithm: Stepwise Backward Elimination(SBE)**

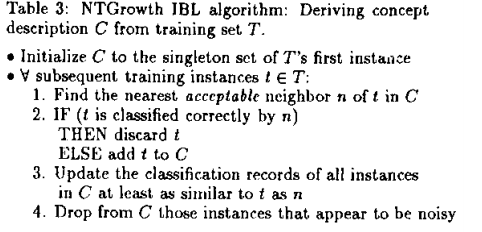
I have implemented Stepwise Backward Elimination for Feature Selection over the DataSets. The pseudo code of the algorithm is as follows:



**NT Growth Algorithm:**

For handling the noisy data, I have implemented the NTGrowth algorithm. Instance based algorithms are highly sensitive to noisy training instances. By implementing NTGrowth noisy data can be detected and removed from the training data. As proved in the Aha and Kibler paper, this method is shown to be performing better in case of noisy data and provides better classification.

Below is the pseudo code of the algorithm:



IRIS DataSet Effect of K:

Effect of K on Heart DataSet:

Effect of K on Tic-Tac-Toe DataSet:

Effect of K on Credit Screening DataSet:

Effect of K on Voting DataSet:

Effect of K on Wine DataSet:

DataSets Used:

1. Tic-Tac-Toe
2. Credit Screening
3. Pima Indians Diabetes
4. Heart
5. Voting
6. Wine

Results:

1. Tic-Tac-Toe
2. Credit Screening
3. Pima Indians Diabetes
4. Heart
5. Voting
6. Wine