ITCS 6156: Machine Learning

Homework 1

Student Name: Rajdeep R Rao Student ID: 800972470

For this homework assignment, I chose to work with python because I haven't worked with this language before, so it was a wonderful opportunity to learn. Also the numerous libraries that are available made it easier to do the project.

Before starting to implement, I tried to learn the concepts from a variety of sources online. I tried to build naïve Bayes and decision tree model code from the ground up, but hit too many dead ends, so I decided to use libraries instead.

Steps Taken:

I first read, said data into a DataFrame using one of the libraries in python named 'Pandas'. I then converted the data into a binary representation of 1s and 0s instead of number of occurrences. This data frame was then converted into an appropriate format required for naïve Bayes and Decision Tree functions in scipy library.

Naïve Bayes:

The Naïve Bayes algorithm assumes that the value of one feature is independent of any other feature. This can be represented mathematically as,

If there is a vector $x = (x_1, x_2, \ldots. x_n)$ where there are n independent variables then conditional probability can be computed as

$p(C_k /x) = p(C_k)\, p(x|C_k) \;/\; p(x)$

Naïve Bayes Implementation:

For the implementation of Naïve Bayes, I used a function called 'MultiNomialNB'. I have also used a smoothing factor (α) as 1. This function performs the Naïve Bayes algorithm on given data when inputted after transforming it into an appropriate format and then, a trained model is returned. This trained model is then tested against the test data. Finally, I matched the output with the labels in the test.data file by copying both the produced output and the correct output into an excel sheet for those first 500 documents (as mentioned in the info file) and compared the two rows via excel operations. I found the **accuracy of this model to be around 76%**. Also, since there were so many more words in the test dataset that weren't in the training dataset, my model was thrown off by it towards the end, thereby producing poorer results. Also, Since I've truncated the value of the tables, I've hardcoded the column values of the frequency tables in the code. So, it's not flexible for all datasets.

Decision Tree:

Decision tree is a supervised learning mechanism which is most suitable to be used when there are finite set of values.

Decision Tree Implementation:

Data was transformed into an appropriate format and fed into the DecisionTreeClassifier function which is also a library function call. I have considered the depth to be 10 and again, received an output for the model. This model was again tested in the same unconventional-excel method with the test data and found to have an accuracy rate of **67%**.

Information Gain:

Information gain is defined as the measure of information received from the data. When analysing huge dataset only data of value is important which can be used. Information gain and entropy are the two forms where measure of the information can be calculated. It’s kind of broken. Since I had no backup of the code for 500 documents, I’ve had to upload it like that**.**

Cross-Validation:

Implemented cross validation for naïve Bayes and decision tree using 10 folds. I used train_test_split function from sklearn library in python. And displayed the score at the end of the 10 iterations. It gives a good 85-95% accuracy as I’ve noticed.