# Project 05: Machine Learning

## Group No. 24


| | | |
|---|---|---|
| Charuta Pethe | 111424850 | cpethe@cs.stonybrook.edu |
| Deven Shah | 111482331 | dsshah@cs.stonybrook.edu |
| Nikhil Mehta | 111471539 | nsmehta@cs.stonybrook.edu |
| Rohan Karhadkar | 111406429 | rkarhadkar@cs.stonybrook.edu |


## Contents

---

# 1 Question 1 - Clickstream Mining with Decision Trees

## 1.1 Accuracy and Size

The following table describes the results obtained on running our implementation of the ID3 algorithm:

| Threshold | Training accuracy | Test Accuracy | Size (Number of nodes) (Leaf + Internal = Total) | Time |
|---|---|---|---|---|
| $1x10^{-16}$ | 80.645% | 74.732% | 41 + 10 = 51 | 58 sec |
| $1x10^{-13}$ | 80.915% | 74.804% | 61 + 15 = 76 | 1 min 3 sec |
| $1x10^{-8}$ | 80.955% | 73.508% | 81 + 20 = 101 | 1min 37 sec |
| 0.0001 | 81.0875% | 74.016% | 173 + 43 = 216 | 1 min 55 sec |
| 0.0005 | 81.135% | 73.768% | 205 + 51 = 256 | 2 min 4 sec |
| 0.001 | 81.21275% | 73.484% | 245 + 61= 306 | 2 min 6 sec |
| 0.005 | 81.42% | 72.556% | 353 + 88 = 441 | 2 min 52 sec |
| 0.01 | 81.5975% | 72.772% | 433 + 108 = 541 | 2 min 56 sec |
| 0.05 | 83.19% | 70.504% | 2553 + 638 = 3191 | 6 min 54 sec |
| 0.07 | 83.84% | 70.104% | 2949 + 737 = 3686 | 8 min 3 sec |
| 0.1 | 85.0675% | 68.84% | 3885 + 971 = 4856 | 10 min 26 sec |
| 1 | --very high-- | --- | --- | 10 hrs 25 min (estimated) |

## 1.2 Analysis

From the table given above, we observe that as the value of the threshold increases:

1. The number of nodes pruned decreases.
2. The size of the tree increases exponentially.

When the value of the threshold is provided as 1, it means that the tree is not pruned at all. Hence, the size of the tree will be of the order of $5^{274}$ in the worst case.
We measured the amount of time it takes to complete generating one branch at depth 5, which was 1.5 minutes. Hence, the total time taken will be:

$$\begin{aligned} T &= 5^4 * 1.5 \; minutes \\ &= 5^4 * 1.5 / 60 \; hours \\ T &\approx 10.45 \; hours \end{aligned}$$

Therefore, we have not generated the entire tree for the threshold value of 1.

**Training accuracy:**

1. We observe that as we increase the threshold value, the training accuracy increases.
2. This means that as the tree's size increases, its fits the training data better, but at the cost of more memory and more training time.
3. Similarly, if the threshold is low, then the pruning increases, but at this point, if we are unable to assign label, we count the number of positive and negative examples, and assign ‘T’ or ‘F’ to the node according to the higher of the two values.

**Testing accuracy:**

1. The maximum observed accuracy on the test set is 74.804% for a threshold of $1x10^{-8}$.
2. As we increase the threshold value, the accuracy on the test set tends to reduce due to overfitting.
3. As we decrease the threshold value, the accuracy tends to reduce because the tree is not able to generate enough nodes to make a meaningful decision.

**Improving accuracy:**

Initially, we had assigned ‘T’ to all the examples by default if the chi squared stopping criteria was satisfied. However, this was resulting in a very low accuracy of 43.048% for a threshold of 0.05. Therefore, if the stopping criteria is satisfied, we count the number of positive and negative examples, and assign ‘T’ or ‘F’ to the node according to the higher of the two values.

# 2 Question 2 - Spam Filter

## 2.1 Accuracy

The following is the accuracy we achieved with various smoothing techniques:

| Parameter | Accuracy |
|---|---|
| Additive Smoothing with Smoothing Parameter = 18 | 93.7 |
| Without Smoothing | 92.4 |
| Without Optimization | 89.3 |
| Additive Smoothing with Laplace Smoothing | 47.7 |
| Laplace Smoothing | 46.9 |

## 2.2 Analysis

From the above table, we observe that the accuracy depends on the smoothing parameter. With Additive Smoothing, we ensure that no word has a probability of 0 so that all the words contribute to the final probability.

Also, the accuracy without smoothing depends on multiple parameters such as the length of the words, if the words are commonly used words (at, the, for, etc), if the word is a number, etc. Also, to consider only relevant data, we have added a threshold for the count of each word.

The optimizations we implemented while doing this are as follows:

1. We ignored the counts of words which were composed of only digits (i.e. numbers) while classifying emails.
2. If the length of the words is less than 4, then we ignored them because such words are usually common words which do not affect the context of the emails to a large extent.
3. We created a list of commonly used words (stopwords) and ignored them during classification, as they also do not affect the context of the emails to a large extent.
4. We scaled down the frequency of outliers to the maximum threshold value.

5. If the probability of a word of being a spam is near 0.5 (between 0.48 and 0.52 in our case), we have ignored such words for classification purpose as these words are neutral words, which do not contribute to whether or not the mail is a spam.

The maximum accuracy achieved in our classification model is 93.7% for Additive Smoothing Parameter of 18. We tested the classification model by trial and error for various values of this parameter, and found that a value of 18 gives us the highest accuracy on the given test set.

## 2.3 Extra Credit

Useful features to improve classifier accuracy:

1. Binary variable indicating whether the count of any word is above a particular threshold. (For example, if the word “pizza” appears more than 100 times in an email, it is very likely to be spam.)
2. Binary variable indicating whether the email contains words like “won”, “lucky winner”, “dollars”, “selected for”, “free” etc., indicating that you have won something worth very high value. Such emails are very likely to be spam.
3. Binary variable indicating whether the email has a subject line or not. Emails without any subject line are likely to be spam.
4. Binary variable indicating whether the contents are in all CAPS.
5. Binary variable indicating whether unusual characters such as exclamation marks and semicolons are present in the email with high frequency.
6. Binary variable indicating whether the receiver’s name is in the sender’s contact list.

## 3 Contribution

| Name | SBU ID | Contribution |
|---|---|---|
| Charuta Pethe | 111424850 | ID3 algorithm implementation, code optimization, testing, report |
| Deven Shah | 111482331 | Naive Bayes algorithm implementation, parameter optimization, report |
| Nikhil Mehta | 111471539 | Naive Bayes data smoothing, code optimization, report |
| Rohan Karhadkar | 111406429 | ID3 - chi squared implementation, code optimization, testing, report |