Artificial Intelligence
Report for Project 05: Machine Learning

1. Clickstream Mining with Decision Trees:

For each value of threshold, what is your tree's accuracy and size (size equals number of internal nodes and leaves)? What do you observe? If all your accuracies are low, tell us what you have tried to improve the accuracies and what you suspect is failing.

Given below is a table of the Decision Tree's accuracy and size for different p-values:

p-Values	Tree's Accuracy	Tree Size (Total number of nodes)
0.01	0.7396	186
0.05	0.74668	331
1	0.74832	1371

We observe that as the p-Values increases the number of nodes increases. This is logical, as an increased p-Value means the tree levels don't get pruned compared to low p-Values. So, there will be an increase in the total number of nodes in the tree.

We also observe that the Tree's accuracy increases a bit on the test data set as the p-Value increases. This should not hold for a larger data set as a higher p-value implies that the decision tree gets a chance to learn all the noise associated with the data and hence may overfit itself during the learning phase. So, for other test datasets the decision tree that learned with a higher p-value may result in a lower accuracy due to overfitting in the training dataset.

2. Explain which options work well and why?

Since accuracies for p-value=0.05 and p-value=1(full tree) do not have much difference, we can say the accuracy approximates when using p-value=0.05, compared to p-value=1. So we can obtain a shorter tree and yet a decent accuracy using lower p-values.

2. Spam Filter:

In this part of the assignment we are asked to build a Bayesian classifier which correctly identifies spam and ham emails. In order to build the Bayesian classifier, we first extracted all the data from the training set and divide it into two separate classes for spam and ham. Then for each email present in either spam or ham we then count the instances of each word and store it in a dictionary. Similarly, for the Ham emails we store the words present in the email.

Now to classify whether a given email is spam or ham we calculate the conditional probability of each word present in the email for both the classes and multiplied them to assign a particular weight to the given email for each class. Whichever class has higher weight will give us the label of that email.

While considering the probability of each word present in the email since the values are actually small we take the logarithm of the values so as to avoid underflow. We have applied Laplacian smoothing to help solve the problem of unknown words being present in the test set email.

The probability that a given email is Spam or Ham is given below

$$Pr(Spam) = \frac{Number\ of\ Spam\ Emails}{Total\ Number\ of\ emails}$$

$$Pr(Ham) = \frac{Number\ of\ Ham\ Emails}{Total\ Number\ of\ emails}$$

The probability that a word is present in the Spam email is given by

$$Pr(W \mid Spam) = \frac{Number\ of\ times\ word\ W\ occurs\ \ in\ Spam\ emails}{Total\ Number\ of\ words\ present\ in\ all\ the\ spam\ emails}$$

However, this probability turns out to be really small and cannot handle unknown words present in the new emails. We make use of Laplacian smoothing which gives us the value of

$$Pr(W \mid Spam) = \frac{Number\ of\ times\ word\ W\ is\ present\ in\ Spam\ emails\ +\ alpha}{Total\ Number\ of\ words\ present\ in\ al\ the\ spam\ emails\ +\ (Distinct\ words\ present\ *\ alpha)}$$

Here, alpha is the Laplacian smoothing parameter and we have considered it as 10. We have tried alpha values as 0.01, 0.1, 1 ,10, 100 in order to identify improvements in the predications if any. Here are the values that we got.

Alpha	Accuracy	F1 Score
0.01	90.2%	0.874680
0.1	90.2%	0.874680
1	90.2%	0.874680
10	90.4%	0.876606
100	90.4%	0.874015

We can see that the F1 Score increases from alpha=1 to alpha=10 and then drops again from alpha=10 to alpha=100. Hence, we can consider alpha = 10 as our optimal Laplacian Smoothing parameter.

Now according to Bayes theorem, we have

$$Pr(Spam|W) = \frac{Pr(W|Spam) * Pr(Spam)}{Pr(W)}$$

By applying the chain rule, we then obtain the cumulative probability whether a given email is spam or ham.

Thus, by using the above equation we can calculate whether a given a email is spam based on whether the W is present in it or not.

The confusion matrix generated for the given test data file.

Actual / **Predicted**	**Ham**	**Spam**
Ham	341	79
Spam	17	563

Accuracy obtained on the given test dataset is 90.4% with an F1 score of 0.876606.

Contribution:

Harsh Wardhan Agarwal (111465389):

Implemented the Naive Bayes classifier algorithm.

Rajeev Sebastian (111486045):

Made use of Laplacian smoothing to check for improvements in the accuracy.

Rahul Bhansali (111401451): Implemented the Decision Tree ID3 algorithm, Chi-Square helper function and entropy function.

Hae-Na Lee (111207004): Implemented the ChooseBestAttribute function, argument parsers and debugged various code bugs and checked for improvements in the accuracy of the Decision Tree. Helped in implementation of ID3.