TASK 1 Stopping Criterion: Chi-squared criterion

1. Observations for each value of p

P	Accuracy	Tree size
0.01	74.924	186
0.05	74.688	331
1.0	74.431	1371

Observation:
As P's value increases number of nodes increases, we can conclude that lower the P value more the chances of pruning the tree and reducing the depth of the tree.

2. Which option works well and why?

Lower the P is better. Because, tree is pruned earlier if the behavior is not as expected and same feature is tried later at other place giving better prediction.
If we wouldn't have pruned the tree and have forced attribute at that location then

Task 2 Spam Filter

- We built a Naïve Bayes classifier for classifying email as ham or spam. We used the multinomial naïve bayes classifier.
- Libraries used:
    1. argparse – for parsing command line arguments
    1. pandas – reading csv, creating dataframes, writing output to csv
    1. math – calculating log
- Formula for calculating label:

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

Now, the conditional probabilities of each word can be very small, and python cannot calculate power of such a small number. So, we represent the formula in log-space:

$$\log p(C_k \mid \mathbf{x}) \propto \log\left(p(C_k)\prod_{i=1}^{n} {p_{ki}}^{x_i}\right)$$
$$= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki}$$
$$= b + \mathbf{w}_k^\top \mathbf{x}$$

- Accuracy:
87 % (without smoothing)
Smoothing Parameters:

Laplace Smoothing

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d),$$

i. With alpha = 1, and d = number of samples: 89.2%
i. With alpha = 1, and d = square of number of samples: 91.4%
i. With alpha = 2, and d = number of samples: 90.1%
i. With alpha = 2, and d = square of number of samples: 91.3%

The best accuracy which we got was using (ii), that is, alpha = 1, and d = |sample size|², which is 91.4%.

- Contributions

SBU ID	Name	Contribution
111511679	Bhushan Sonawane	ChiSquare, maxInfoGain, Tree construction, Debugging
111447198	Nishant Borude	MaxInfoGain, Tree construction, Pandas, data parsing, debugging
111462188	Mihir Chakradeo	Data parsing, Training, Smoothing
111447727	Aditya Yele	Calculating conditional probabilities, Training, Testing