Assignment 3 Project Report

CMPT 310
Simon Fraser University
Spring 2021
Names: Sumrit Sanghera, Pawan Gill, Yevhenii Strilets, Nicholas Hagan and Caleb Whitehead

Exploration

Our group chose to explore the data using decision tree learning. We started by following the article provided to us. The only attribute selection filter this included was the CfsSubsetEval with best first search. Running the J48 classifier on the resulting nodes yielded poor and inconsistent accuracy. The runtime for the attribute selection was also quite long. We also noticed that the remaining attributes were frequently punctuation, single characters, or fragments of contractions. We thought we might get better results if we focused on whole words instead, so we experimented with using different tokenizers and stopwords in our StringToVector. This only decreased accuracy. Eventually, we stopped trying to get different attributes, and began working on identifying the useful parts of the information we had. This led us to filtering attributes using InfoGainAttributeEval. After this, we tried running the CfsSubsetEval on this smaller set of attributes, and we got much better results. Changing parameters within our classifier did affect accuracy, but the changes were usually minor compared to the effects of preprocessing. Furthermore, with one exception, the effects were often inconsistent between our individual trials.

We achieved an average of 85.9% accuracy with the following process:
Pre-processing Filters Applied:

1. StringToVector
2. AttributeSelection
    a. evaluator: InfoGainAttributeEval
    b. search: Ranker
        i. numToSelect: 200
3. AttributeSelection
    a. CfsSubsetEval
    b. BestFirst

Classifier:

1. J48 algorithm
    a. confidenceFactor: 0.25
    b. minNumObj: 3

Explaining Final Results

The basic StringToVector was used to separate the text into individual attributes each composed of a single word or character. We then ranked each attribute using the AttributeSelection filter with the InfoGainAttributeEval setting and kept only the top two hundred results. This forced the J48 algorithm to only examine attributes that would provide a relatively high amount of information gain. We also used the CfsSubsetEval with BestFirst to eliminate redundancies amongst our two hundred attributes. This prevented J48 from examining multiple attributes that provided similar information during discrimination. Configuring our classifier to require at least three objects at a leaf added a minor amount of accuracy to our results. This likely helped curb the effect of outliers by discounting the cases where only one or two instances reached a leaf. The effect of this change can be seen in figure 1.

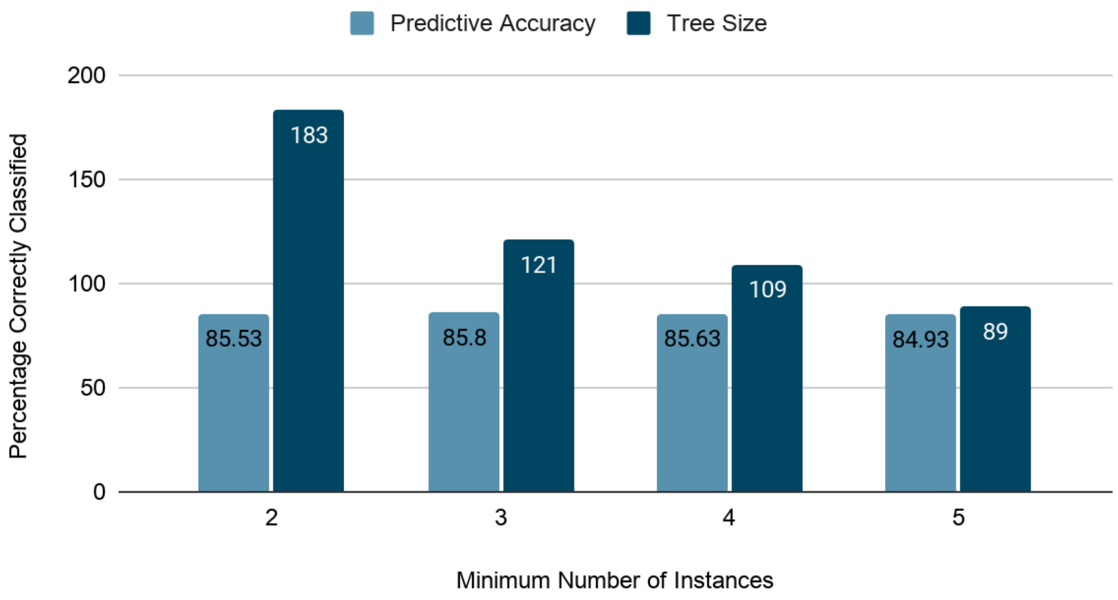

Figure 1: Minimum Number of Instances vs Accuracy

Insight

The root node is “share” and this is significant because in many of the fake news articles, there is usually a prompt to share the article, and spread the fake news. This could either be directly part of the article, or a “share to facebook” link at the beginning or end of an article. Therefore this does seem true that an instance of the word “share” would result in the article being fake.

Fake news articles tend to have hyperlinks in them, so they can be spread with more ease. That is why the nodes “https”, “-”, and “--” are an indicator of a fake news article.

Conjunctions of feature values (where there is low entropy in the decision tree) make sense in most of the cases. Many of the nodes relate to the US presidential election. A surprising informative conjunction could be at the node “email”, which suggests that if that word is present in the article, then it is practically guaranteed to be fake.

Some of our nodes were single letters such as “s” or “I”. These letters could obviously be in either real or fake articles, therefore this could be a reason for discrepancies.