Project 8 - Decision trees and neural networks

Caitlin Thaxton & Bekah Grant

Presentation
https://docs.google.com/presentation/d/1c7VGWtBSseHVsMqKrVsjG7BtHFkOLwcsZNeM24aoCkE/edit?usp=sharing

Project Folder
https://github.com/bekahgrant33/cs5830_project8

Introduction

Online news platforms play a crucial role in getting information to the public. There is so much content that is getting published to the internet, so how do you get your information to go “viral”? In our analysis we aimed to understand the factors that contribute to an article's popularity. This information would be very beneficial to publishers, writers, content creators, and marketers, who would be interested in knowing information such as what to write about and how to format and advertise news articles to promote popularity.

The goal of this project was to create a model that would classify whether an article will receive high or low engagement (defined by the number of shares). Through analyzing information from about 40,000 news articles we were able to train and fit decision tree and neural network models to successfully predict popularity.

Dataset

For this project, we used a dataset about Online News Popularity from the UC-I Machine Learning Repository. This dataset includes information for nearly 40,000 online news articles from Mashable. Data was acquired on January 8, 2015 and includes articles from the years 2013–2014. Information includes features such as the number of words, keywords, number of links, data channel, day of the week of publication, etc. There are 60 features, so we selected a few features based on perceived ease of use, what we found interesting, and what we thought would be good predictors. We decided on using the features that indicated the number of words, number of keywords, whether the article was published on a weekend, the data channel, the average number of shares on referenced articles, and the polarity of positive and negative words. The dataset also includes the number of shares, which we used to determine whether an article was popular (if it had at least 1,400 shares). This dataset is related to domains such as journalism and marketing and can be helpful by allowing people in these domains to determine what types of articles will gain popularity.

Analysis Technique

We analyzed the data using both decision trees and neural networks. Both of these techniques are suitable for classification, which is what we were doing with this dataset by trying to predict whether an article would be “popular” or “unpopular".

In order to compare potential models, we created three decision trees with different maximum depths and three neural networks with different numbers of hidden layers/nodes per layer. We also used visualization tools to map what the trees and networks looked like. For the decision trees, the visualization helped us look at which attributes the algorithm decided to make decisions on and compare the bias and variance

from the models with different depths. These comparisons could help us determine which features are more influential in predicting popularity as well as whether those predictions are likely to be generalizable.

Neural networks in Python have difficulty converging if the data is not normalized, so we had to pre-process the data. We pre-processed the data by scaling our data to accommodate the sensitivity of the multi-layer perceptron. After scaling the data for standardization we were able to train the model. Then we were able to fit the training data in order to make predictions and evaluate the test data. The neural network analysis model was able to identify patterns in the data that influence article popularity based on predicted number of shares.

Results

Decision Trees:

The decision trees for the dataset resulted in the following. We created three decision trees with maximum depths of 3, 4, and 5.

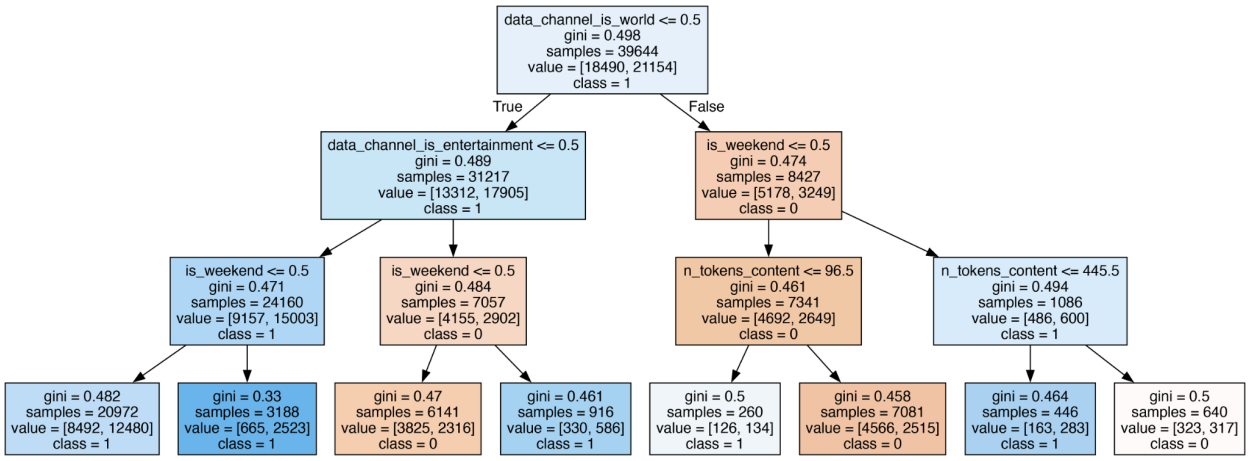


Figure 1.1: Decision Tree with maximum depth = 3

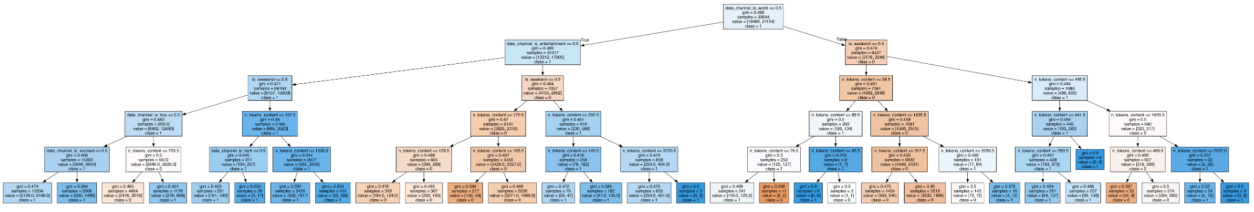


Figure 1.2: Decision Tree with maximum depth = 5

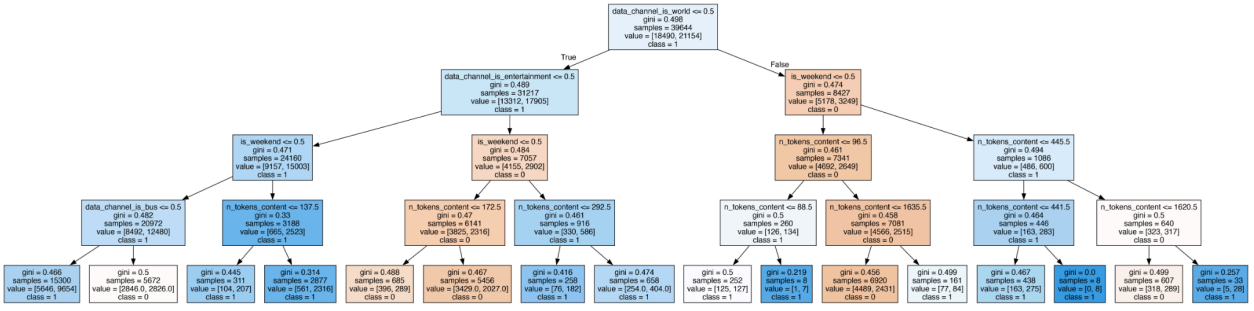
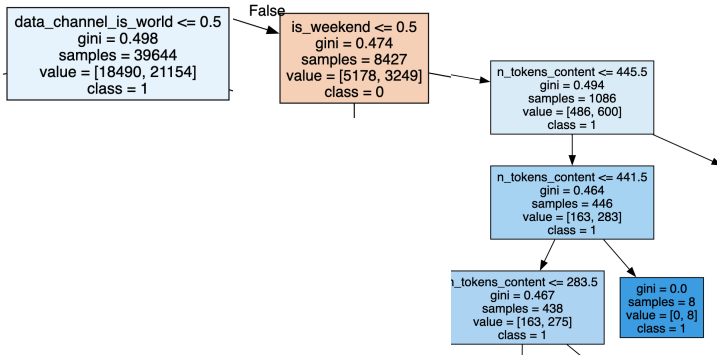


Figure 1.3: Decision Tree with maximum depth = 4

One interesting thing to note is where the trees decided to split the data. Out of the features we chose, it appears that whether the data channel is “World” is the best predictor, where articles that aren’t in that category are more likely to be popular. Whether the article was published on the weekend also appears quite high on all the branches, where articles that are published on the weekend appear to be more popular.

We also were able to compare the trees of different depths. The deeper the trees, the more we get strong predictions, but there are also big concerns of overfitting. For example, with the tree with a maximum depth of 5, many of the leaf notes with a strong prediction for popularity have very few articles in them. This having been trimmed down so much from the original nearly 40,000 articles raises the concern that these predictions might not be very generalizable because there are so few samples following that pathway. For example, Figure 4 shows a node representing articles that are in the “World” data channel, posted on the weekend, have 445 or fewer words, and have 442 or greater words. It happens that all World articles that were posted on the weekend with 442-445 words were popular, but that isn’t necessarily a guarantee that if another article like that was posted that it will be popular.

Figure 1.4: An example pathway that shows how the tree with max depth 5 is perhaps overfitted



Neural Networks:
To determine whether an article was popular, we decided that if an article had more than 1,400 shares the article would be deemed as popular. The model took into account the length of the title, the length of the article, the number of images, the number of keywords, what genre the article was (i.e. Entertainment, world, tech), the average negative and positive polarity, and if the article was released on a weekend.

The neural network models were trained and fitted to predict whether or not an article would be classified as popular or unpopular based on the number of shares. After finding the model’s predictions our model was about 62%-68% correct in determining if an article was deemed as popular or unpopular.

There are a lot of variables that go into an article's popularity. One major factor in popularity is the publication site. However, we could not account for this because all the articles in our dataset came from the same news source.

Technical

The data required some cleaning before we could perform the analysis. First, each of the column names started with a space. This caused problems with the target column, so the space had to be removed. Cleaning our data also consisted of dropping unwanted columns such as the article’s url, number of links, rate of non stop words and other unwanted features.

Decision trees and neural networks were appropriate for this dataset as we were using numerical data to classify each item into groups. However, the dataset was originally built for regression, so we had to decide on a threshold for sorting the shares into popularity. At first, we thought it would be best to divide the shares into the bottom 75% and top 25%, but that resulted in getting very few predictions for “popular” articles. So, we instead decided to define the threshold at the median of 1,400 shares.

The most difficult part with the decision trees was setting up the visualization tool, but once it was set up, it was fairly easy to create trees and compare them.

When using Neural networks you have to thoroughly prep and pre-process the data you are given. The dataset that we found had about 60 features, so we reduced the number of features for simplicity and better runtime.

The neural network in python has difficulty converging if the data is normalized as described in the analysis technique portion of the report. To pre-process the data we used sklearn’s built-in StandardScalar() in order to scale our data to better our model.

Using SciKit’s Multilayer Perceptron Classifier model made training our model fairly simple. After initializing the MLP model we tested different architectures by changing the hidden layer parameter. The dataset has 14 input nodes and 2 output nodes. The output nodes are 1 for is popular and 0 for is not popular. We tried the hidden layer to be 5, which can be visualized in figure 2.1.

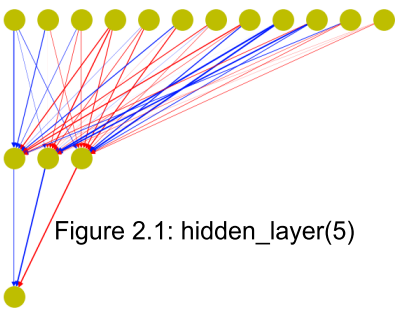


Figure 2.1: hidden_layer(5)

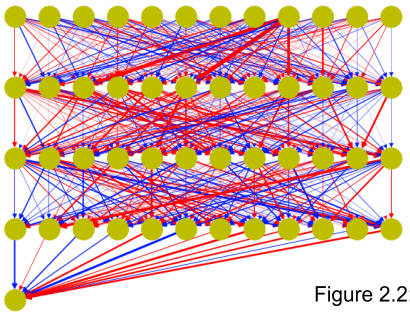
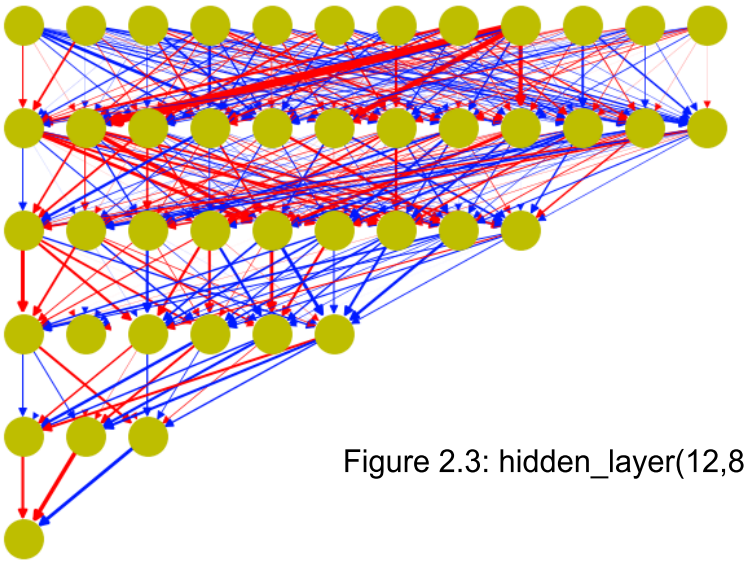


Figure 2.2: hidden_layer(12,12,12)

We also had a figure with hidden_layers (12,12,12) which is visualized in figure 2.2.
. No matter the altercation to the architecture the results were similar.



We added another dimension, thus we tested the model with a hidden layer of (12,9,6,3). This resulted in a s-score of 0.62 for classifying unpopular articles, a 0.67 for classifying as popular articles. This neural network is visualized by the graph, figure 2.3

Figure 2.3: hidden_layer(12,8,6,3)