Machine Learning Engineer Nanodegree

Capstone Proposal

Thomas Kauth
May 30, 2018

Using Machine Learning to Predict Online News Article Shares

Domain Background

It's no secret that the world of newspapers, magazines and broadcast news has been upended by the advent of the internet. Where people once learned of the events of yesterday in the morning paper, they now read about events as they are currently unfolding by getting their news from the internet. Where authoritative opinion was once the domain of professionals with auspicious credentials armed with facts, that space has been invaded by the modern day equivalent of carnival barkers and snake oil salesmen, armed with click bait. And it's not just politics. Every category of news has been affected by the internet. The Wall Street Journal cannot ignore the various sources of business news on the internet, and People Magazine must know that by the time an article about Rihanna hits the newsstand, most readers already know the news already.

All traditional news sources have undergone some degree of transformation, and many are still struggling with relevance and profitability in the 21st century. They all have some online presence, of course. So across the sphere of the internet, they must compete with dozens of non-traditional news sources.

One relevant question that all news sources should be asking is, what makes a news story popular? This is not to suggest that news sources should all merely pander to whatever it is that the news reading audience finds most exciting or provocative. Instead, they should be arming themselves with data about what it is that appeals to readers for simple editorial decisions, e.g., the length of the title, length of the content, positive/negative words in the content, etc. Simple decisions such as these may drive popularity more than writers and editors realize.

Problem Statement

To gain insight into why a news article can be considered to be popular, we need a target measurement. Page views are one metric, but it is impossible to know if the reader read the article and liked it, read it and hated it, or read the first paragraph and then gave up. A better metric is user shares. If a user is sharing the article, that is a very good indication that that user enjoyed the article enough to share it with friends or colleagues.

Additionally, we also need metrics describing the content of each article in objective terms. If we are to answer the question of why one article was more popular than another, we need a

way to measure the different qualities of the articles – i.e., we need a number of objective descriptors of the contents of the articles in the dataset.

With good descriptive categories and a target measurement, we can then apply a number of classification machine learning algorithms to tease out the qualities that make a news article popular. This could then lead to better decisions on the part of writers and editors of news articles.

Datasets and Inputs

The dataset for this project is posted at the UCI Machine Learning Repository. The news articles were all posted to the website Mashable during a two year period ending in 2015, but the original content is not part of the dataset. Instead, researchers Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez[1] analyzed the content to create the dataset, which contains 61 attributes, such as:

- number of words in the title
- number of words in the content
- average length of words in the content
- day of the week the article was published
- text sentiment polarity
- number of images
- number of videos
- number of shares (target variable)

The Mashable news articles span a number of categories, such as Entertainment, Business, and Technology. All of the data extracted (except the target) is data that would be known pre-publication. The authors use natural language processing to compute several interesting fields related to sentiment polarity and subjectivity.

Solution Statement

I plan to evenly divide the training data into two parts based on number of shares. Then I will use a number of different classification techniques to find the best solution for predicting which articles in the test set will get better than average number of shares, and which articles will get worse than average number of shares. I plan on trying KNN, AdaBoost, SVM, random forest, neural networks, and possibly others.

1 K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

Benchmark Model

Fernandes, Vinagre and Cortez in their study achieved an accuracy of 67%, an F1 score of 0.69 and an area under the Receiver Operating Characteristic curve of 73%. This was achieved using a rolling window approach (10,000 training samples vs 1,000 test samples). I will be taking a more traditional approach of dividing up the 39,000+ dataset into training, validation and testing sets. Nevertheless, I aim to achieve similar results.

Evaluation Metrics

I plan to use the same evaluation metrics that Fernandes, Vinagre and Cortez used in their study – accuracy, F1 score and AUC of the ROC. And like Fernandes, et al., I plan on undertaking a study of feature importance. In their study Random Forests was found to be the most accurate classifier, and they used the feature_importances method of the Random Forests classifier. I plan to do the same type of study, depending on which classifier returns the most accurate predictions. I will then compare my rankings to theirs. The results should suggest insights into what features a news writer or editor could modify to garner more interest from readers and increase readership.

Project Design

My first step will be to examine the dataset and check for any missing values. If I find any, I will decide if it would be best to leave them alone, fill in a mean value, or possibly eliminate the feature(s).

Following that I will divide the data into training, validation and test datasets.

Then I will perform any needed data transformations on the training data such as re-scaling or one-hot encoding.

Next I will undertake an Exploratory Data Analysis, visually exploring the features themselves, creating histograms to check their distributions, checking their relative scales, seeing if anything stands out. Depending on what I find, it may be beneficial to do some feature engineering – combining features or finding a way to extract more information from them.

Then it's on to training. Using gridsearch, I'll try a number of different classifiers (KNN, AdaBoost, SVM, random forest, neural networks, etc.) to see which one performs best. Once I've found the top classifiers, I'll modify their hyperparameters to further narrow down which one is the best. Once I believe I've found the best classifier, I'll continue to modify the hyperparameters to come to a conclusive decision. Hopefully my results against the test set will be close to those achieved by Fernandes, et al. (though their rolling window approach may not yield comparable results).

Lastly, I'll extract the feature importance to see which features are judged to be the most important when targeting a high number of shares. This data is what Mashable's writers and editors could use to increase engagement with their readership, and it may possibly have value well beyond the writers and editors of Mashable.