Machine Learning Engineer Nanodegree

Capstone Proposal

Tyler Lanigan
November 7th, 2016

Domain Background

The quality of the comments made on internet forums has always suffered due to anonymity of it's users. When users comments on, for instance, a YouTube video, there is no repercussion for what they say, and the dialogue generated is often not helpful. Different websites have tried different methods for extracting more useful comments. Reddit uses an upvote system, Quora fosters a community that values high quality responses over low quality ones, and Amazon allows for it's users to to rate the "helpfulness" of reviews left on their products. Amazon's system, in particular, then allows for the higher rated comments to be displayed at the top of the review forum so that new users can see the top rated comments in order to help them make their own purchasing decisions.

Even though Amazon's helpfulness rating system seems to work on the surface level, poor quality comments still seem to be at the top of their review forms. Having poor quality reviews hurts Amazon's business, as a major reason that people are willing to buy consumer goods online without seeing the items themselves, is that they have access to others peoples opinions of the item. For example, a review at the top of an app called "Friday Night at Freddies 4" is as follows:

> "I love this game so much but at first I though it was lame but when I go in the game I can't beat the first night because cause I put it to full volume and I can't here the breathing bonnie strike at 4 am Chica at 5 and plus it not lame it's better than fnaf and fnaf 2 plus get this game when u buy fnaf"

This comment, despite being at the top of the forum, is difficult to understand, a run on sentence, and full of spelling errors. The reason for the failure is part of the algorithm for determining the order of the reviews relies on how recently the review has been made. The offending review was the most recent, but it's helpfulness score was far less than previous reviews. This illustrates the difficult balance that must be struck between showing the highest rated reviews, and showing the newest reviews, to be rated by the community. An ideal system would predict if a review is helpful or not, so that poor quality reviews would not need to be displayed the top.

Problem Statement

The problem being addressed in this project is the poor quality of Amazon reviews at the top of the forum despite the "helpfulness" rating system. The problem arises from the "free pass" given to new reviews to be placed at the top of the forum, for a chance to be rated by the community. The proposed solution to this problem is to use machine learning techniques to design a system that "pre-rates" new reviews on their "helpfulness" before they are given a position at the top of the forum. This way, poor quality reviews will be more unlikely to be shown at the top of the forum, as they do not get the "free

pass” because they are new. The proposed system will use a set of Amazon review data to train itself to predict a helpfulness classification (helpful, or not helpful) for new input data.

Datasets and Inputs

The dataset used for this project is provided by the University of California, San Diego at a download link on their website (also referenced below). As the original dataset is massive, the problem will be formulated in the context of the Apps for Android subset of data, that is available as a separate download. The data is provided in “json” format and will be converted to a pandas data frame for use in this project. Each review in the data frame has the following information:

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. [2,3]
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

There are 752,957 reviews in total. For our problem we will use the reviewText to generate features using natural language processing.

The ‘helpful’ score can be explained as follows: a user can either rate the review as “helpful’ or ‘not helpful’. The dataset records each of these in an array. For our problem we want to classify the email as either ‘helpful’ or not ‘helpful’. For training, this label can be generated by dividing the ‘helpful’ ratings by the total ratings and seeing if it exceeds a certain threshold (e.g 0.5). When we are testing, we will try to predict this classification using features from the test reviewText.

Solution Statement

The proposed machine learning system will use the Android App Amazon ‘reviewText’ data and natural language processing to train itself to predict a ‘helpful’ or ‘not helpful’ classification for new input data. For training labels it will use the ‘helpful’ scores present in the dataset to generate a binary classification. The system will be tested by comparing it’s prediction to a held-out testing set of data from the same dataset. The metric for determining the success of the system will be the roc_auc and is discussed in more detail below.

Benchmark Model

The benchmark model used for our solution will be the result gained from randomly guessing if a review is helpful or not helpful.

Evaluation Metrics

Since our problem is a binary classification problem (helpful or not helpful). We will use the ‘Receiver Operator Characteristic Area Under the Curve’ or roc_auc score. The curve is created by plotting the

true positive rate (TPR) against the false positive rate (FPR). The area under the curve is used to give a score to the model. If the area under the curve is 0.5, then the TPR is equal to the FPR, and the model is doing no better than random guessing. A perfect model would have an AUC of 1.0, meaning it as 100% TPR. The equation for the AUC of the ROC is as follows:

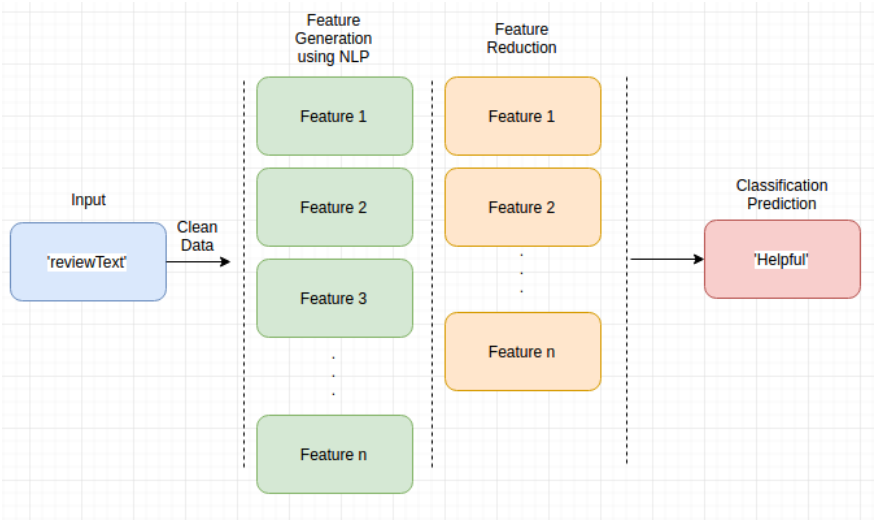
$$A = \int_{\infty}^{-\infty} \mathrm{TPR}(T)\mathrm{FPR}'(T)\, dT$$

Project Design

A basic work-flow is presented as follows:

1. The data must be obtained and parsed into the correct format. In our case, the data will be placed in a pandas data frame as this is a commonly used data type in machine learning.
2. The data will be cleaned by stemming, setting to lowercase, removing any null entries, punctuation and weird characters/html tags. Other columns not used in the design of the model will be removed, leaving only the reviewText (to generate features) and the 'helpful' score (to generate labels.
3. Features will be generated using natural language processing techniques. Features could include 'number of spelling mistakes', length of review, or amount of capitals in a row. Additional features could be generated using Term frequency and inverse document frequency (TFIDF).
4. The binary classification label will be generated by dividing the number of "helpful' rating over the total amount of readings and seeing if it exceeds a certain threshold.
5. The data will then be shuffled and split into training_features, training_labels, test_features and test_labels.
6. As this system will be a real time system editing the reviews as they come in performance must be taken into consideration. Therefore a feature reduction stage will take place. Methods used could be PCA or select K-best.
7. A classifier will be trained on the training_features and training_labels. Different algorithms will be looked at in order to determine which works the best including: logistic regression, boosting algorithms and neural networks.
8. The classifier will be tuned using a parameter grid and k-fold validation.
9. The classifier will then be used on the test set's 'reviewText' to to try and predict a 'helpful' or 'non-helpful' classification. This result will be compared to the label generated in step 4 using the auc_roc. The result will be a classifier that can predict if a review is helpful or non-helpful using the review text alone.

The following figure shows this work flow graphically:



References

1. Inferring networks of substitutable and complementary products. J. McAuley, R. Pandey, J. Leskovec *Knowledge Discovery and Data Mining*, 2015.
2. Image-based recommendations on styles and substitutes J. McAuley, C. Targett, J. Shi, A. van den Hengel *SIGIR*, 2015.