CSc 180 - Intelligent Systems

Alec Resha

Project 1 Report

Problem Statement

The primary problem addressed by this project was how to use machine learning to analyze business based on their reviews. To accomplish this, the text from reviews was used to train along with the star value of the review. To estimate the score of a single business, we will estimate the review of each review then take the mean of all the predicted review scores to estimate the score of the business.

Methodology

To begin analyzing review text, the first step is to narrow down the data if necessary. In the case of yelp reviews, the data set contained far more information than what would be useful. The data from the reviews and businesses was trimmed down to only include The business id, business categories, business name, review stars, and review text. The first four are solely used for identifying results and giving a more readable output, such as business name, predicted rating, and actual rating. To analyze the review text, it is first vectorized to turn it into a useful frequency analysis format. The stars data was useable as is for training and testing. Once the text was in a useful format, the text and stars were split into a training and test data. For the implementation of this project, 100,000 reviews were used for training/testing with 1/4 of the data being used for validation and the other 3/4 used for training.

Experimental Results and analysis

To experiment, the number of layers, number of nodes in each layer, layer activation, and optimizer were all varied and the RMSE score was used to judge them. Based on 10 separate tests, the one with the best RMSE score had 8 layers of 20 nodes each and 1 layer with 1 node. The activation used for this test was tanh, the optimizer was adam, and the resulting RMSE score was 0.9789.

Task Division and Project Reflection

This project was done alone so there was no project divison.
I learned a lot while working on this project. It was a really good crash course for getting me back into python programming, but it was primarily a big learning experience for figuring out how to code my own machine learning processes. If I had more time, I would definitely like to expand on this idea. I think with a better computer it would be better to train the algorithm on a larger

dataset than 100,000 reviews and 600 maximum elements, but I continually ran out of memory with higher numbers. Another way that could result in a more accurate algorithm would be to focus on individual categories. It is likely that the breakdown of good or bad words for a specific category would be more useful rather than applying to any kind of business. For example, the identifying words for a good or bad salon will be much different than the good or bad identifying words for a supermarket or charity. An additional useful piece of information for training would be to analyze the time reviews were submitted and weigh recent results more heavily. If a business started with low reviews but improved then the resulting score should be higher, or vice versa.

Handling low memory was one of the largest issues that I encountered. When I ran the program with more review text, it became much more accurate. The problem was certain parts would fail to work since 100% of my computers ram was being used. This resulted in the kernal crashing many times while tuning the amount of results that could be processed.
The other main issue was with predicting models for businesses with few reviews. The predictions were more accurate with a higher max_features, but there are a lot of businesses with 20 to 30 reviews that did not reach that number of unique words and would not be accepted by the algorithm. I got around this by lowering the number of max_features to 600 in increments of 100 until businesses stopped being rejected when calculating their average score.