Final Project: CSE 142

Ian Kirk, ikirk@ucsc.edu, ID:1601671
Jacob Baginski Doar, jjbagins@ucsc.edu, ID:1577517
Julius Fan, jzfan@ucsc.edu, ID:1522743
Group 21

1 Tools Used

All code was written in Python 3. Libraries used include numpy and random.

2 Diversity

2.1 Jacob

I am bisexual and closely connected to many people who are active within the LGBTQ community so while I do not consider myself active in the community, I have grown up with it being important for many people around me. My father is also an immigrant from England so I grew up with parts of the culture in my home life.

2.2 Ian

I strongly identify with my Taiwanese heritage and am often mistaken for being Chinese. I also grew up in a nearly homogeneous neighbourhood composed of mostly wealthy Chinese and Caucasians. Although I was not noticeably a minority, at times I did feel like an “other”.

3 Abstract

Although we all used the same extracted features, each of us implemented a different algorithm to learn a linear boundary. This was then extended for multi-class classification through varied techniques.

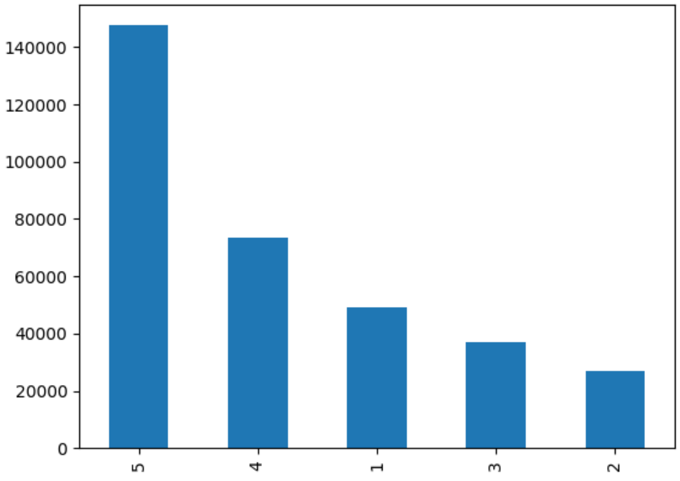

Fig. 1. Rating Distribution

4 Data Pre-processing

As you can see in the figure, we noticed that the extremes for ratings were much more common than the moderates. We chose not to subsample because these distributions were so extreme and we did not want to through away so much data. Although we also considered super sampling to, we ultimately decided against it too because some of our models were already taking a long time to run, and adding more instances to match the quantity of 5-rated instances would hamper run time too much.

5 Feature Extraction

We decided to 80 different features from the reviews. Most of these features were ”keyword” occurrences, and the other very small minority were other various quantities.

5.1 Keywords

Keyword occurrences made up 76 of the 80 extracted features. They tracked the number of occurrences of a particular ”keyword” in the review. For example, one of our keywords was the word ”good”. If the word ”good” appeared 3 times in a review, then that instance's feature vector would the value of 3 in the position that corresponds to the word ”good”. We wrote a short program to determine and the frequency of all words used in all reviews. After running this program, we sorted the words by descending frequency and hand-picked ones that occurred often enough to represent the data set and could signify the outlook of review. Some more keywords were ”great”, ”slow”, and ”expensive”.

We also considered not counting individual keywords, and instead counting the sum total of keywords in a particular group. For example, we might have instead grouped all keywords into one of three categories: positive, negative, and indifferent. With this, instead of having a keyword feature vector of 76 dimensions, it would be 3 dimensions and look like $< good, bad, indifferent >$. If an instance contained ”good”, ”great”, and ”awful”, its feature vector would look like $< 2, 1, 0 >$. In the end, we decided against this because this would have meant our feature vector would have a much lower dimension, and since we were learning linear models only, we wanted it to be more complex.

6 Approaches

6.1 Logistic Regression

6.2 Perceptron

6.3 Naive Bayes

7 Experimental Set-up

8 Results

9 Conclusion

10 Ideas for future work