1. Introduction

1.1. Project Description

For our project, we conducted sentiment analysis on homework review data. By running different sentiment analysis processes on our input data, we were able to analyze subjective characteristics of our input data and extrapolate further information that can be used to draw conclusions and improve the course.

1.2. Motivation for Sentiment Analysis

Sentiment analysis is a way to analyze the mood or emotion in a large collection of text documents. Sentiment analysis is a particularly powerful tool to use because it combines natural language processing and data analysis. By using sentiment analysis on our input data, we are able to extract qualities of our input data and then display those qualities in visual formats like graphs and charts. For our homework reviews data, we are able to break down our results even further by things like homework, topic, or semester. This allows us to easily filter our output based on different audiences we may have or specific problems we are trying to solve.

1.3. Sentiment Analysis in the real world

Sentiment analysis is also used in the real world for many different applications. Some examples of real world application of sentiment analysis include running searches that separate positive tweets about a company from other tweets to better gauge what the company is doing well. Sentiment analysis is powerful because it allows people to assess and quantify opinion data, which in turn can help the company make better decisions. Because of this, sentiment analysis is

used widely used in sales and marketing to determine what the best advertising strategies are. The approach to this problem is outlined below in sections 3-4.2

2. Pre-processing

Before any analysis could be done, the raw data had to be cleaned so that it could be effectively processed and analyzed by the spark applications. The pre-processing involved the following steps: filtering files by length, removing white space and stop words, and stemming. The first step was removing files that were less than 50 words in length, as these files would not be effective to use in analysis. After removing stop words, many of these files would barely have any data to analyze. Sections 2.1-2.3 explain the process of analyzing each individual review. For reference, appendix 8.1 has a sample of the pre-processing that a review underwent.

2.1. Removing White Space and Stop Words

Now that files that were deemed long enough to analyze were removed, it came time to remove excess information that would not prove beneficial to our analysis. Initially white space was removed to make the data easier to process later on and make files more uniform. Then, we removed stop words. Stop words are words that are frequently removed when conducting natural language processing because these words are frequently used within writing in that language. For example, some of our stop words included “is”, “the”, “was”, “and”, “with”, and “it.” Although the pre-given list covered most of the stop words that needed to be removed, there were certain additional words that needed to be removed given the context of our homework reviews. Some examples of these words included “homework,”

and “problem.” “Problem” was extremely important to remove as it would be identified as a negative word, although many students may have been referring to it as “the problem” in a non-negative context. Now that most words had been removed, one problem arose: words like “no” and “not”, which are important for negation handling, had been removed by some our our pre-processing. We had to rerun these steps to make sure that these words were not removed so that we could later do negation handling.

2.2. Stemming

Given the dataset with all stop words removed, in order to make the analysis more accurate, stemming was implemented. Stemming is the process of reducing words to their “root.” For example, the word “amazingly” would be reduced to “amaz” through stemming. The stemming we used was Porter’s stemmer. The value of stemming is that it allows the analysis to combine words whose roots are the same, but might look different due to conjugation or other context. For example, the words “thankfully,” “thanks,” and “thankful” should be considered the same word and assigned the same sentiment score since the only difference is the grammatical structure of a sentence. As we explain later in the evaluation of the model, the model that included stemming proved to be more effective in determining the sentiment of a review.After stemming, the review data was ready to be analyzed. The positive words file and negative words file were also stemmed so they could be used later in the model.

2.3. Formatting

In order to get access to important information like the homework assignment number as well as the "ground truth" of the review, we incorporated these into the review so that they could be easily accessed when assessing the accuracy of the model later on. The format chosen was "HW# \t ground truth \t review that is pre-processed." Now in this form, the data is easily accessible to be analyzed.

3. Local Development: Model Creation and Evaluation

In the local development stage, the analysis aimed to determine sentiments of students in regards to homework in CSE427s Cloud Computing and Big Data Applications at Washington University in St. Louis. The basic component of this section was to analyze homework review data and predict if the review was positive and negative. Section 4 discusses the accuracy and formulation of the model, as well as other model choices that were considered; section 5 discusses extensions to this analysis.

3.1. Sentiment Score Calculations

In order to determine the sentiment score of a given review, the following calculation was used:

$$Sentiment\ Score\ = \frac{positive\ words - negative\ words}{Total\ Words}$$

The sentiment score was different between the sum of the positive words and the sum of the negative words in the review divided by the total words in the review. In order to determine if a word was positive or negative, we used the list of positive and negative words that was

supplied. In Appendix 8.2 there is a sample walkthrough of this calculation. The other calculation that was considered was the following:

$$Sentiment\ Score\ =\ positive\ words\ -\ negative\ words$$

As will be discussed in the evaluation of the model, the sentiment score that divides by total words proved to be more effective in identifying if a review was positive or negative. Due to the way it is calculated, the range of values were [-1,1], where 1 is perfectly positive and -1 is perfectly negative.

The last part of the sentiment calculations involved negation handling. In order to account for phrases like “not good” in which the negation word “not” changes the sentiment meaning of the word “good.” In order to handle this, we stored if a word had a negation word (“not”, “no”, “neither”, etc.) before it, and if it did, the value of the word was multiplied by negative to flip the sentiment of the original score.

3.2. Basic Sentiment Analysis

Since the way of quantifying positive and negative sentiments was established, the reviews were then processed. However, before the model could be completely finished, we had to determine the bounds of positive, negative and neutral homework reviews. See *figure 1* for a plot of the values that our analysis returned.

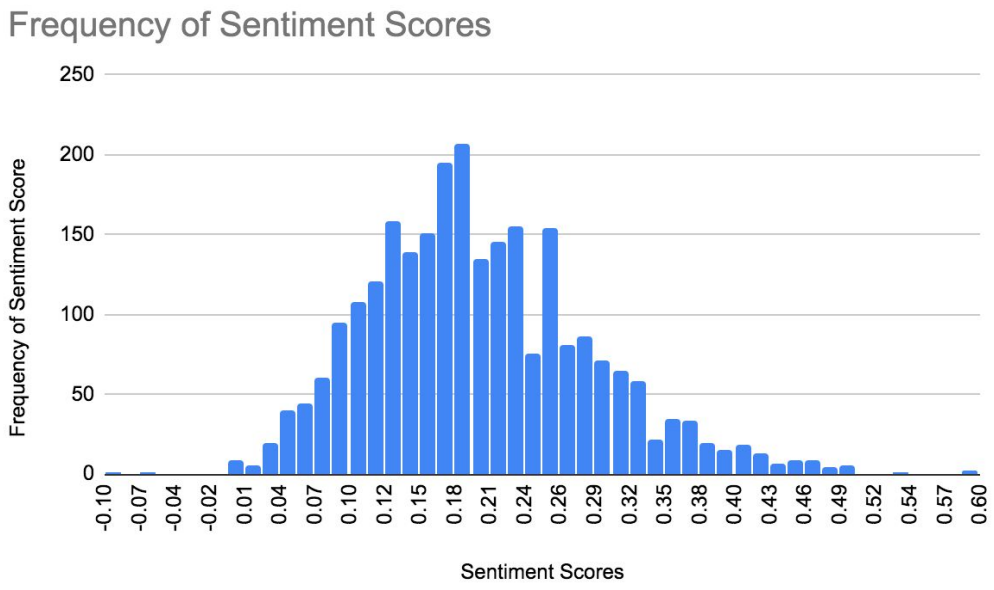


Figure 1 shows the distribution of sentiment scores based on our model.

Although it may have seemed that neutral should take on the value of the mean (~.18) of this normal distribution, we did not select this value since the reviews were skewed more positive (they were not evenly split between the three types). Refer to figure 2 to see the distribution of positive, negative and neutral reviews (using the underlying classification). This was only used to determine where our bounds should be.

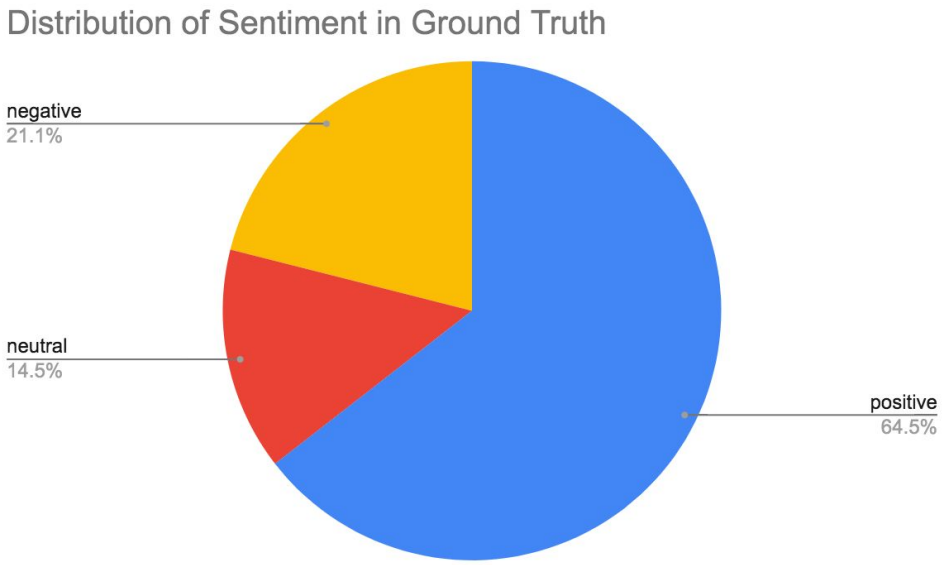


Figure 2 shows the breakdown of positive, negative, and neutral reviews from the ground truths we were given.

Because of this, the following bounds were selected: negative:[-1, 0.04], neutral (0.04, 0.3], and positive (0.13, 1]. Now that bounds have been set and sentiment scores calculated, it was possible to determine the effectiveness of the model.

3.3. Results and Evaluation of Model

3.3.1. Distribution of Sentiments

After running the data on the spark application, the first test we ran was to check to see if the distribution of our sentiments roughly matched the ground truths we were given. We found that with the bounds given, our model matched a similar distribution of the ground truths. See figure 3 for the breakdown from our model.

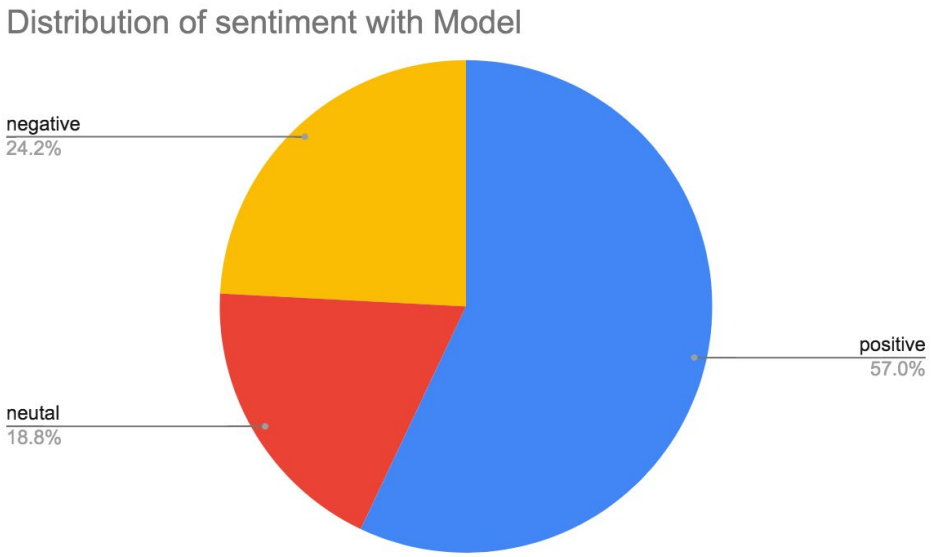


Figure 3 shows the breakdown of sentiment scores based on our model.

Since the two distributions are roughly the same, we were comfortable moving forward since our model identifies positive and negative sentiments at the appropriate rate given the input data.

3.3.2. Success of Predictions

Once comfortable with the bounds, we proceeded by analysing the accuracy of our model in determining the sentiment of a review. In order to do this, we calculated a percentage. For the final model selected, the percent of reviews correctly identified was 62%. Figure 4 depicts the accuracy of the final model:

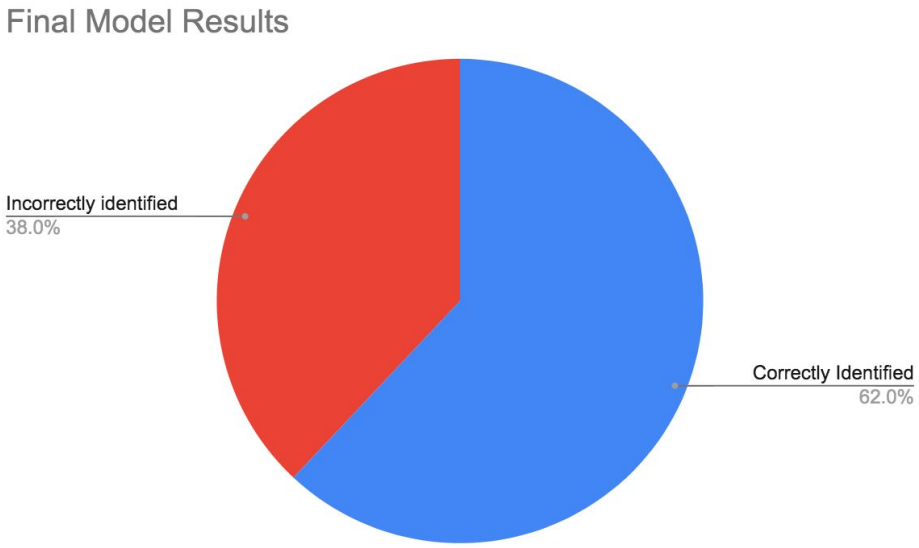


Figure 4 shows the overall accuracy of our model.

Since this model did not meet the 70% threshold given, other model variations were tested to see if they could improve the result, but as we discuss in the next section this was not the case. The 62% accuracy was still very close and overall seemed to do very well with the given data. Potential reasons for this shortcoming is discussed in section 7.1.

3.3.3. Comparison with other Model Variations

Below is a short discussion comparing the outputs of the other models that we explored. The first different model variation explored was not stemming the files. The purpose of this variation was to see if stemming actually yields better results.

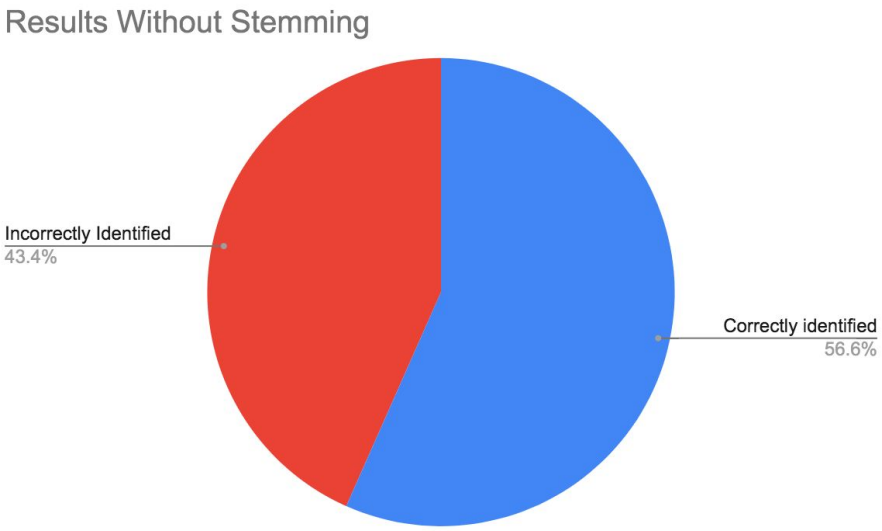


Figure 5 shows the accuracy of our model without stemming the words.

As seen in figure 5, the success rate of the model without stemming was 56.6% compared to the 62% with stemming. This shows that stemming actually helps improve the accuracy of the model. Furthermore, below shows the results of the model when calculating the sentiment score using just the difference between the positive and negative words.

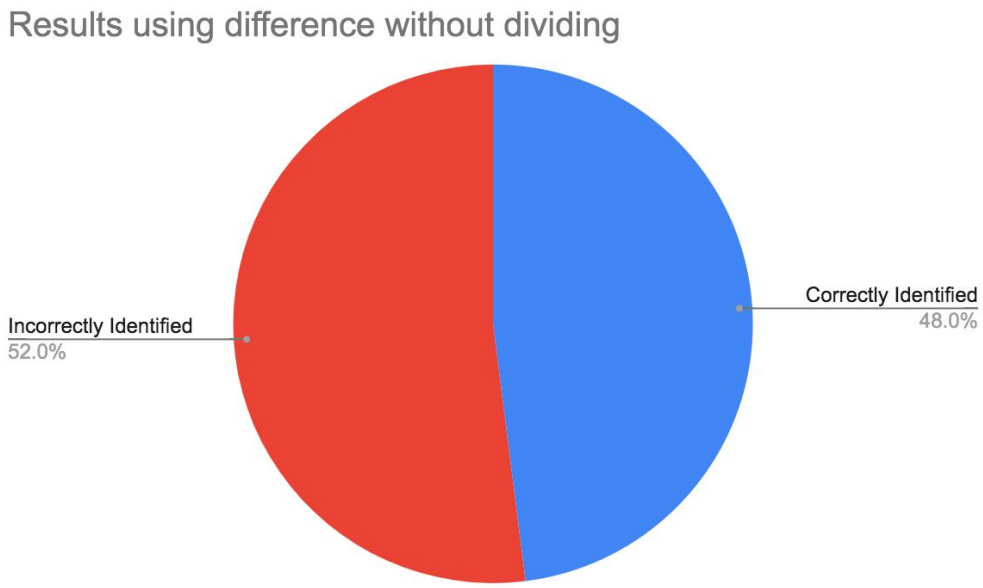


Figure 6 shows the accuracy of our model without averaging the overall sentiment.

It is evident in figure 6 that changing the way the sentiment score was calculated significantly decreased the effectiveness of our analysis. The number correctly identified fell to 48%, which is way below the final model's result of 62%. One possible reason for this is that as people write longer and longer reviews, their scores may eventually include words that may not match their inherent sentiment; furthermore, since the value returned by this score is discrete it is harder to create more refined cutoffs, which allowed for higher accuracy in the final model.

Overall, the model that was finally chosen was the best of the different variations as it had the highest rate of correctly identified sentiments.

4. Local Development: other Analyses on Homework Review Data

4.1. Popular Positive and Negative Words

The first extension of our assignment was to determine the most popular positive and negative words that students were using in their reviews. The purpose of this extension was to see if there were certain complaints or praises that students had that might hint at what makes a homework assignment enjoyable or not.

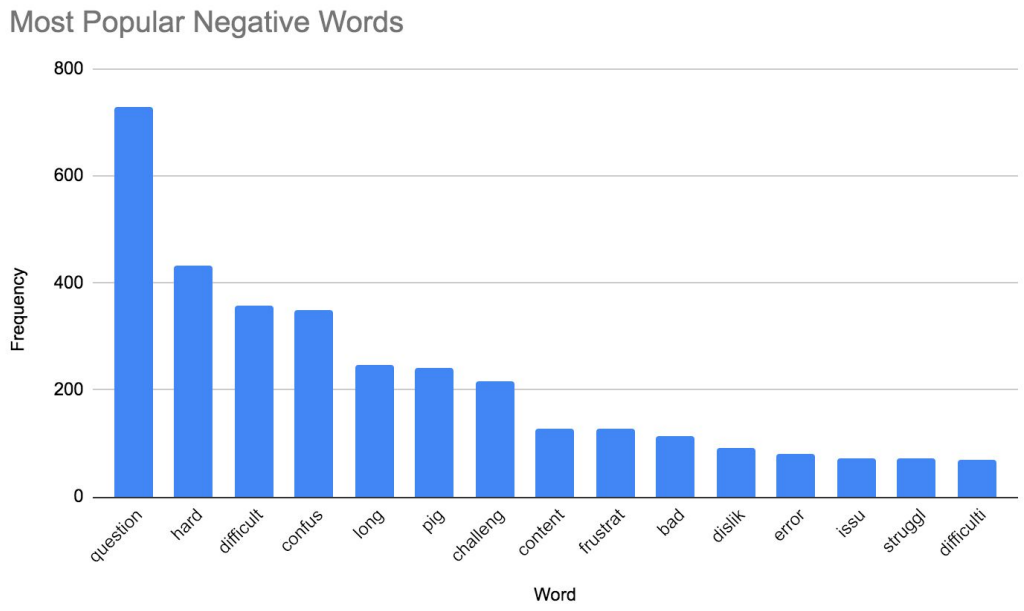


Figure 7 shows the frequency of the most occurring negative words.

As shown in Figure 7, you can see the most negative words that occurred. Although the number one word is question, we realize that this was an important stop word we forgot to take out. However, based on the next three words, “hard, difficult, and confuse(confusing)” we can see that students often feel that homeworks, which they don’t like are hard or confusing to understand. This makes sense logically as students often feel frustrated when

homeworks are hard to complete or to understand. Later, we will explore which homeworks include these words so that we can make more targeted recommendations on how to improve the course. In those homeworks, if time had permitted, we could have explored N-grams to determine which parts of the homework people felt were confusing or too long.

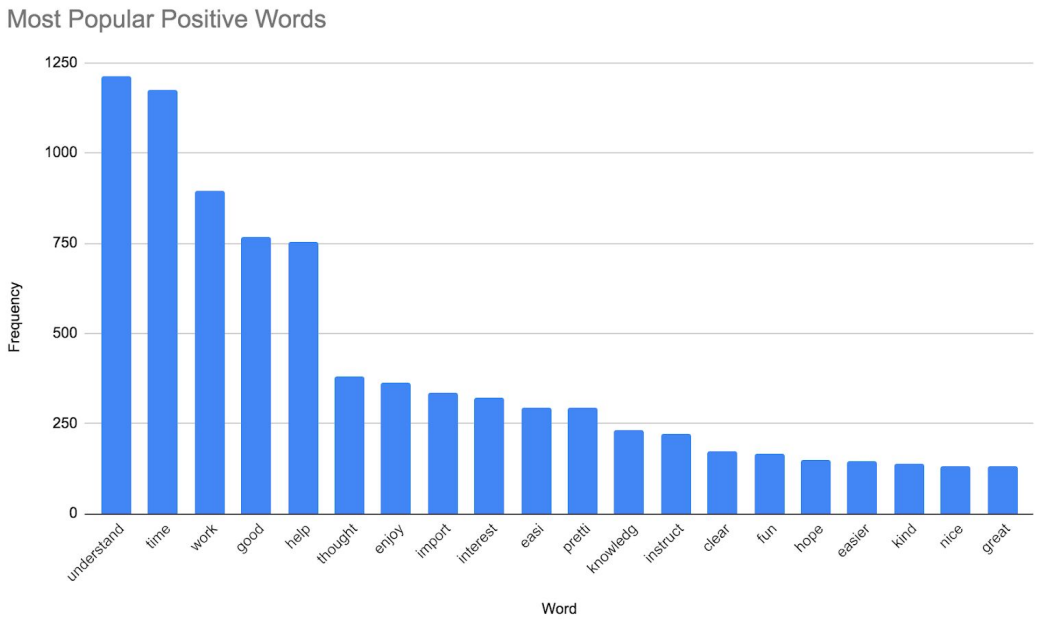


Figure 8 shows the frequency of the most occuring positive words.

Above is the graph showing the positive words that were most common in the review data set given. Some of the words that really give insight into why students enjoy homework assignments are “understand,” “help”, “clear”, “knowledge.” It is clear from this that students enjoy homeworks where they feel they really understood the material as well as had clear instructions on how to complete it. Professor Neumann should find the assignments that students enjoyed the most and use that as a model from which to build new assignments and

homeworks for students. In the next section we will explore which homeworks were the most and least popular based on the sentiment analysis.

4.2. Sentiments by Homework

In the next extension of our sentiment analysis we wanted to determine which homeworks students enjoyed the most and the least, so we could determine what topics students enjoyed the most.

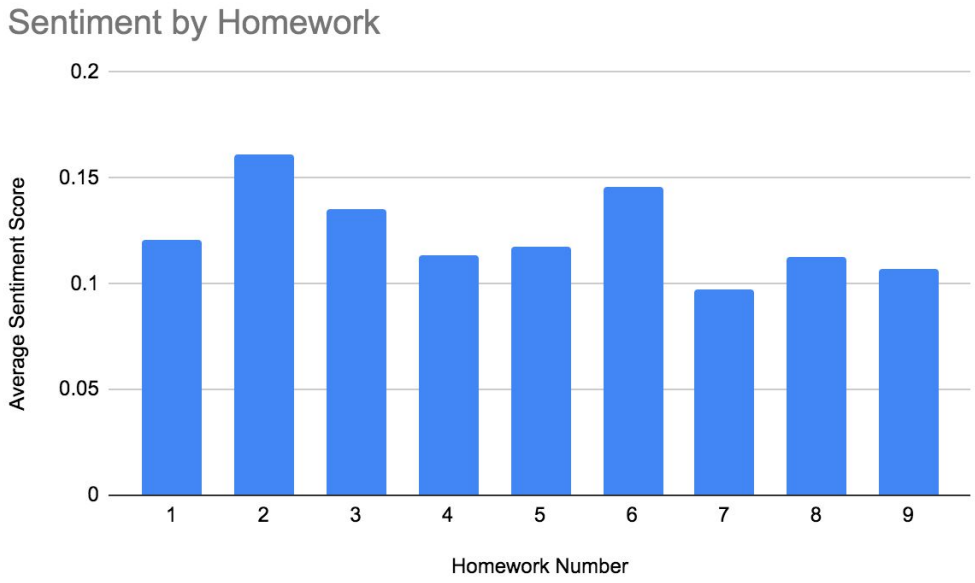


Figure 9 shows the average sentiment score by homework.

Based on the chart above, it is evident that students generally feel pretty similar about homeworks. On average, it seems, based on our bounds provided, that most students feel neutrally or a bit positive about homework assignments. It is also clear that the favorite homework assignment is homeworks 2 and 6, which the least favorites are 7 and 9. From this data, we can now explore which topics were most popular. Overall, it seems that theory

(homeworks 1-3), were more popular on average than implementation (homeworks 4-9). Additionally, it is evident that on average students seem to enjoy MapReduce (homeworks 4-7) more than Spark (homeworks 8-9). Homework 7, however, focuses a lot on Top-N-Lists as well as Word co-occurrence, which were both difficult implementation topics; therefore, we recommend splitting these topics up to different homework assignments or shortening the assignment, so that students can more effectively complete it. Also given students love for theory, Professor Neumann should consider expanding the theory section of homeworks as this seems to pique students' interests. Below shows the breakdown of sentiment scores by positive, negative and neutral labels for each homework.

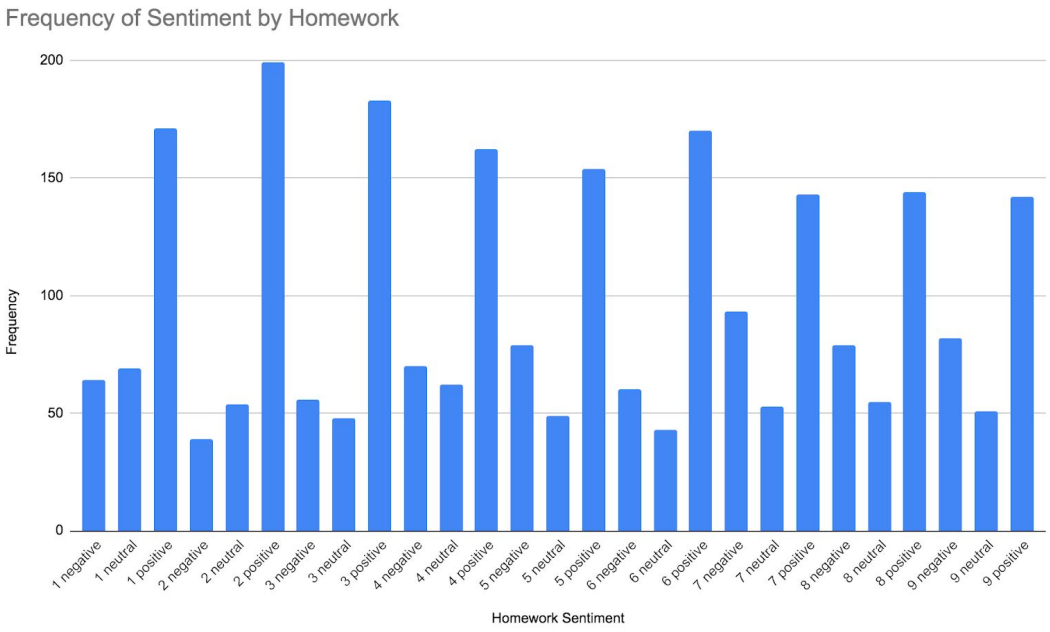


Figure 10 shows the breakdown of sentiment scores in terms of number of positive and negative reviews by homework number.

4.3. Popular Words by Homework

Finally, the last analysis we wanted to run was to determine for each homework, what were the most popular words that were used. The reason for this was to see what did students enjoy or find frustrating about the homework. In the future, to improve this analysis we would want to get rid of words that are neither positive nor negative. Below are the results for homework 2 and 7, which were the favorite and least favorite homework.

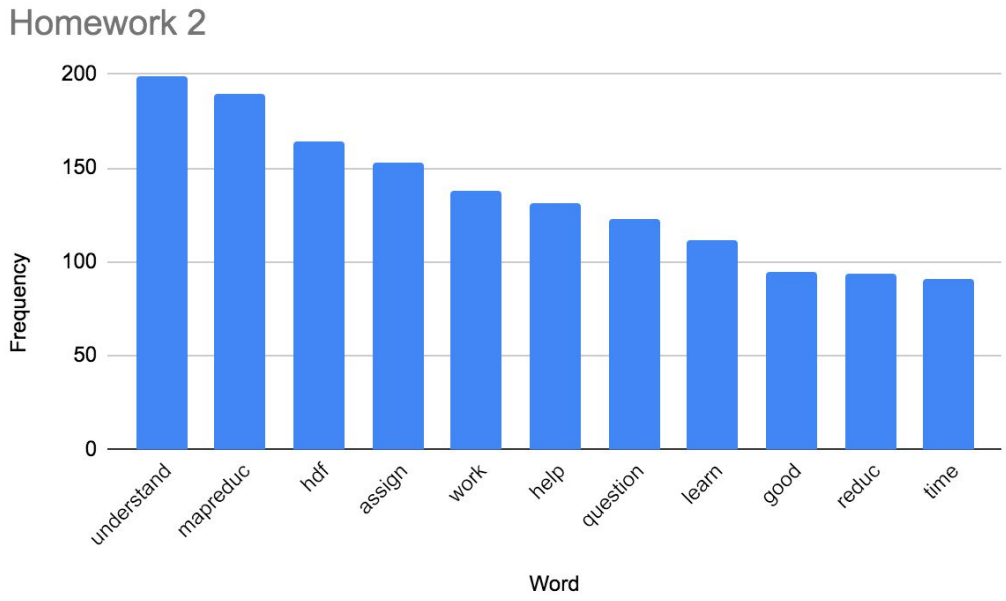


Figure 11 shows the breakdown of the most popular words from homework 2.

Homework 2, which covers the theory of MapReduce, was the favorite homework assignment amongst students. Although some words that are the theory (mapreduce, hdfs, assign,etc) were in the most popular words, there are also words like “understand,” “learn,” and “good,” which matches what we expected to see in the most popular words, as students generally enjoyed this assignment. From this we can safely say that homework 2 should be used as a model by which Professor Neumann bases other assignments!

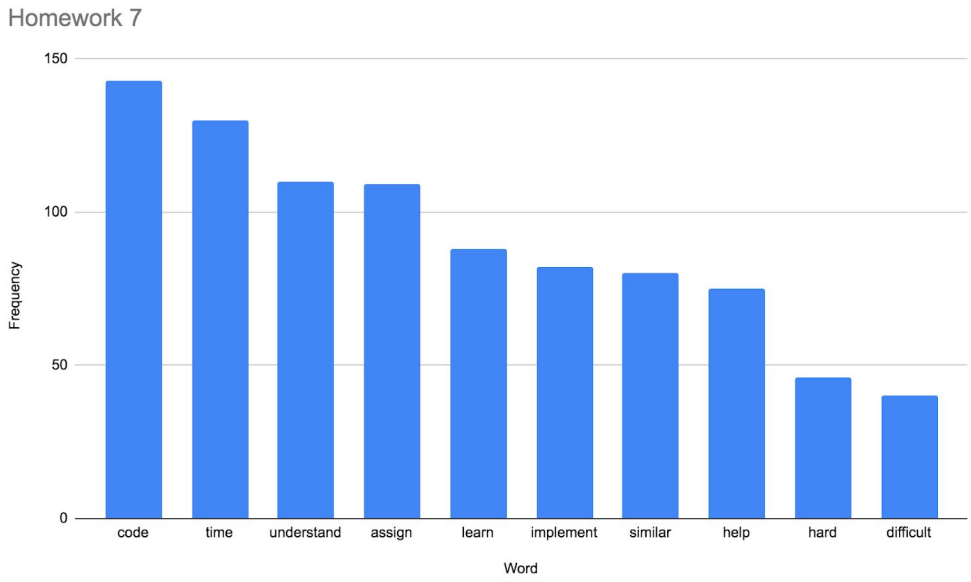


Figure 11 shows the breakdown of the most popular words from homework 7.

Alternatively, homework 7 was one that students did not enjoy very much. This can be shown by having words like “hard” and “difficult.” Although most words are positive, we do not know if there was a lot of negation in these reviews (i.e. “This homework was not understandable.”), so we cannot say for sure that all of these frequencies correctly correlates to the sentiment. However, since hard and difficult came up so much more frequently than in other homeworks, it is clear that Professor Neumann should aim to reduce the difficulty of this assignment by either breaking it up into chunks, or by providing additional office hours during the week that she assigns this homework.

5. Cloud Execution and Extension

5.1. Approach for Cloud Execution

Now that we had a working example on a smaller dataset, we wanted to extend our sentiment analysis to a big data set:Amazon Product Reviews. In order to successfully run this sentiment Analysis we followed the following approach:

5.2. Big Data Dataset

The dataset we selected was the Electronics Reviews which had 1,689,188 reviews. The raw data was originally in a JSON file, so the reviews and star ratings had to be extracted. This file set is considered big due to the sheer number of reviews. Below, we outlined how we extracted and then analyzed the data.

5.3. Preprocessing and Analyzing Amazon Reviews

In order to process this large dataset on the cloud, we used Amazon EMR by modifying our earlier Spark program. Once reviews were extracted from the JSON file they were placed into a text file in the following format “rating \t review text.” Since it was in this format, we were able to apply a very similar process to before; however, with a few adjustments. Due to some complications due to time to running the job, we did not stem for this portion of the analysis. We still removed stop words and non-alphanumeric characters, so that we could enhance our ability to analyze the data. After the preprocessing was complete, the data was placed into another file in the form of “rating \t pre-processed text review.” After this, we used our same sentiment score calculation of:

$$sentiment\ score = \frac{positive\ words - negative\ words}{total\ words\ in\ review}$$

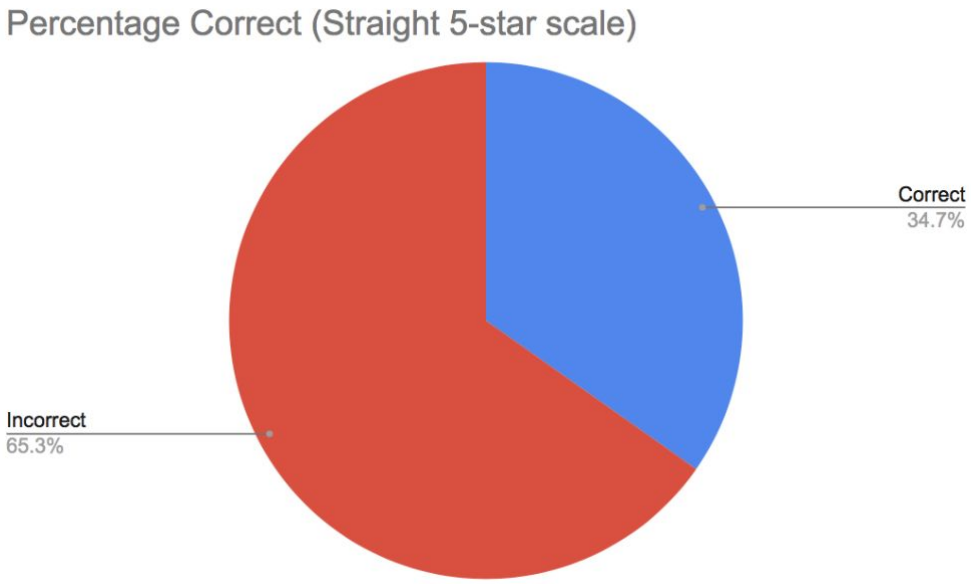
Then we chose bounds to identify what we would consider a "5 star" vs other star ratings. We loosely based this on where the bounds from earlier were placed and we decided on the following:

Star Rating	Bound for Sentiment Score
1	[-1,0]
2	(0,0.025)
3	(0,0.04)
4	(0,0.08)
5	[0.08,1]

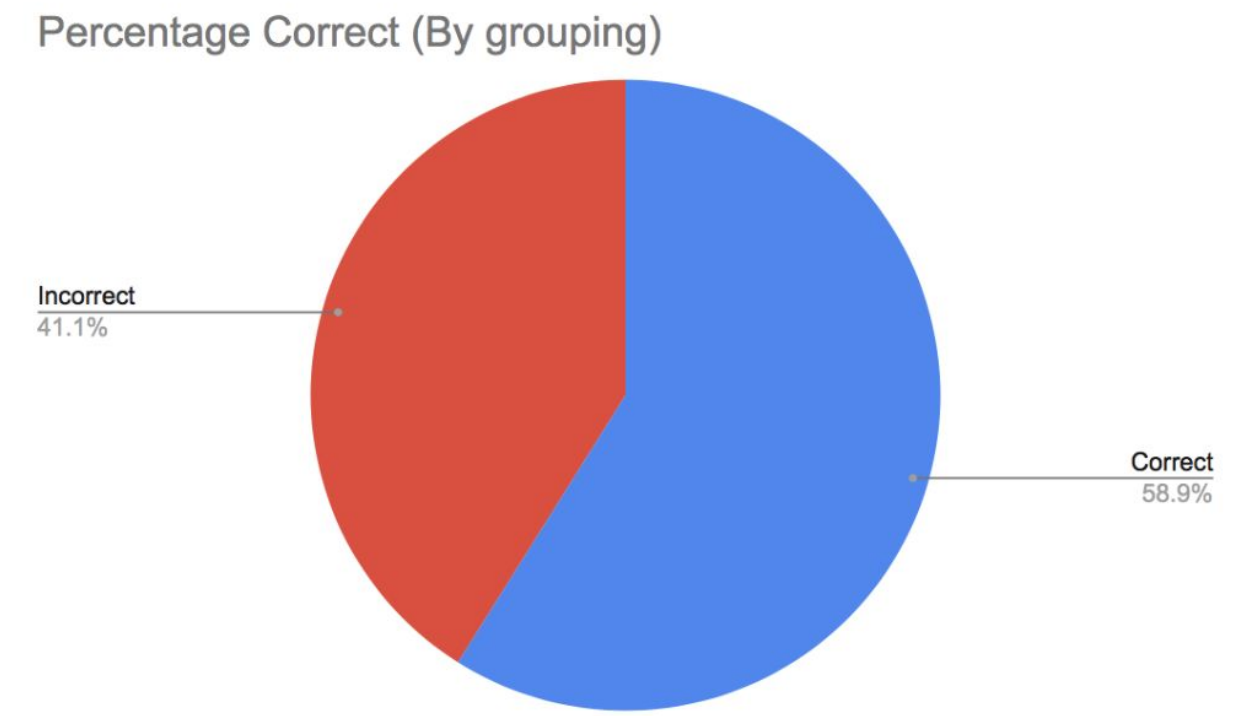
We chose to make 5 the largest category as most Amazon reviews tended to be 5-star reviews. After calculating the sentiment scores of each individual review, we were able to see how well our sentiment Analysis was able to be extended. This process took a long time and the runtime was 1 hour and 52 minutes. We should look for ways to try to bring down this runtime, but given the huge dataset it makes sense that it took this long to process.

5.4. Amazon Reviews Results

In order to determine how effective our sentiment analysis was on Amazon reviews, we calculated the percentage of how many of the reviews we correctly identified by their star rating:



As you can see from above, the percentage of correctly identified reviews was 34.7%, which is much lower than from the homework reviews. The main reason for this is that there are more categories to choose from so it is more difficult to determine the more minute difference between a 4 and a 5 then between negative and positive. To show this, we grouped ratings 1 and 2 together, 3 as its own category and 4 and 5 as a category together (to roughly match negative, neutral and positive). Below shows the results of our analysis:



As is evident, when we reduced the number of options from a 5 star scale down to our 3 separate groups, our analysis performed much better and correctly identified the grouping with 58.9% accuracy. Although this is lower than homework reviews, we were still satisfied with this result. We obviously would love to improve this percentage, and we discuss potential ways to improve in section 6.1. In section 5.4, we will discuss potential extensions we could have explored that Amazon could use to get valuable insights on their products.

5.5. Possible Extensions for Amazon

In order to provide valuable insights to Amazon using this sentiment analysis, one such extension we could explore is we could see which products had the highest average

sentiment score versus lowest average sentiment score. In doing so, Amazon would be able to tell which products they may be able to charge more for or if there is a busy season coming up they could increase the amount they have in stock. Alternatively, if there is a product with an extremely low rating they could look to cut it from their collection.

Another extension that would be possible would be similar to above, but using N-grams to determine what parts of the product people enjoyed or found frustrating. In doing so, Amazon could sell this information to their suppliers, so that the product could be improved to match customers concerns or frustrations.

Finally, they could look at how the average review of a product changed over time to see if adjustments or updates to a product or service was perceived positively or negatively. In doing so, Amazon can see if they should continue those updates or change it back based on customer feedback.

6. Conclusion

6.1. Improvement for the future

After completing our model, we think it is important to recognize potential improvements and next steps we can make. Firstly, we explored the idea of assigning a range of scores to words instead of -1, 0, and 1. By assigning very positive words a higher number and very negative words with a lower number, we further polarize these words, and would potentially improve the accuracy of our model. Some examples of words that may receive a more positive score include “spectacular” and “amazing”, whereas words that would receive a more negative score would be words like “dreadful” and “horrible.” Neutral-positive words and neutral-negative words like “good” and “bad” would be assigned a score closer to zero. Although this approach would hopefully yield a more accurate result, there are a couple

limitations to it. It is much more complex to assign each positive and negative word to a category within the positive and negative range. Additionally, we would have to be more careful about our negation handling because if we miss a word deemed to be "very positive" or "very negative," it could more greatly affect the sentiment of the review. An example of this is the sentence sentence, "I did not think this homework was spectacular."

Furthermore, in order to improve the accuracy of our model, we have come up with a few potential next steps. Firstly, we realize we could handle more complex cases of negation handling in our model. Our current model only handles negation words that are directly next to each other (e.g. "not good"). Although this approach will handle many cases of negation, not all negations come in this form. For example, the sentence "I didn't do very well" has an overall negative sentiment, however our model would have given this a positive score because the negation does not directly precede the word. By coming up with a more complex algorithm for negation handling, we could improve the overall accuracy of our model.

Additionally, there are ways to improve our extension of getting word frequency by homework. For example, it would be much better if we took N-Grams of each topic i.e. "MapReduce" or "Hive" to see what people are saying specifically about that topic. Additionally, we could also improve this analysis by removing neutral words and instead showing what were the most popular words from our positive and negative lists. With these small tweaks, we would have been able to make much more targeted and effective recommendations on how to improve homeworks and the course as a whole.

6.2. Lessons Learned

Through this sentiment analysis process, we learned several valuable lessons. Firstly, we wrote our sentiment predictor in a way that was very easy to change our parameter cutoffs for positive, negative,

and neutral words. The ease of changing these words allowed us to easily test different values to come up with the ones that yielded the most accurate results. We also learned the value of writing modularized code and testing it incrementally. Before preprocessing all of our data and running it through our sentiment predictor, we tested our code and algorithms on a small subset of our data. This allowed us to test and tweak our code efficiently without having to re-process our entire dataset every time we made a change.

Beyond the sentiment analysis process, we learned many ways by which Professor Neumann can improve her course. For example, by changing homework 7 or spending more time on theory, we hypothesize based on our results that students will be more excited to come to class and find homeworks more enjoyable to complete!

7. Appendix

7.1. Text Pre-processing

Original:

```
Hw 9, positive, I had a break-through on this problem. For a
while I was running into storage issues and they were very
frustrating, but once I figured out how to run a pyspark program
through a python text file things went a lot more smoothly. I am
a little worried about my written answers. I should double check
with the textbook.
```

Pre-processed:

```
9    positive    break run storag issu frustrat figur run pyspark
program python text file thing lot smoothli worri written answer
doubl check textbook
```

7.2. Sentiment Calculation

Example sentence: Finals are hard, but I am confident that I will do well.

This sentence would return a value of 2 (positive words: confident and well) -1 (negative word: hard)/12 = 1/12 = 0.08