Sentiment Analysis and Prediction Using Amazon Review Data

Al Bashir Muhammad, Kelsey McBratney,

Blake Brown, Ojaswi Sinha

Colorado State University

CS 435 Big Data

Professor Sangmi Pallickara

12/2/2023

Table of Content

1. Introduction

In this project, we aim to develop a system that can predict a user's rating based on their review, using the Amazon Review Data (2018) dataset [2]. The primary objective is to accurately determine the sentiment of the review, whether it is positive, negative, or neutral, given a user's comment on a product. This tool will provide businesses with a powerful tool to monitor and enhance customer satisfaction, tailor marketing strategies, and even predict market trends based on customer feedback patterns.

According to Amazon, it calculates a product's star rating using machine-learned models instead of a simple average [3]. These models consider factors such as how recent the rating or review is and verified purchase status. They use multiple criteria that establish the authenticity of the feedback. The system continues to learn and improve over time. To maintain trustworthy review scores, only customers who have spent at least $50 on Amazon in the last 12 months can submit ratings and reviews. Before posting a review, they also check if it meets some set guidelines. That includes rules against creating, editing, and removing reviews in exchange for compensation. They also check if the reviewer bought or used (e.g., streamed) the item on Amazon and paid a price available to most Amazon shoppers. When these two conditions are satisfied, the review is labeled *Verified Purchase*. Reviews without this label can also be helpful. For example, a customer buys an item from a different company but wants to share their opinion on Amazon.

But why would any company go to this length to engineer a highly functional review system? The significance of this capability lies in the fact that customer reviews have a significant impact on purchasing decisions and perceptions of product quality in the e-commerce industry. Our project aims to distill the essence of these reviews into a quantifiable sentiment rating, providing a clear and immediate understanding of customer opinions. This automated interpretation of sentiment is essential for product feedback and improvement, as it enables manufacturers and sellers to quickly gauge customer reactions and adjust their strategies accordingly. In addition, the vast amount of reviews generated on platforms like Amazon makes manual analysis a challenging and sometimes impossible task. Our approach addresses this challenge by automating the sentiment analysis process. This allows us to process large volumes of text data efficiently, extract valuable insights, and save time while increasing the scalability of review analysis.

Why is it interesting as a Big Data problem and who would use it if it were solved?
Sentiment analysis and prediction on Amazon datasets are interesting Big Data problems for several reasons:

- Scale of Data: Amazon accumulates a massive volume of user reviews and product-related data due to its vast product catalog and user base. Analyzing sentiments across such a vast dataset requires scalable and efficient big data solutions to process them.

- Diversity of Products: Amazon sells a wide variety of products, ranging from books and electronics to clothing and groceries. Sentiment analysis needs to be adaptable to diverse product categories, making the problem more complex and requiring sophisticated models capable of understanding context across different domains.
- Textual Complexity: User reviews can be highly nuanced and contain complex language, slang, and sentiment expressions. Understanding the sentiment behind such diverse and subjective language is a challenging natural language processing (NLP) task.
- Contextual Understanding: Sentiment analysis requires understanding the context in which words and phrases are used. The same words can have different meanings based on context, making it essential to capture the context in which sentiments are expressed.
- Imbalanced Data: User reviews are often imbalanced, with a larger number of positive reviews compared to negative ones. This imbalance can affect the performance of machine learning models, requiring techniques to handle skewed datasets.

If the problem of sentiment analysis on Amazon product datasets were solved, a range of stakeholders could use it:

- Amazon would be able to provide enhanced shopping experiences with more accurate product recommendations, and the ability to make informed purchasing decisions based on the sentiments of customers. Improved customer satisfaction, better product recommendations, and informed decision-making for marketing and inventory management.
- Developing advanced machine learning models for sentiment analysis. Insights into customer feedback can help brands and sellers understand how their products are perceived, identify areas for improvement, and respond to customer concerns. By understanding the sentiment behind user reviews, Amazon can recommend products that align with individual preferences and increase the likelihood of user satisfaction. This Model can also be used for other sentiment analysis tasks with varies datasets
- Analyzing sentiments can reveal trends and shifts in consumer preferences and market dynamics. This information is valuable for decision-makers at Amazon to adjust strategies, optimize inventory, and respond to changing market demands.

Unique Characteristics of Our Dataset

Our dataset, *Home and Kitchen*, is a JSON format data of around 3.48GB. The attributes within each JSON object include overall, review time, asin, style, color, reviewer name, review text, Summary, and unix review time. Additionally, some reviews include a vote attribute, indicating the number of helpful votes received by the review.

In summary, our project aims to contribute significantly to the understanding and utilization of big data in customer sentiment analysis. By harnessing the power of machine learning and natural language processing, we aim to transform raw data into meaningful insights, ultimately benefiting both businesses and consumers in the digital marketplace.

2. Methodology

The Algorithms

We chose TF IDF and RandomForest Classifier because they are well suited to this type of problem and fairly easy to use.

Term Frequency-Inverse Document Frequency (TF IDF)

Term Frequency (TF): It measures the frequency of a term (word) within a document. The idea is that the more often a word appears in a document, the more important it might be. This means the raw frequency is divided by the maximum raw frequency of any term k in the article j.

$TF_{ij} = 0.5+0.5* (f_{ij} /maxk\ f_{kj})$

Where f_{ij} is the frequency (Number of occurrences) of the term (word) i in the Wikipedia article j.

Inverse Document Frequency (IDF)

IDF is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term. Terms common across all documents in the corpus receive lower scores.

$IDF_i = \log10(N / n_i)$

where N is the total number of articles (corpus).

TF IDF Score

The TF IDF score for a term in a document is the product of its TF and IDF scores. It reflects both the local importance of the term within the document and its global importance across the corpus. In other words, the TF IDF score is high when a term occurs frequently in a document (TF) but less frequently in other documents in the corpus (IDF). Therefore, terms with the highest TF IDF score are considered words that best characterize the document.

$TF\text{-}IDF= TF_{ij}$ x IDF_i

Random Forest Classifier

Random Forest Classifier is a powerful machine learning algorithm that can be employed to predict the sentiment of reviews based on various features extracted from the data. Random

Forest belongs to the ensemble learning category, meaning it builds a model by combining the predictions of multiple individual models. It involves the following steps:

- **Text Vectorization:** We converted the textual content of Amazon reviews into numerical values using TF-IDF (Term Frequency-Inverse Document Frequency)
- **Data Splitting:** We split the dataset into training and testing sets. The training set is used to train the Random Forest model, while the testing set is used to evaluate its performance.
- **Training the Random Forest Model:** We initialized a Random Forest Classifier and fed it the training data. The classifier builds an ensemble of decision trees, each trained on a random subset of the data and features.
- **Model Evaluation:** We evaluated the performance of the trained model on the testing set using metrics like accuracy, precision, recall, and F1-score.
- Adjust hyperparameters if needed to improve model performance.
- **Prediction:** Use the trained Random Forest model to predict sentiment labels for new, unseen Amazon reviews. The model assigns a sentiment label (positive, negative, neutral) based on the learned patterns from the training data.

Our Approach

Our approach to solving this problem is comprehensive and systematic. It is designed to process and analyze vast amounts of text data efficiently. Our strategy involves several key steps which are elaborated on below:

Step 1: Feature Extraction Using TF-IDF

The first step involves extracting meaningful features from the review text. To do this, we use the Term Frequency-Inverse Document Frequency (TF-IDF) technique. TF-IDF is a statistical measure that assesses how relevant a word is to a document in a collection of documents. It involves multiplying two metrics: how many times a word appears in a document (term frequency), and the inverse document frequency of the word across a set of documents. This results in a set of features that highlight the most significant words in each review, essential for understanding the sentiment expressed in the text. This method is particularly effective in filtering out common words that are less informative about the overall sentiment.

We also focused on preprocessing and cleaning the review text data before extracting features. This includes removing irrelevant characters, correcting misspellings, standardizing the text format, handling missing values and removing duplicates. We also go through the data to remove any common stop words, these are words that commonly occur that don’t add any usefulness to this text. Such as A, I, Our, He, etc.

Step 2: Labeling Based on Star Ratings

After extracting the features, we assign sentiment labels to each review. We accomplish this by analyzing the star ratings associated with each review. Generally, high ratings (4 or 5 stars) indicate positive sentiment, low ratings (1 or 2 stars) suggest negative sentiment, and moderate ratings (3 stars) are considered neutral. This labeling process is critical because it provides the ground truth for our machine-learning model. It enables the model to learn and predict sentiment accurately.

Step 3: Model Training with Random Forest Classifier

With our features extracted and labels assigned, we proceed to train our machine learning model. We have chosen the Random Forest classifier for this purpose due to its robustness and effectiveness in handling large and complex datasets. Random Forest works by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes of the individual trees. This approach is particularly beneficial for our project, as it is less prone to overfitting and can handle the nuances and variabilities in text data efficiently.

We perform hyperparameter tuning of the Random Forest classifier to find the optimal configuration for our specific dataset. We did this process manually, and found different values of Max Depth, and Max Trees that gave us the best results.

Step 4: Predictions for New Products

The final step involves applying the trained model to make sentiment predictions for new product reviews. Our model can now analyze reviews of products that were not part of the original training dataset. By inputting the text of these new reviews, our model can quickly categorize them into positive, negative, or neutral sentiments, providing immediate insights into customer opinions on new products.

Framework

The framework we choose for our project are the following.

- **Hadoop, MapReduce - Java**
- **Apache Spark - Java**
- **Spark MLlib - Java**

Hadoop:

- The key components of the Hadoop FrameWork are:
  - **HDFS**: Hadoop Distributed File System (HDFS) stores the data across multiple machines by dividing the large files into smaller blocks normally in the range of 128MB to 256 MB and replicates it across all the nodes in the cluster to create fault tolerance. Parallel processing is possible along with efficient data extraction.
  - **MapReduce:** It is used for computing large datasets. There are 2 phases: Map and Reduce Phase.
  - **YARN:** Yet Another Resource Negotiator (YARN) is a resource management layer allowing multiple applications to share resources in the Hadoop cluster.The flexibility of yarn makes it easier to run workloads like: MapReduce and Apache Spark

MapReduce:

- **Input:** the data is divided into splits and each split is processed by a different Mapper task.
- **Map Function:** Its a user defined Map function which is applied to each record in the input data split independently. This will emits a key-value pair immediately
- **Shuffle & Sort:** The framework groups and sorts the key-value pairs based on the key
- **Reducer:** It is also a function which is defined by the user which is then applied to each group of intermediate key-value pairs under the same key/ the output of this function is the final set of key-value pairs

Apache Spark:

- Apache Spark is a distributed system that provides a fast and general purpose cluster for big data processing. This allows us to process large amounts of data in a relatively quick way.

Advantage of Random Forest (Spark MLlib):

- Very easy to measure the relative importance of each feature in the prediction. It's a feature importance matrix

- There is less featuring engineering and there are fewer statistical assumptions in Data distribution and relationships between the features
- It's used for both regression and classification

Disadvantage of Random Forest:

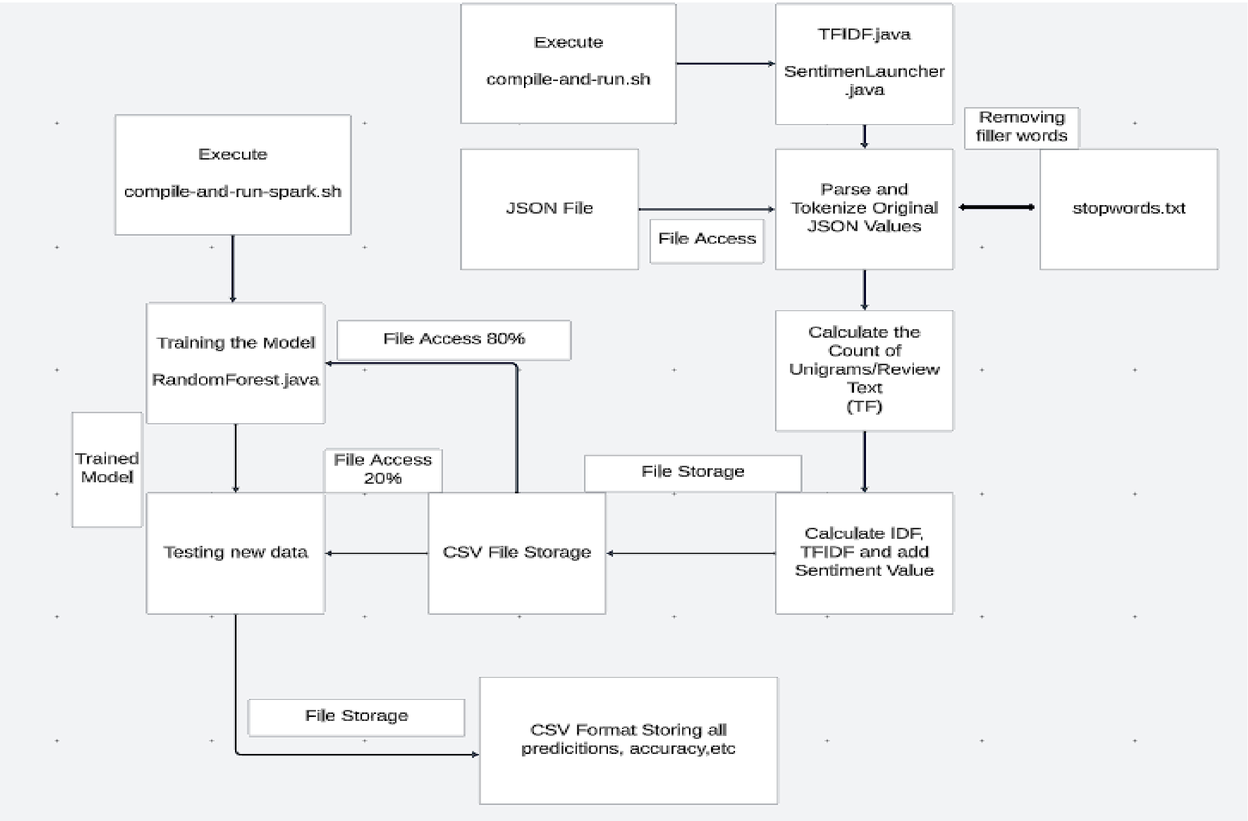
- It is highly complex compared to decision trees
- Due to its high Complexity, its training time is also significantly increased

Workflow

Below is the overall workflow of our code. Starting with Map Reduce, we parse our Amazon data and collect the relevant information, while also removing and stopping words that are in our list. Once all 3 Map Reduce Jobs are complete the output is saved to our HDFS cluster as a tab delimited file, containing the reviewText, Rating, Unigram, TF IDF, Sentiment.

The next step is to run this data through Apache Spark and SparkML. We take this input data and split it into a 80/20 train, and test model. Once the model is trained it is compared to the test data and we output the results to our HDFS cluster. The output file contains reviewerID, indexedLabel, prediction, precision, recall, and f1.

<u>Visual representation of the workflow.</u>



3. Dataset

Our project is based on the Amazon Review Data (2018). We focus specifically on the 'Home and Kitchen' category, as it is well-suited for our analysis, being comprehensive and relevant to everyday consumer products. Below are the specifics:

Size and Manageability: The chosen subset *Home and Kitchen* data is around 3.48GB, which is both significant for meaningful analysis and manageable for our processing capabilities. The full Amazon Review Data is over 100GB, presenting significant challenges in terms of computational resources and processing time. By focusing on this subset, we achieve a balance between data richness and practical feasibility.

Data Format: The dataset is structured in JSON format, which is versatile and widely used in data processing.

Data Attributes: Each entry in the dataset includes various attributes that are crucial for our analysis. These attributes include:

- reviewerID: a unique identifier for the reviewer.
- asin: Amazon Standard Identification Number, a unique code for identifying the product.
- reviewText: the actual text of the review.
- overall: the star rating given by the reviewer, ranging from 1 to 5.
- Additional attributes, such as votes for review helpfulness, product style or format, summary of the review, time of review, and images posted with the review.

Data Characteristics: The reviews in the dataset span from May 1996 to October 2018, providing a wide range of customer reviews over time. The reviews include different lengths of text, writing styles, and sentiments, making it a rich source for sentiment analysis. The inclusion of product metadata, such as category information, price, and brand, adds depth to our analysis, allowing us to examine sentiment about these factors.

Here are some examples of our dataset, illustrating the structure and type of data we are working with:

```
{
 "reviewerID": "A2SUAM1J3GNN3B",
 "asin": "0000013714",
 "reviewerName": "J. McDonald",
 "vote": "5",
 "style": {"Format:": "Hardcover"},
 "reviewText": "I bought this for my husband who plays the piano...",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```

Each of these entries provides a wealth of information that we can analyze to derive insights into customer sentiment and product reception.

4. Discussion and Analysis:

Here are some results from some of the sample runs which trained the model, we experimented with different Max Depths, and Max Trees. We found for the best results we had a Max Depth of 12, and a Max Tree size of 100. This landed us an accuracy of **73.8%.** Below is the output of a few of our best runs.

Results:
Dataset: Sample - 14,429 entries.
Max Depth, Max Trees

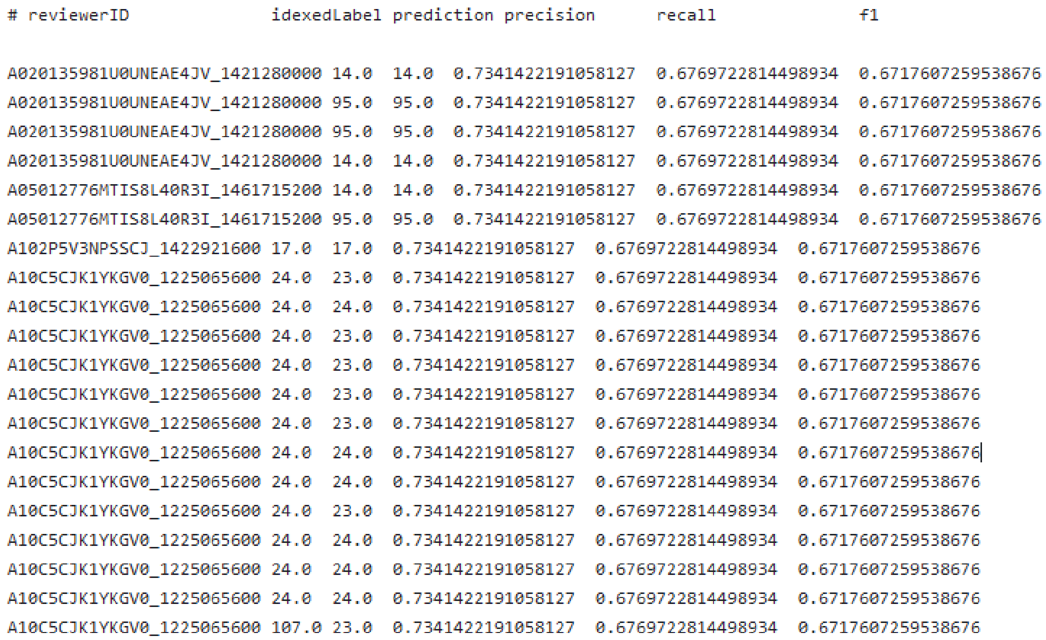
Max Depth 10, Max Trees 100
Precision = 0.7341422191058127
Recall = 0.6769722814498934
F1 Score = 0.6717607259538676
Training Error Percentage = 27.645286267757207%
Testing Error Percentage = 32.30277185501066%
Test Accuracy = 0.6769722814498934

Max Depth 8 , Max Trees 200
Precision = 0.6120851348604399
Recall = 0.6019900497512438
F1 Score = 0.5761587332121166
Training Error Percentage = 35.21308652604391%
Testing Error Percentage = 39.80099502487562%
Test Accuracy = 0.6019900497512438

Max Depth 12 , Max Trees 100
Precision = 0.7747338977155214

Recall = 0.7377398720682303
F1 Score = 0.7303161435158251
Training Error Percentage = 21.179509255273356%
Testing Error Percentage = 26.226012793176967%
Test Accuracy = 0.7377398720682303

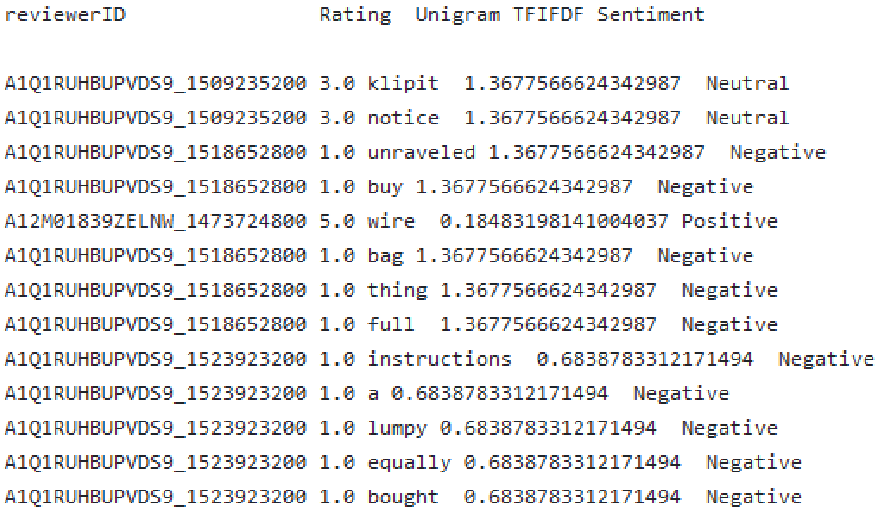
Output Results from Spark ML



```
# reviewerID               idexedLabel prediction precision      recall             f1

A020135981U0UNEAE4JV_1421280000 14.0  14.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A020135981U0UNEAE4JV_1421280000 95.0  95.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A020135981U0UNEAE4JV_1421280000 95.0  95.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A020135981U0UNEAE4JV_1421280000 14.0  14.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A05012776MTIS8L40R3I_1461715200 14.0  14.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A05012776MTIS8L40R3I_1461715200 95.0  95.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A102P5V3NPSSCJ_1422921600 17.0  17.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  24.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  24.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  24.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  24.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  24.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 24.0  24.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
A10C5CJK1YKGV0_1225065600 107.0 23.0  0.7341422191058127  0.6769722814498934  0.6717607259538676
```

Output Data of Hadoop TFIDF



```
reviewerID                Rating  Unigram TFIFDF Sentiment

A1Q1RUHBUPVDS9_1509235200 3.0 klipit  1.3677566624342987  Neutral
A1Q1RUHBUPVDS9_1509235200 3.0 notice  1.3677566624342987  Neutral
A1Q1RUHBUPVDS9_1518652800 1.0 unraveled 1.3677566624342987  Negative
A1Q1RUHBUPVDS9_1518652800 1.0 buy 1.3677566624342987  Negative
A12M01839ZELNW_1473724800 5.0 wire  0.18483198141004037 Positive
A1Q1RUHBUPVDS9_1518652800 1.0 bag 1.3677566624342987  Negative
A1Q1RUHBUPVDS9_1518652800 1.0 thing 1.3677566624342987  Negative
A1Q1RUHBUPVDS9_1518652800 1.0 full  1.3677566624342987  Negative
A1Q1RUHBUPVDS9_1523923200 1.0 instructions  0.683878331217494  Negative
A1Q1RUHBUPVDS9_1523923200 1.0 a 0.683878331217494  Negative
A1Q1RUHBUPVDS9_1523923200 1.0 lumpy 0.683878331217494  Negative
A1Q1RUHBUPVDS9_1523923200 1.0 equally 0.683878331217494  Negative
A1Q1RUHBUPVDS9_1523923200 1.0 bought  0.683878331217494  Negative
```

We evaluated our Random Forest Classifier by calculating accuracy. This is based on the assumption that our dataset and classes are balanced.

Accuracy is calculated by dividing the number of correct predictions made by the model by the total number of predictions. This metric gives us an overall idea of the model's correctness but does not provide details of the specific types of errors (such as false positives or false negatives).

Accuracy = (Number of correct decisions made) / (Total number of decisions made)
F-measure = 2(precision x recall)/(precision + recall)
Precision = TP / (TP+FP)
Recall = TP / (TP+FN)
Accuracy = (TP + TN) / (P + N)

Where TP: True positive
TN: True negative
FP: False positive
FN: False negative

Challenges

1. Due to time constraints, we could not include a confusion matrix as part of our evaluation.
2. We ran into a memory problem which caused us to use a subset of the chosen file rather than the entire dataset. Instead of the entire 96 million entries from the whole data set after ranking TF IDF values. We cut this dataset down to 100,000 entries to train our dataset.
3. Due to the large size of the test files, processing took several hours. Most of the time it ultimately crashes with JavaOutOfHeapMemory after several hours. This is one of the big downsides of using RandomForest, it consumes a lot of memory for datasets that contain a large amount of unique key values.

5. Project Contributions

a. Setting up a rough framework for Hadoop (Kelsey, Al Bashir)
b. Implementing MapReduce for TF-IDF calculations (Kelsey, Al Bashir)
c. Labeling sentiment and starting model training (Ojaswi, Blake)
d. Resolving issues and initial predictions (Blake, Ojaswi)
e. Finalizing the model and preparing for the presentation (Kelsey)

6. Bibliography

1. Kieran O. and Amar C., “Why Ratings on Everything from Wine to Amazon Products Improve Over Time”, *Harvard Business Review*, https://hbr.org/2018/10/research-why-ratings-on-everything-from-wine-to-amazon-products-improve-over-time, October 03, 2018
2. Project, UCSD CSE Research. “Amazon Review Data (2018).” *Amazon Review Data*, https://nijianmo.github.io/amazon/index.html#subsets. Accessed Oct. 25, 2023.
3. Understanding customer reviews and ratings, https://www.amazon.com/gp/help/customer/display.html?nodeId=G8UYX7LALQC8V9KA. Accessed on Dec. 2, 2023.
4. Aamir R. and Ching-yu H., "Sentiment Analysis on Consumer Reviews of Amazon Products", *International Journal of Computer Theory and Engineering*, Vol. 13, No. 2, May 2021.
5. T. U. Haque, N. N. Saber and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews," *2018 IEEE International Conference on Innovative Research and Development* (ICIRD), Bangkok, Thailand, 2018, pp. 1-6, doi: 10.1109/ICIRD.2018.8376299.