

**Big Data Technologies and Applications**

**Final Project Report**

**Amazon Product Reviews and**

**Helpfulness Score**

Group We’R, Ayşe Çetinel Yağşi, Engin Erdemli, Erçin Eldeleklioğlu, Yonus Kula

*Istanbul Technical University, MSc in Big Data and Business Analytics*

**I. ABSTRACT**

Product reviews and ratings are popular tools to support buying decisions of consumers. These tools are also valuable for online retailers, who use rating systems in order to build trust and reputation in e-commerce. Many online shops offer quantitative ratings, textual reviews or a combination of both. This project aims to provide statistical insights into Amazon product reviews dated from May 1994 to July 2014, examine their helpfulness in recommending products, suggest a new way to predict helpfulness score of the user reviews.

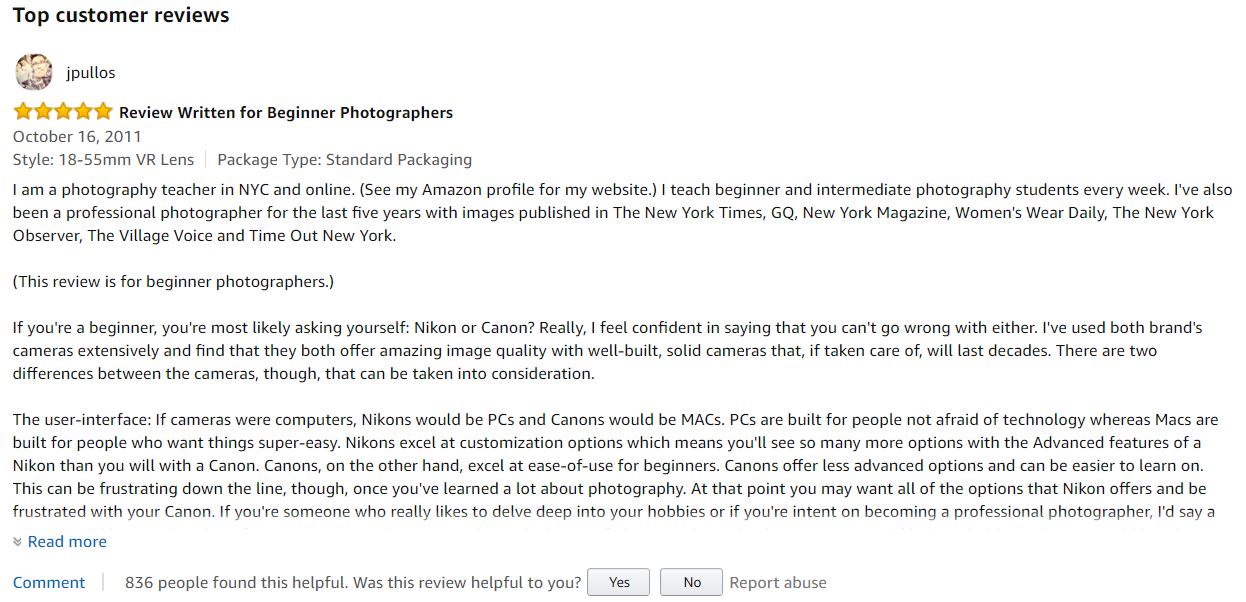
The number of reviews on Amazon has grown significantly over the years. Customers who made purchases on Amazon provide reviews by rating the product from 1 to 5 stars and sharing a text summary of their experience and opinions of the product. The ratings of a product then are averaged to provide an overall product rating. We analyzed how Amazon customers write reviews and what type of ratings they give in a specific category in order to find underlying statistical trends providing insights into inner workings of Amazon’s review system. We trained various classifier models using a training set of preprocessed reviews. The classifier predicts whether the review is helpful or not. The performance of the classifiers was tested on a test set containing 30% of the dataset.

**II. MOTIVATION**

Retail ecommerce sales worldwide posted solid gains in 2017, rising 23.2% to $2.290 trillion. In 2017, for the first time, ecommerce sales accounts for almost one-tenth of total retail sales worldwide. China and the US combine for $1.584 trillion in ecommerce sales representing 69.1% of global ecommerce [1].

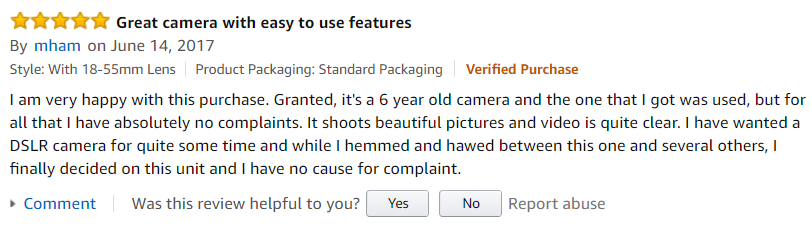
Ecommerce now a ubiquitous business continuously seeks ways to achieve increasing sales by giving online consumers necessary information to make the decision to purchase a product or service. A growing customer base also gave rise to an online consumer community in ecommerce platforms wherein customers publish their reviews about products and share opinions. These reviews are now an integral part of the ecommerce experience since they allow other users to make informed choices. Indeed, Deloitte Consumer Report revealed that 81% of people read reviews and check ratings. For the majority of consumers, family and friends, consumer reviews and independent experts are the most trusted sources of information [2].

A product review is a textual review of a customer, who describes the characteristics (e.g. advantages and disadvantages) of a product. A product rating on the other hand represents the customer’s opinion on a specified scale. A popular rating scheme in ecommerce platforms is the star-rating, where more stars indicate better ratings [3]. When a user browses for a particular product on Amazon’s web site, s/he would see the **top customer reviews** as he scrolls through the web page:



On purchase of a product, Amazon invites the buyer to rate the product and write a review. It also asks other users to vote if a review is helpful or not. As indicated on the image above, the top review for Nikon D5100 DSLR Camera with 18-55mm has been labeled helpful by 836 people [4]. However, the top review is of the year 2011 and it might not be reflective of the current state of the product in terms of quality, design, price etc. While prioritizing top customer reviews, Amazon relies on the number of helpful votes. Indeed, helpfulness of review is defined as the ratio of the number of users who found it helpful to the total number of users who had read and evaluated the review.

Amazon review sorting mechanism results in reviews with more votes always being in the front pages. This makes reviews that are generated later to have less chance to be seen or voted on, even if the review was more helpful. This causes absolute first-mover advantage, which can harm the enthusiasm of later reviewers [5]. Highlighting a recent review as shown below can offer updated information that users are seeking to make buying decisions.

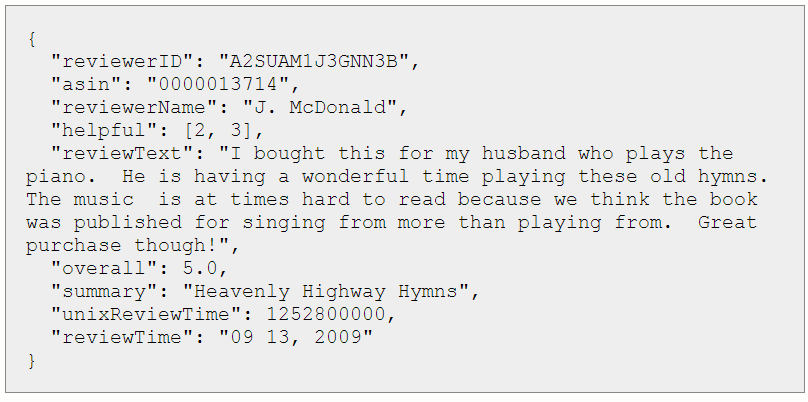


In this project, we first deep dived into a specific product category (Electronics) to unveil statistical trends how Amazon customers review and rate products. This provided us some insights into Amazon review system. Then, we suggested a new way to automatically classify a review as helpful or not without needing the users to vote on it. This would enable the system to present the most recent helpful review to the user.

**III. DATASET**

This project utilizes the review dataset, which was collected from one of the biggest e-commerce websites, Amazon, by Asst Prof Julian McAuley [6]. We acquired the dataset and permission to use it via an email request sent to Asst Prof McAuley at University of California San Diego. The raw dataset hosted on Stanford servers is 20 GB and contains 142.8 million reviews. The dataset is also provided as subsets of user review data, product review data, ratings only data, 5-core data in which all users and items have at least 5 reviews etc. Availability of dataset in different data sizes and subsets allows us the flexibility to experiment with the datasets in numerous ways.

Here is a sample review:



The review data consists of the following attributes:

● 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)

The product metadata consists of the following attributes:

● asin - ID of the product, e.g. 0000031852

● title - name of the product

● price - price in US dollars (at time of crawl)

● imUrl - url of the product image

● related - related products (also bought, also viewed, bought together, buy after viewing)

● salesRank - sales rank information

● brand - brand name

● categories - list of categories the product belongs to

The datafiles in json.gz format located on Stanford servers were uploaded into our S3 Amazon Web Services account via AWS Command Line. Amazon S3 provides storage through web services interfaces. We first retrieved the files with wget command line from Stanford servers then unzip the files with gunzip command. Finally, files were copied into S3 wer amazon review bucket with the following command line: aws s3 cp filename s3://wer-amazon-review/

To get acquainted with the content of data, we experimented with different data subsets in csv and json formats. As the final implementation of the project, we merged 2 datasets containing reviews and metadata of all product in all categories. Our final unzipped dataset was 40 GB containing 30 GB review data and 10 GB metadata.

**IV. CONTEXT AND LITERATURE SEARCH**

Our initial project proposal was inspired by a blog post by Max Wolf titled as “A Statistical Analysis of 1.2 Million Amazon Reviews” where Amazon Review data on Stanford Network Analysis Project was cited as a source of the data [7]. Indeed, this dataset has been the focal point on series of academic research papers studying online reviews.

For instance, Amazon review data was analyzed based on the consumer perspective by performing statistical tests to explore the influence mechanism of the reviewer, the review, and the existing votes on review helpfulness. This empirical study indicated that review helpfulness has significant correlation and trend with reviewers, review valance, and review votes [8].

Evaluating quality of reviews is a subject that has been also researched extensively. If we consider two examples of review from Amazon, we can showcase the characteristics of a good review:

● “In this review, I will focus on a feature that has not been covered well by other reviewers. There a few things you need to do to make it work. Firstly you have to enable the HDMI-CEC feature on your TV. HDMI-CEC is marketed under different names by different manufacturers...”

*(79 of 93 people found this review helpful)*

● “I don't have it, and won’t be getting it. It appears to be another canned apps streaming device. I had Google TV. Content was limited on google tv, and streaming from chrome tabs is in beta, and reported not to work very well...”  
*(1 of 76 people found this review helpful)*

A helpful review is likely to possess the following characteristics:

It provides a large quantity of detailed information about the product. For instance, in the first review in the above example, it not only gave information about how to set up the device, but also focused on the opinions that are different from other product description or reviews. Also, the sentence structure is clear and contains less spelling or grammar errors. The providers of these reviews tend to be active and received positive feedback from other consumers. In comparison, the less helpful reviews provide less information and add no additional value to the reader. Specifically in the second review above, the writer of that review mentioned that he didn't have experience with using the product he was reviewing, thus the information he provided was limited and vague.  
 Automatically evaluating the quality of online reviews has gradually attracted more and more attention in recent years. Most of the works have focused on automatically predicting the quality (helpfulness or usefulness) of reviews by using a set of observed textual or social features. Textual features include features that are based on text statistics such as length of the review, the average length of each sentence, percentage of nouns or adjectives, etc. Social features are information extracted from the reviewer's social context, such as the number of the reviews posted by this author, the past average rating for this author, etc. Most of the current works have formulated the problem of evaluating review quality as a classification problem or a regression problem using observed features [9]. For instance, in a final project presented by students of Columbia University, few machine learning algorithms were applied to predict review helpfulness [10].

**V. APPROACH AND TOOLS**

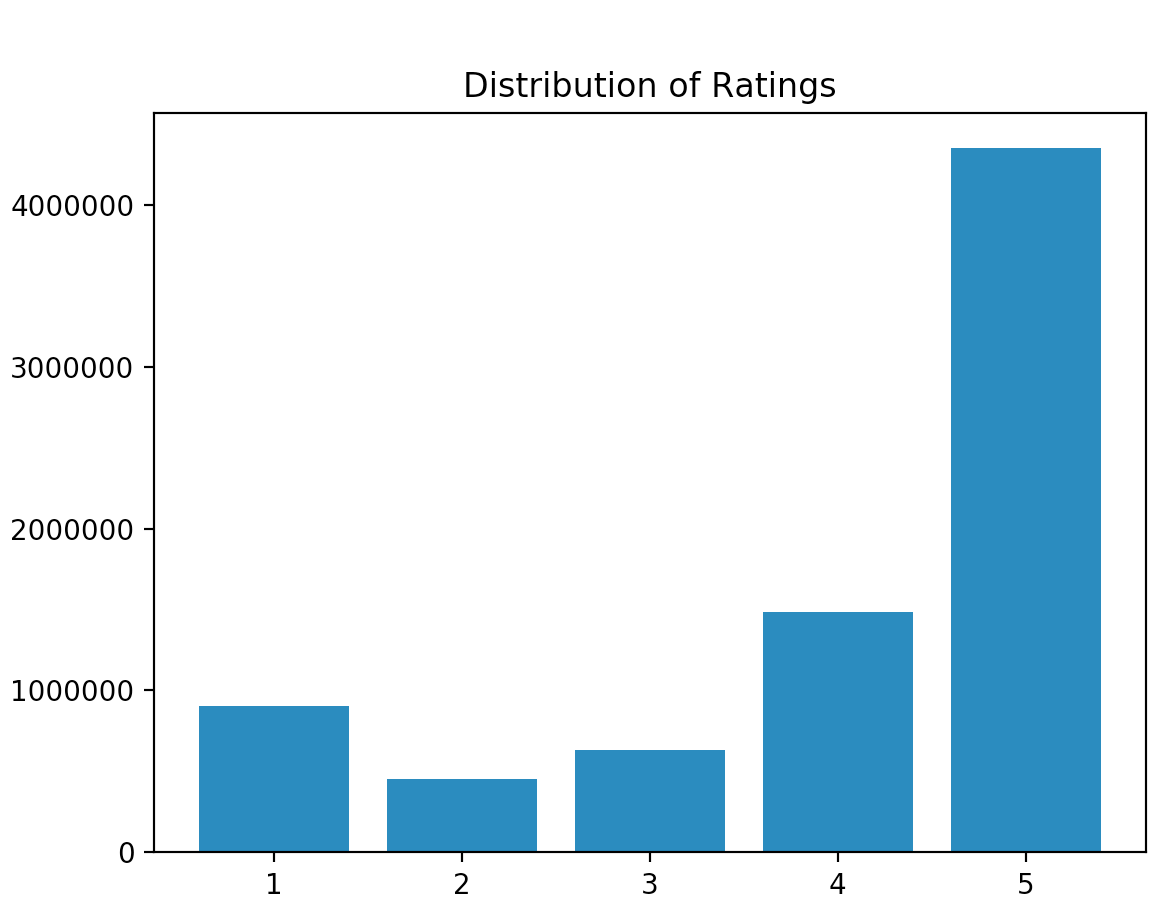
Due to large size of the dataset, we we first worked with a subset of data, specifically Electronics product category to explore the content and design the prediction model. We ran some of our analysis locally in our personal computers. In later stages, we implemented our code to Amazon Web Services Elastic MapReduce (EMR) environment. Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances. We configured 1 master and 5 slaves of m4.xlarge instances with 16vCore and 32 GiB memory in Frankfurt zone with the following softwares installed: Hadoop 2.8.3, Zeppelin 0.7.3, Ganglia 3.7.2, and Spark 2.2. While creating the cluster, the following configuration script for Python 3 was required in Cluster Advanced Settings Software step: [{"classification":"spark","properties":{"PYSPARK\_PYTHON":"/usr/bin/python3","maximizeResourceAllocation":"true"}}]

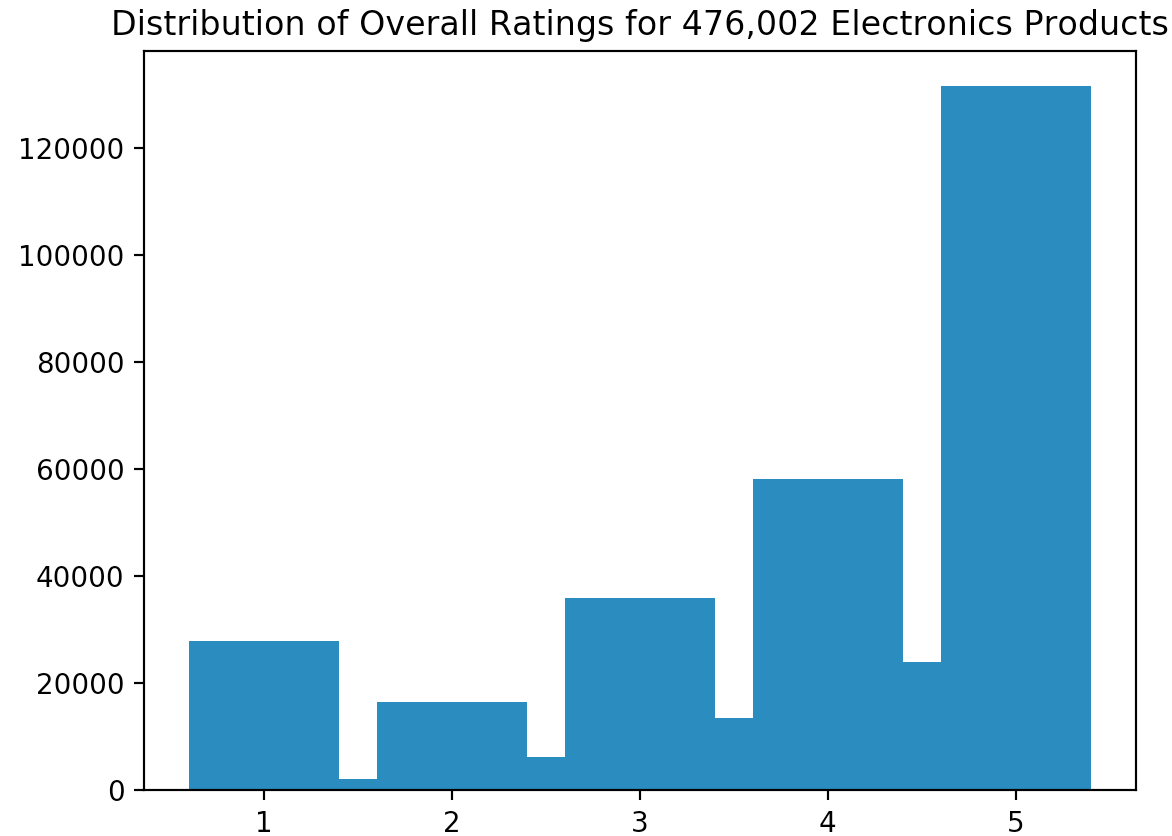
As mentioned in Dataset section, our datasets were uploaded into Amazon S3 which is offered by Amazon EMR as a file systems that can be used when processing cluster steps. This allowed EMR cluster to access data from S3 with a few lines of code written in Python through Spark Python API (Pyspark). For code implementation, we used Apache Zeppelin as a notebook for interactive data exploration and visualization tool on Amazon EMR. We used both Resilient Distributed Datasets (RDD) and Dataframes in Apache Spark to process our data in a distributed fashion. Spark natively supports applications written in Python, and it also includes libraries for SQL ([Spark SQL](https://spark.apache.org/sql/)) and machine learning ([MLlib](https://spark.apache.org/mllib/)).

**VI.DATASET EXPLORATION**

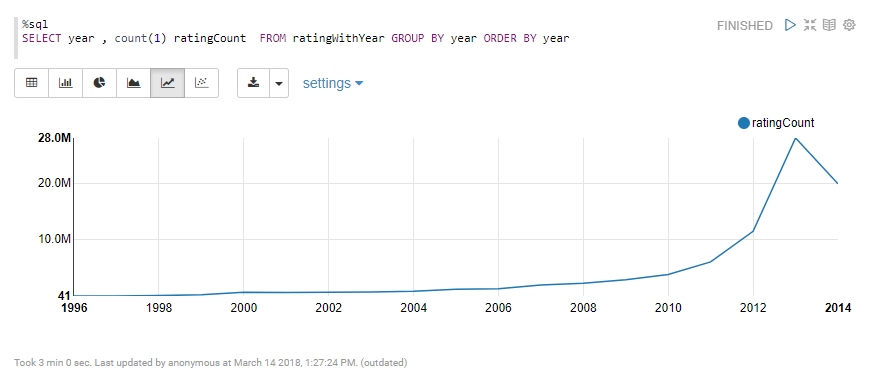
As a part of the exploratory data analysis, we summarized the dataset in different ways such as number of unique consumers and products, rating distribution etc. This part of the analysis gave us some familiarity with the dataset attributes. We implemented our code on csv and json file formats using both Resilient Distributed Datasets and Dataframes coupled with Spark SQL on Amazon EMR. Some exploratory statistics results are shown below:

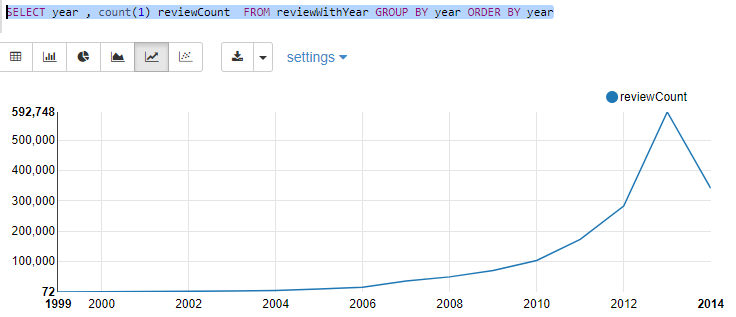
* 7,842,482 Total Reviews
* 4,201,696 Unique Reviewers
* 476,002 Unique Products
* Category Average Rating ~4.01 Variance ~1.91





In addition, distribution of ratings and reviews by year fluctuation starts in 2012 and arrive the top point in 2013. Cause of “Top Reviewers On Amazon Get Tons Of Free Stuff” movement.





**VII. FEATURE ENGINEERING**

The raw dataset contains a lot of attributes requiring some data preprocessing like eliminating/filtering some of which not helpful in our existing context. The various features that we are able to obtain from the dataset are:

● ReviewText

● SummaryText

● UnixReviewTime

● Overall Product Rating

● Helpfulness score

● Product ids

Using these features we generated new features which gave more meaning to the data:

● Using the ReviewText, we generated the **term frequencies** of text words and a **review length**. In order to generate the term frequencies, we first eliminated stopwords and used TFIDF and CountVectorizer to tokenize the text.

● Using the SummaryText, we generated a **summary length**

● Using the Product ids, we calculated the **count of the reviews** for each product

● We converted the helpfulness score (e.g. [a,b] where ‘a’ out of ‘b’ users found the review helpful) into a binary classification ‘hfactor’ having 1 if a/b is greater than 0.7 else 0. This feature is the output value for our project.

**VIII. PREDICTING HELPFULNESS SCORE**

The data was divided into training (70%) and testing data (30%). We used various classifiers on our disposal from scikit-learn package to predict helpfulness score and evaluated our prediction accuracy with multiple methods [11].

The various classification models we are looking into are as follows:

**Multinomial Naive Bayes**

Naive Bayes classifiers are simple probabilistic classifiers based on applying Bayes theorem with strong independence assumption between features. These classifiers are highly scalable. Tf-Idf vectors are often used as training and testing features for these classifiers since one of their main uses is in text categorization. Naive Bayes assumes conditional independence between features. The accuracy score for classification by this model is the highest among three models we used, thus yielded the worst model to predict the helpfulness score.

**Random Forest Classifier**

A random forest is a collection of decision trees that have been trained on randomly selected subsets of the training instances and explanatory variables. Random forests usually make predictions by returning the mode or mean of the predictions of their constituent trees; scikit-learn's implementations return the mean of the trees' predictions. Random forests are less prone to overfitting than decision trees because no single tree can learn from all of the instances and explanatory variables; no single tree can memorize all of the noise in the representation.

**Gradient Boosted Classifier**

The Gradient Tree Boosting is a classification algorithm which is a generalized ensemble model. It transforms the model in a forward stage-wise method. The algorithm trains the model by using multiple decision trees. Every successive tree is trained by using the previous tree. This minimizes the error rate. This algorithm is particularly good for the project because the scale of feature values does not affect it, so there is no need to normalize the textual features. We will apply the algorithm by using the library implementation of the Gradient Tree Boosting from scikit. The output of the Gradient Tree Boosting algorithm is generated in the form of probabilities which is passed to the metric functions. This algorithm is comparatively more complex than the other algorithms.

**IX. EVALUATION OF RESULTS**

We used accuracy score to compare the models used. The accuracy score is the number of correctly predicted values from the total number of values. We obtained the following results for our features.

|  |  |
| --- | --- |
| Model | Accuracy Score |
| Naive Bayes | 0.527796 |
| Random Forest Classifier | 0.291337 |
| Gradient Boosting Classifier | 0.284403 |

We concluded that Gradient Boosting classifier produced the best results with a value of 0.284 for Accuracy score.

**X. CONCLUSION**

In this project, we demonstrated how we can automatically calculate the helpfulness of a review using various lexical and quantitative data to derive features along with various classification algorithms. Our results can be considered on the right track since we accurately predicted approximately 70-75% of the test sample using various classifiers.

However, to improve accuracy, the scope of the project may be extended further in multiple ways. We may attempt to improve the accuracy of classifiers by engineering better features. Testing which features most impact the output prediction and parameter tuning via grid search method would be beneficial in the next steps. In addition, cross validation and further evaluation with different metrics can be done as well. We can incorporate characteristics of reviews such as product categorization and product features mentioned in reviews into the features.

Moreover, we may modify the project to be able to predict helpfulness on a scale from helpful to very helpful. This would involve using multi class classification algorithms. Furthermore, identifying the spam reviews and reviewers may further improve the accuracy of our model and be worth to focus our efforts to.

**ACKNOWLEDGMENT**

We would like to thank Assoc. Prof. Dr. Altan Çakır for his guidance throughout the course. We would also like to thank Professor Julian McAuley at UCSD for allowing us to use the Amazon review data for this project. Additionally, we would like to thank to the Foreks (Strategic Partner with ITU MSc in Big Data and Business Analytics Programme) for providing access to Amazon Web Services account.

**REFERENCES**

[1] Worldwide retail and ecommerce sales https://www.emarketer.com/Report/Worldwide-Retail-Ecommerce-Sales-eMarketers-Estimates-20162021/2002090

[2] Deloitte Consumer Review 2014 https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/consumer-business/consumer-review-8-the-growing-power-of-consumers.pdf

[3] Importance of online product reviews from a consumer’s perspective, Georg Lackermair , Daniel Kailer, Kenan Kanmaz. http://www.hrpub.org/download/201307/aeb.2013.010101.pdf

[4] Nikon D5100 DSLR Camera with 18-55mm f/3.5-5.6 Auto Focus-S Nikkor Zoom Lens (OLD MODEL)

https://www.amazon.com/Nikon-Camera-18-55mm-3-5-5-6-Focus-S/dp/B004V4IWKG/ref=cm\_cr\_arp\_d\_product\_top?ie=UTF8

[5] [8] Analysis of review helpfulness based on consumer perspective. Yuanlin Chen, Yueting Chai , Yi Liu, and Yang Xu. 2015. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7128942

[6] Amazon product data http://jmcauley.ucsd.edu/data/amazon/links.html

[7] Max Woolf’s Blog http://minimaxir.com/2014/06/reviewing-reviews/

[9] Automatically predicting the helpfulness of online reviews. Yadong Zhang. 2014 https://csus-dspace.calstate.edu/bitstream/handle/10211.3/121581/Report\_YZhang\_revised2\_final.pdf?sequence=2

[10] http://systemg.research.ibm.com/bigdata/reports/201412/report-201605-12.pdf

[11] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011

[12] Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. Ruining He, Julian McAuley http://cseweb.ucsd.edu/~jmcauley/pdfs/www16a.pdf