HELPFULNESS PREDICTION OF PRODUCT REVIEWS USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

BL.EN.P2CSE17013 KAGOLANU TRSHUL

in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

JUNE 2019

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF ENGINEERING, BANGALORE, 560035

BONAFIDE CERTIFICATE

This is to certify that the Dissertation (16CS799) report entitled " **Helpfulness Prediction of Product Reviews using Machine Learning** " submitted by "KAGOLANU TRISHUL (BL.EN.P2CSE17013)" in partial fulfillment of the requirements for the award of the **Degree Master of Technology** in "**Computer Science and Engineering"** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Bangalore.

SUPERVISOR	CHAIRPERSON
Dr. Srinath R. Naidu	**Dr. Amudha J**
Associate Professor	Chairperson
Dept. of Computer Science & Engineering	Dept.of Computer Science & Engineering
Amrita School of Engineering, Bengaluru	Amrita School of Engineering, Bengaluru

This project report was evaluated by us on ....

EXAMINER 1 EXAMINER 2

ACKNOWLEDGEMENT

I would like to thank my project guide Dr. Srinath R. Naidu for his guidance and support in performing the research project. I also would like to thank the panel members for their evaluation, inputs and advice towards the project.

ABSTRACT

Customers express their opinion on products through reviews. Since there will be a lot of reviews that will be posted, only those reviews which are helpful to the customer need to be identified and should be accessible to the customer. Hence, helpfulness of review needs to be predicted. In this work, the helpfulness votes received by review is taken as a validator for helpfulness and this score is predicted. The features are divided into three categories namely reviewer features, review text features and review meta data features. Machine Learning is used for prediction of helpfulness using these features. The algorithms Linear Regression, Random Forests and Extreme Gradient Boosting are used for prediction and the problem is taken as a regression problem. It is observed that rating of a review has the highest influence on predicting helpfulness followed by user average rating deviation, difficult words and positive words. The work defines features such as stem sim length and lem sim length which are derived from the product description which also have performed reasonably well. Certain readability features of review text had high inter-feature correlation among themselves. In most of the cases Extreme Gradient Boosting showed the best performance. Using all the features with Extreme Gradient Boosting algorithm for prediction gave the best performance in automatically predicting helpfulness. The work also defines an Inter-feature Correlation Graph which helps in feature-set size reduction which in turn leads to optimization. The revised feature-set performed almost as good as the total features in predicting helpfulness.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ABBREVIATIONS & NOMENCLATURES

1. RMSE: Root Mean Squared Error
2. MAE: Mean Absolute Error
3. User / Reviewer: The person who is the author of reviews of products
4. LR: Linear Regression
5. RF: Random Forest Regression
6. XGB: Extreme Gradient Boosting.

CHAPTER 1

INTRODUCTION

In today's world, which is the era of internet, people express their opinions about a product, service or event through various platforms. E-commerce is an area where customers give reviews about products. Consumers generally have to make purchase decisions based on incomplete information about a product. These reviews are helpful for customers who want to buy a product and to manufacturers or business owners who want to assess the performance and quality of their product. In general, the customer reviews are provided in addition to product descriptions, product suggestions etc.

As there are large number of reviews present for the products, it would become difficult for the customers to go through all the reviews. Also, as there is no editorial or quality control imposed on these reviews, there is a drastic variation in the quality of the reviews which can range from being of high-quality to extremely pointless. Hence, it is ideal that the customers should read few good quality and helpful reviews than reading all of them. A Helpful customer review can be defined as a judgement and analysis of a product by a customer that expedites the process of deciding to buy the product by other customers [6]. The websites generally ask the readers of a review whether they found the review as a helpful one or as unhelpful. These helpfulness votes are used for determining the helpfulness of a review and in general, it is calculated as the proportion of helpfulness votes received upon the total votes given to that review [11].

However, using the helpfulness votes does not completely solve the problem and this methodology has its own issues. Most of the reviews in the e-commerce websites have very less votes given to them and some are not given any votes at all. The products which are not popular have reviews which might not be seen by many users. The listing of reviews also suffers from Matthew effect [10] which states that a popular review always stays popular and highly voted as it is easily visible to most of the users and they might get inclined to the review and a review at the bottom remains at the bottom because it hardly gets noticed. Also, review posted earlier has a high chance of getting higher number of votes than a recent review which may get fewer votes. Hence, there is a need to automatically predict the helpfulness of a review and rank them according to this prediction so that the *customers have access to the most helpful reviews.

A model needs to be developed which can take the review data as input and predict its helpfulness. Machine learning models which can automatically learn from the data and predict helpfulness can be implemented.

CHAPTER 2

LITERATURE SURVEY

Helpfulness prediction of product reviews has seen significant research interest over the years. The process of helpfulness prediction of the reviews is performed in four stages namely Dataset collection, Feature extraction, Prediction and Evaluation. In prediction, generally machine learning algorithms are used and respective metrics are used in evaluation. Such procedure is done works such as [24], [25].

2.1 Datasets

The data needs to be collected for the required analysis. Generally, publicly available datasets are taken, or the data is crawled from the respective websites. Amazon data is a popular choice among researchers for all the related studies. The Julian McAuley Amazon product review dataset [8,9] was used for the analysis in [1] & [3]. Data was crawled from the amazon website in [1], [2] & [5]. Reviews of products of various categories were obtained and also the metadata information was also obtained in most of these cases.

2.2 Regression vs Classification

The problem of helpfulness prediction can be taken as a regression problem or a classification problem or a combination of both. In the scenario of regression, the helpfulness score is predicted by the model whereas in the scenario of classification, the helpfulness is segregated into classes and the class to which a review belongs to is predicted. This was taken as a complete regression problem by [1] & [3]. In [5] this was taken as a classification problem. From the data, the helpfulness score (P value) was calculated and a threshold ‘t’ was set for each category ranging from “Extremely Helpful” to “Not at all Helpful” as shown in table 2.1. In [2], a combination of classification and regression was used. The prediction was performed in two stages with classification followed by regression. In classification stage, reviews are organized into “low-quality” and “high-quality” reviews. In regression stage, the helpfulness score was predicted only for those reviews that fell into the category of “high-quality” reviews.

Table 2.1 Class labels for helpfulness scores

P value	Classes (Label)
> 80%	Extremely Helpful
60% to 80%	Very Helpful
40% to 60%	Somewhat Helpful
20% to 40%	Not very Helpful
< 20%	Not at all Helpful

2.3 Feature Extraction

Feature extraction is the most crucial step and the results highly depend on what features are extracted, how they are extracted and how they are represented. The features can be broadly classified as Review Text, Reviewer and Review metadata.

2.3.1 Review Text

The behavior of the review and its characteristics can be obtained from the review text. [1] have classified the variable further into linguistics, Psychological, Summary Language and Text complexity. They have proposed pronouns, article words, prepositions and auxiliary verbs in the review text. They have also considered the length of the review as word-count, character-count and sentence-count. [1] have considered readability variables as another parameter they indicate the effort required by the readers of a review to read and understand the review. These include Automated Readability Index (ARI), Gunning Fog Index (GFI Coleman-Liau Index (CLI), Flesch-Kincaid Grade Level (FKGL), Flesch-Kincaid Reading Ease (FKRE). In [2] we see review text variables such as unique word count which is referred as set length, Wrong Words, Lex Diversity which is the proportion of unique words in the review along with Noun, Adjective and verb and One letter Words. Also, they have used Flesch Reading Ease and Dale Chall

Reading which indicate the ease of reading of the reviews. [3] have considered psychological variables such as ‘Analytic’ which is the level of formal, logical and hierarchical thinking, ‘Clout’ which indicates the level of expertise and confidence in the review, posEmo which is the amount of positive words in the review, negEmo which is the amount of negative words in the review. They have also considered linguistic variables such as WordCount, Words Per Sentence (WPS) which is the average word-count per sentence indicating the conciseness along with readability of sentences in the review and ‘Compare’ which is the proportion of comparison words such as ‘better’, ‘smaller’ etc. and is useful in situations where products or its features need to be compared with other products. [5] have extracted the positive and negative sentiment words from the review text and calculated the features subjectivity, polarity, neg_refs_per_ref, pos_refs_per_ref, senti_diffs_per_ref whose definitions are shown in figure 2.1.

$$polarity = \frac{p-n}{p+n}$$

$$subjectivity = \frac{n+p}{N}$$

$$pos\_refs\_per\_ref = \frac{p}{N}$$

$$neg\_refs\_per\_ref = \frac{n}{N}$$

$$senti\_diffs\_per\_ref = \frac{p-n}{N}$$

p: number of positive words
n: number of negative words
N: number of words

Figure 2.1 Definitions of sentiment features

2.3.2 Reviewer

Along with the review text of the review, the information regarding the reviewer may also be useful in estimating how helpful the review of the reviewer can be. Profiling the activity of the reviewer is an important aspect of this. [1] has introduced two variables of reviewers namely productivity score and helpfulness per day. They have stated the importance of temporal dimension in making a prediction. Reviewer Helpfulness per day is computed as the ratio of aggregate helpfulness attracted by the reviews on the number of days between which the user was active in giving reviews. This parameter represents how frequently a customer writes reviews which have received high helpful votes. This implies that reviewers having high helpfulness per day make more helpful reviews available to the customers that help them making better purchase decisions. Productivity score adds the temporal dimension to the reviews produced by the reviewer. With the passage of time, several reviews from the reviewer get accumulated. So, combining data of helpfulness votes with this will give us the productivity information of the reviewer. The cumulative productivity score is total amount of reviews given by a reviewer multiplied by corresponding total helpful votes received for these reviews. This divided by the number of days between the first review and recent review gives us the productivity score per day of the reviewer. The formulae for these features are given in figure 2.2.

Helpfulness per day = cumulative helpfulness / number of days between first review and recent review

Productivity score = (total reviews * total helpful votes) / number of days between first review and recent review

Figure 2.2. Formulae for Productivity score and Helpfulness per day

Similarly, in [5] they have considered Reviewer's ranking, helpful percentage and review No. as the features. Reviewer's ranking is posted on Amazon reviewer's profile page. It is calculated by the overall helpfulness of the reviews taking into consideration the amount of reviews written by the customer. Helpful Percentage, the percentage of helpful votes received by the reviewer's previous reviews. Review No. is the amount of prior reviews the reviewer has written. These features take the history of the reviewer's performance to predict how helpful the upcoming reviews of the reviewer will be. The usage of these features is based on the assumption that the reviews given by a reviewer are almost of the same quality. In

[7], they have considered features like reviewer expertise for predicting helpfulness. As the dataset is movie data, the similarity of a movie with the previous movies watched by the reviewer is calculated which indicates the reviewer expertise and is used for predicting the helpfulness of this review.

2.3.3 Review Metadata

Apart from the review text and reviewer variables, the information obtained from the metadata of the review also is significantly useful in predicting the helpfulness of a review. In [2], they have considered the customer question-answer data and product description data for the analysis. The two features extracted are DescSim and QASim. DescSim is the similarity between the review text and the product description. It is calculated as the cosine similarity between the bag of words vectors of product description and review text. Similarly, questions from the question-answer data is taken and the cosine similarity between questions and the review text is calculated which is represented by QASim. [3]&[5] used rating of a review for prediction and considered it as a confirmatory variable determining review helpfulness. Age of a review which is the amount of days since the review was posted is used in [5] for predicting the helpfulness stating that initial reviews tend to be more useful being comprehensive and giving detailed account of the aspects of the product which may leave little room for the newer reviews to contribute and comment about. [7] have pointed out the decay of helpfulness of a review over time. They have considered the Timeliness of a review as a parameter for helpfulness and hypothesized the exponential decay is the behavioural property of helpfulness of a review as formulated in Figure 2.3. where t0 is the release time of the movie, H_3 is the estimated helpfulness, t is review publishing time, d and β are parameters in the model to be estimated. The rate of decay of helpfulness is controlled by β.

$$\hat{H}_3 = e^{-\beta(t-t_0)+d}$$

Figure 2.3 Helpfulness Decay

2.4 Prediction Model

After the features are extracted, the helpfulness of the reviews needs to be predicted based on these features. A model needs to be developed for the prediction. Machine Learning algorithms are popularly used in most of the cases. Prediction algorithms such as Classification and Regression Trees (CART), Non-Convex Penalized Quantile Regression (NC-PQR), Multivariate Adaptive Regression (MAR), Random Forest, Neural Network and Stochastic Gradient Boosting were used in [1] for predicting the helpfulness score. It was observed that stochastic gradient boosting has surpassed the other algorithms in estimating how helpful the reviews can be. In [2], helpfulness prediction of reviews is done in two stages. Firstly, the reviews are segregate into high-quality and low-quality reviews and the helpfulness score was predicted only for the high-quality reviews. The classification step was performed based on a threshold ‘t’ which is the average helpfulness score received by a review of a particular product. If the helpfulness score is above this threshold, the review is considered as high-quality otherwise it is considered as low-quality. Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB) were used for the classification. Gradient Boosting (GB) Regression and Linear Regression algorithms were used for the helpful vote prediction which was in turn used for ranking the reviews. Linear Regression (LR), Support Vector Regression (SVR), Random Forest and M5P algorithms were used for helpfulness prediction in [3]. M5P is a decision tree algorithm for solving regression problems using the separate-and-conquer approach. SVM was used in [5] with sequential minimal optimization (SMO) for training the support vector classifier for classifying the reviews from “Extremely Helpful” to “Not at all Helpful”.

2.5 Evaluation Methods

The evaluation needs to be done once the model makes prediction of helpfulness. [1] have used 10-fold cross-validation for training and testing. Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Relative Root Mean Squared Error (RRSE) metrics were used to evaluate the helpfulness score predictions made by the model. In [2], recall, precision, F1-score and Receiver Operating Characteristics (ROC) curve were used to evaluate the classifier and Mean Squared Error (MSE) was used to evaluate the regressor. They performed train-test split of data in the ratio 3:1 for this purpose. Mean Absolute Error (MAE) was used by [3] for evaluating their regression model with 10-fold cross validation. In [5], the dataset was split as 66% training data and 34% testing data and precision with respect to "Extremely Helpful" category is considered as the evaluator of the model.

CHAPTER 3

SOFTWARE AND HARDWARE REQUIREMENTS AND SPECIFICATIONS

3.1 Hardware Specifications

Processor	Intel(R) Core(TM) i5-7200U CPU Base Speed: 2.71 GHz Cores: 2 Logical Processors: 4
RAM	8GB
Disk	2.5GB

3.2 Software Specifications

Operating System	Windows 10
Platform	Anaconda
Graphical User Interface/ Development Environment	Jupyter notebook
Programming Language	Python

3.3 Packages

Sno	Package Name	Version	Description
1.	numpy	1.14.2	This package expedites the operations and representation of mathematical objects like multi-dimensional arrays.
2.	pandas	0.23.4	Library for data analysis and data manipulation.
3.	scikit-learn	0.19.0	Library containing Machine Learning Models and Performance Measuring Metrics
4.	nltk	3.2.5	Package containing tools related to natural language processing
5.	textstat	0.5.1	Text analysis tool for python containing mainly readability tests of text
6.	swifter	0.260	Package for faster execution of functions on pandas dataframes
7.	matplotlib	2.0.2	Library for plotting simple graphs
8.	seaborn	0.8.0	Library for plotting high-level graphs
9.	xgboost	0.81	Library for Extreme Gradient Boosting implementation

CHAPTER 4
DESIGN

The following section describes the design and architecture followed in implementing the helpfulness prediction system. Firstly, the data is collected and filtered followed by feature extraction. The feature extraction is divided into three categories. The first one is the reviewer features which are the variables extracted based on the reviewer's activity over time. The second category is review text features. Here the features are extracted from the text of the review which mainly focus on the content of text and the way it is written. The third category is review meta data. This includes the data that is present with the review apart from the review text which further describes the review entry and may also have the information of the product on which the review is on. Finally, machine learning is used to predict the helpfulness score where it is taken as a regression problem. The flow of the implementation is described in figure 4.1.

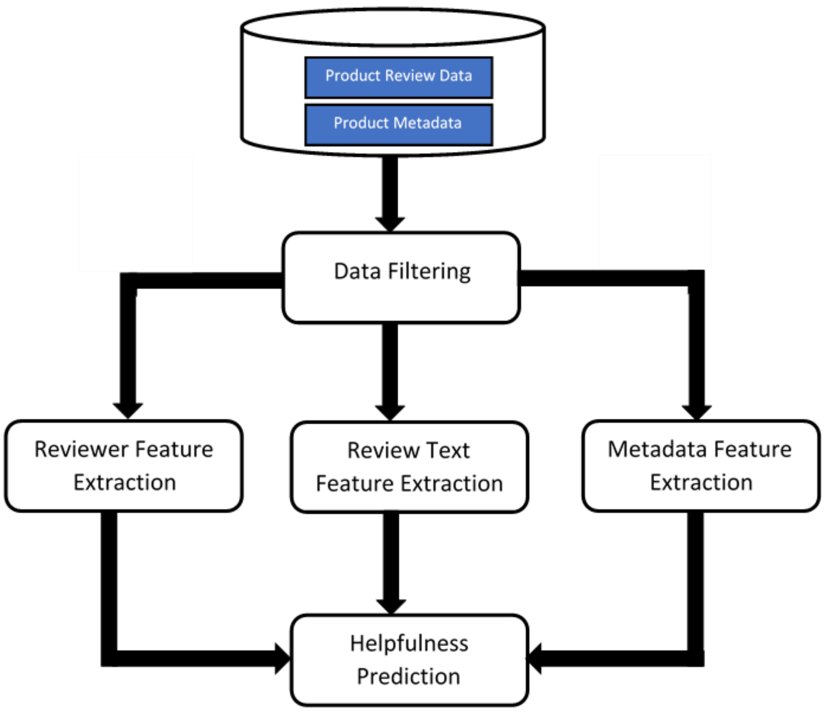

Figure 4.1 Architecture of Proposed System

CHAPTER 5
SYSTEM IMPLEMENTATION

5.1 Data Collection

The data taken is product reviews from Amazon.com. The dataset is extracted from Julian Mcauley, UCSD website [8], [9] and is of category ‘Electronics’. Along with that ‘Electronics’ Metadata is also extracted which contains the metadata about the products on which the reviews are written. After that, only those reviews which received at least 10 votes are considered for analysis. Furthermore, reviews for products which don’t have a description are only used for measuring user-related features and are not-at-all used anywhere else. The distribution of helpfulness score in the data is depicted in figure 5.1.

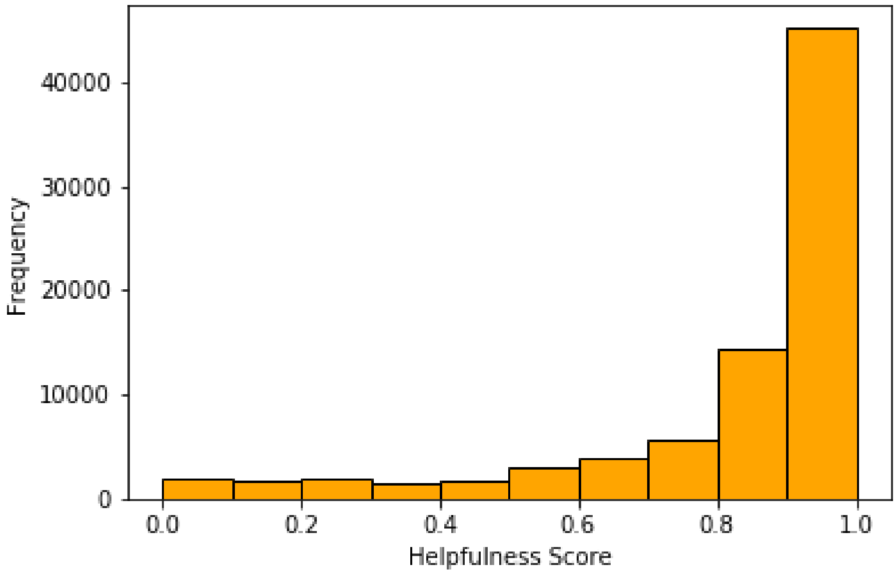

Figure 5.1 Distribution of Helpfulness Score over Data

5.2 Feature Extraction

A total of 21 features categorized into reviewer, review text and review metadata are used for prediction. The features are User Average Rating Deviation, User Average Delay, No of Reviews, Reviewer Days, Difficult Words, SMOG Index, Flesch Reading Ease, Coleman-Liau Index, Linsear Write Formula, Gunning Fog Index, Automated Readability Index, Flesch-Kincaid Grade Level, Dale-Chall Readability Score, Review Length, Sentence Count, Words Per Sentence, No. of Positive Words, No. of Negative Words, Stem Sim Length, Lem Sim Length, Rating. The dependent variable here is helpfulness score which needs to be predicted using these features.

5.2.1 Reviewer Features

i. User Average Rating Deviation

This is the average of the deviation of a user's rating in his/her reviews from the average rating for a product. This feature is generally used in spam detection [12], [13]. Firstly, for each product the average rating given to it by all the reviews is calculated. Then, the deviation of the rating of a review from this average rating is calculated which is the absolute value of the review rating subtracted by average rating given to the product. After that, for each user, the average value of this rating deviation is calculated. This is calculated for the users over the entire data including training and testing data and also including those reviews for which the product description is unavailable

ii. User Average Delay

This feature represents the average time a reviewer takes to give a review for a product. Firstly, the unix time at which the first review was posted for each product is noted. Then, the time difference of each review with the first review of the product is calculated. Finally, for each reviewer the average of this delay of his/her reviews is calculated and stored.

ii. No. of Reviews

This feature reflects the previous experience in writing reviews. It is the number of reviews written by the user. Firstly, the reviews are grouped into users and for each user the review count is calculated.

iii Reviewer Days

This feature again reflects the experience of the reviewer in writing reviews but in terms of time. This is the number of days between first review written by the reviewer and the last one. Each review has a Unix timestamp associated with it. It is in terms of seconds. In order to calculate this, firstly, the reviews are grouped by the users. Then, for each user the minimum and maximum timestamps are noted down. The difference between the first timestamp and last timestamp converted to number of days is the required feature.

5.2.2 Review Text Features

i. Difficult Words

This feature measures the number of difficult words found in the review.
Difficult words are those words which are not used frequently, being specific and in general may be long. For this the function difficult_words from the package textstat was used. For each word in the review, this method checks for the presence of this word in a 3000 word long ‘easy word’ list. If it is not present in the list, then it is classified as a difficult word. The count of these difficult words is the required feature.

ii. SMOG Index

SMOG index, whose full-form is Simple Measure of Gobbledygook, is a readability measure to estimate the education in number of years required to understand a particular text [17].

iii. Flesch Reading Ease

The flesch reading ease gives an approximation of level of education needed to read text with ease which lies between a score of 1 to 100 [18]. Here, more the score is, more is the ease with which a text can be read and vice-versa. The indications of score can be seen in table 5.1.

Score	School level	Notes
100.00-90.00	5th grade	Very easy to read. Easily understood by an average 11-year-old student.
90.0–80.0	6th grade	Easy to read. Conversational English for consumers.
80.0–70.0	7th grade	Fairly easy to read.
70.0–60.0	8th & 9th grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0–50.0	10th to 12th grade	Fairly difficult to read.
50.0–30.0	College	Difficult to read.
30.0–0.0	College graduate	Very difficult to read. Best understood by university graduates.

Table 5.1 Flesch Reading Ease

iv. Positive and Negative Words

The number of positive words and the number of negative words in the review text are taken as features to predict helpfulness. Here, a list of 4800 negative words and 2000 positive words are taken from [14], [15]. The review text is split into words and presence of a word in a particular list tags it to the respective category. The count of these tagged words for each category are the features positive word no and negative word no.

v. Coleman-Liau Index

The coleman liau index is another readability score which measures the level of education required by a person to read a text [19]. But unlike most of the popular readability measures, it takes into consideration the characters of a word rather than the syllables in calculating the score. The formula is mentioned in figure 5.2.

$$CLI = 0.0588L - 0.296S - 15.8$$

L = average number of letters per 100 words

S = average number of sentences per 100 words

Figure 5.2 Coleman-Liau Index

vi. Automated Readability Index

This is a readability measure that predicts the level of education required to understand a text.Just like Coleman-Liau index, this measure also involves characters instead of syllables as calculation directly on characters instead of syllables is often faster [22]. The formula is shown in figure 5.3.

$$4.71\left(\frac{\text{characters}}{\text{words}}\right) + 0.5\left(\frac{\text{words}}{\text{sentences}}\right) - 21.43$$

characters: Number of characters and numbers

words: Number of spaces

sentences: Number of sentences

Figure 5.3 Automated Readability Index

vii. Gunning fog

This measure calculates the education level required to comprehend a text. It takes into consideration the average words per sentence and the number of complex words having three or more syllables of a passage in calculating the score [20]. The formula is shown in figure 5.4.

$$0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right)+100\left(\frac{\text{complex words}}{\text{words}}\right)\right]$$

Figure 5.4 Gunning Fog Index

viii. Dale Chall Readability Score

This is a readability test which indicates the difficulty in understanding a text [21]. It takes difficult words as a parameter for score calculation. The words which does not fall into the 3000-word list of easy words are considered to be difficult. The Formula is shown in figure 5.5.

$$0.1579\left(\frac{\text{difficult words}}{\text{words}}\times 100\right)+0.0496\left(\frac{\text{words}}{\text{sentences}}\right)$$

Figure 5.5 Dale Chall Readability Score

ix. Flesch-Kincaid Grade Level

This measure represents the US educational grade-level required to comprehend a text [18]. The score is calculated as shown in Figure 5.6.

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right)+11.8\left(\frac{\text{total syllables}}{\text{total words}}\right)-15.59$$

Figure 5.6 Flesch-Kincaid Grade Level

x. Linsear Write Formula

This is another measure to estimate the grade level required to understand a text. This is calculated based on number of words having three or more syllables and number of sentences [23].

xi. Review Length

This is the length of the review in terms of number of words.

xii. Sentence Count

This measure is the number of sentences in review text.

xiii. Words Per Sentence

This measure gives the average number of words in a sentence in the review text.

5.2.3 Review Meta Data Features

i. Stem Sim Length and Lem Sim Length

These features measure the ability of the review to comment on the product or aspects of the product. Here, the goal is to find out to what extent the review talks about the product and its aspects.

The features stem sim length and lem sim length attempt to analyze upto what extent the reviews talk about the product or its aspects and how relevant the review is to the context of the product. They basically measure the ability of the review to comment on the product or its aspects approximately. Product description was used for analysis in studies such as [2]. The process here mainly involves three stages.

The first stage deals with the product description. As the description is in the form of paragraph text, it is split into a list of words. After that, the stopwords are removed. Then, pos-tagging is done. Here, only the Nouns are extracted. Stemming of words is done and this list is stored. Similarly, lemmatization of words is done and the list is stored. This is shown in figure 5.7.

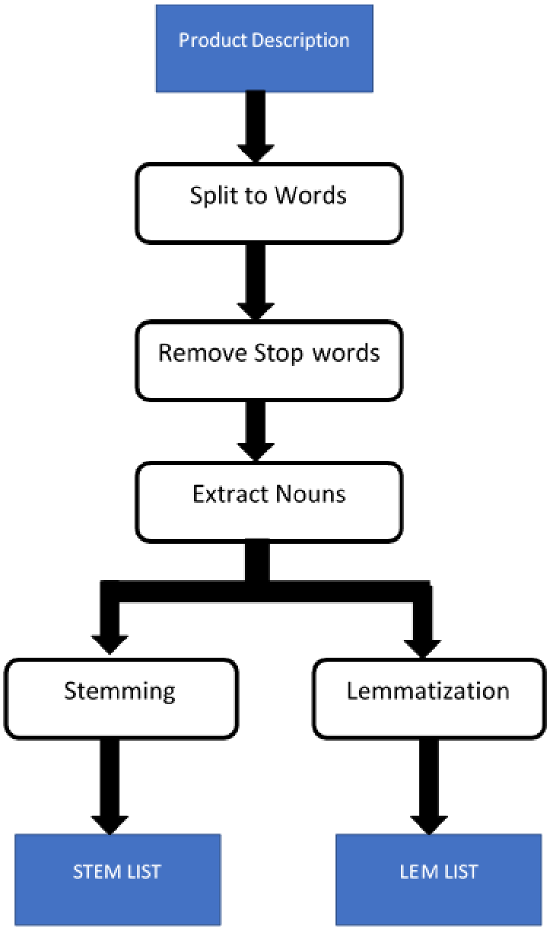

Figure 5.7 Word List Extraction from Product Description

The second stage deals with the review text. Firstly, the review text is split into words. Then, the stop words are removed. Finally, stemming and lemmatization of the words is done. The stemmed words list and lemmatized words list are stored separately. This is shown in figure 5.8.

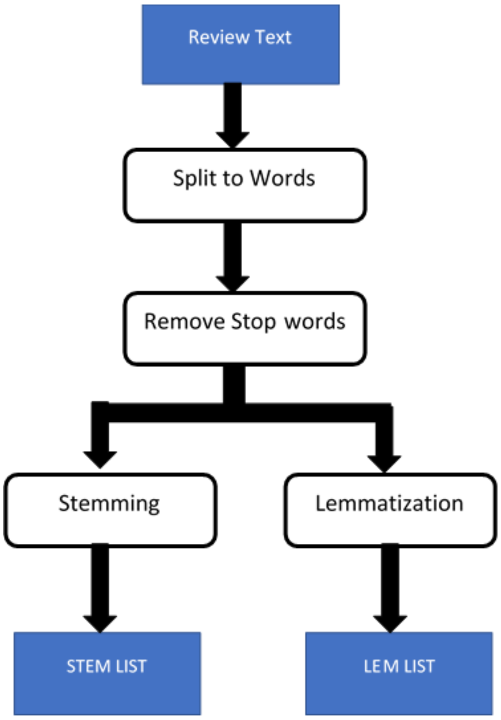

Figure 5.8 Word List extraction from review text

The third stage deals with the comparison of the description and the review. Here, a set intersection is done between the stemmed words of review and the stemmed list of the corresponding product description. The length of this set is the feature stem sim length. Similar procedure is applied for lemmatized words and the feature lem sim length is also extracted.

ii. Rating

This is the score given by the reviewer to the product. [3] has asserted its importance in predicting helpfulness of reviews. Here, it ranges between 0 to 5 and indicates the overall opinion of the user towards the product.

5.3 Helpfulness Prediction

After the features are extracted, they are used by a machine learning model to predict the helpfulness score which ranges from 0 to 1. The machine learning models used here are Linear Regression and Random Forests. The data is split into training data and testing data for the training and testing of the respective models. Scikit-learn package in python [16] is used for this purpose.

5.3.1 Linear Regression

It is popular machine learning algorithm. This algorithm is simpler and easy to interpret than other regression algorithms. It forms a model of linear relationship between the variables and the value to be predicted in the n-dimensional space. The goal here is error reduction.

5.3.2 Random Forests

Random Forests is an ensemble machine learning algorithm which comprises of a certain number of decision trees. In the case of regression, the mean of the predicted values of all the decision trees is the predicted output value by the model. The Random Forests model is used with attributes ‘n_estimators’ as 500 which is the number of trees used in the model and random state as 42 which is the seed used in random number generator in the model.

5.3.3 Extreme Gradient Boosting

Extending this work further, the machine learning algorithm Extreme Gradient Boosting (XGB) was used for predicting helpfulness. The ’xgboost’ package in python is used for this purpose. This is an ensemble algorithm, which uses decision trees as base learners and gradient boosting as the learning methodology. It was observed in many cases that XGB outperformed most of the modern machine learning algorithms in terms of speed of execution and performance of the model. There are various parameters for this model. The parameters focussed here are subsample size, learning rate and number of estimators. The subsample size denotes the fraction of rows that would be sampled randomly for constructing each tree. The learning rate represents the quantum of change that can be applied on the weights of features at each step. The third one is the number of trees used in the model. Grid Search method is used for parameter turning of XGBoost to get best possible result. This uses all possible combinations of parameters from a given set. The values used are shown in table 5.2. Further, instead of using the Grid Search package, a customized Grid Search is developed. It tunes the parameters based on RMSE and MAE with giving preference to RMSE.

Table 5.2 Parameter values used in Grid Search

Sno	**Parameter**	**Values**
1.	Learning Rate	0.01, 0.09, 0.1, 0.2, 0.5, 0.9
2.	No of Estimators	200, 300, 400, 500
3.	Subsample Size	0.3, 0.5, 0.9, 1

5.4 Performance Evaluation

The performance of the models is evaluated using the metrics Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). MAE is the average of absolute values of difference between the predicted and actual scores. In RMSE, the values of error which is the difference between predicted and actual score, are squared and averaged and root of this is taken.

5.5 Feature-set size Reduction using Inter-feature correlation Graph

One of the important goals in any experiment or project is optimization and efficient resource utilization. In this section, the target is to reduce the complexity of the model with least possible impact on the performance and results. Currently there are 21 features used for predicting helpfulness. So, reducing the feature-set size would reduce the complexity of the model and would lead to optimization. For this to be achieved with least effect on error Inter-feature correlation graph is created.

5.5.1 Inter-feature Correlation Graph

It is a Graph with the features being the vertices. There will be an edge between two vertices (features) if the correlation between the features is greater than or equal to 0.8.

5.5.2 Feature-set Reduction

There are multiple connected components in the graph. This may indicate that the features in a component are highly correlated with each other. Having more features from each component for prediction may not add significant value to prediction performance as each of these features mathematically represent almost the same data behaviour. So, retaining a subset of these features would not impact much on the prediction performance.

In the current work, one feature is selected from each of the connected components. Firstly, features from each connected component are grouped. These features are ordered based on their individual performance in predicting helpfulness. The feature giving the least error is chosen and finally the residual features are used for elpfulness prediction. The metric chosen here is RMSE.

CHAPTER 6

SYSTEM TESTING AND TEST CASES

The system testing is done through cross-validation. The dataset is divided into 60% training and 40% testing. Random state is 101 which is the seed of the random number generator for random selection of records into training set and testing set. Mainly, there are three sets of test cases to be considered. The first one being per-feature test cases where the predicting of helpfulness is tested and evaluated for each feature. Next is the category-wise testing where performance of each category namely reviewer features, review text features and review meta data features is being tested for helpfulness prediction. The third one is all-feature algorithm-wise testing. Here, all the available features are involved and prediction performance if tested for the algorithms of linear regression and random forests.

CHAPTER 7

RESULTS AND DISCUSSIONS

The experiments were conducted and results were recorded feature-wise, category-wise and using the entire feature set across different machine learning models.

7.1 Feature-wise Correlation with helpfulness Analysis

This section deals with the correlation analysis of each feature with helpfulness score. Higher the absolute value, higher is the correlation and higher the ability to explain the behavior of helpfulness by the feature in a linear fashion which is simple and easy to interpret. The correlation of helpfulness with each feature sorted in descending order of the absolute value of correlation is shown in table 7.1. It can be observed that the rating of the review (overall) has the highest correlation with helpfulness with a value of around 0.47. This is followed by user average rating deviation, Number of positive words and difficult words.

Table 7.1 Feature – Helpfulness Correlation

feature	correlation
overall	0.4752
pos_no	0.2232
difficult_words	0.2167
review_length	0.1961
lem_sim_length	0.1719
stem_sim_length	0.1704
sentence_count	0.1219
neg_no	0.1194
smog_index	0.1154
dale_chall_readability_score	0.1136
automated_readability_index	0.1110
flesch_kincaid_grade	0.1106
gunning_fog	0.1100
wps	0.1097
coleman_liau_index	0.1075
reviewer_days	0.0920
no_of_reviews	0.0718
linsear_write_formula	0.0141
user_delay	-0.0947
flesch_reading_ease	-0.1120
user_deviation	-0.3305

7.2 Feature-wise Results

Here, the goal is to analyze which feature predicts helpfulness with minimal error. The metrics used here are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The results represented by these metrics for Linear Regression, Random Forests and XGBoost are shown in Table 7.2. In the graph represented by figure 7.1, the RMSE of least of Linear Regression, Random Forests and XGBoost algorithms of the top ten features with least error is shown. It is observed that rating of the review gives the least error which is 0.2086 RMSE. This is followed by no of positive words, difficult words and user average rating deviation. In the graph represented by figure 7.2, the MAE of least of Linear Regression, Random Forests and XGBoost algorithms of top ten features with least error is shown. Similar to RMSE, it is observed that the rating of the review gives the least error which is 0.1441. This is followed by Number of Positive words, User Average Rating Deviation and Difficult words respectively.

Table 7.2 Feature-wise Results

Feature	**rmse-LR**	**mae-LR**	**rmse-RF**	**mae-RF**	**rmse-XGB**	**mae-XGB**
flesch_reading_ease	0.2365	0.1724	0.2394	0.1733	0.2341	0.1700
smog_index	0.2363	0.1721	0.2348	0.1707	0.2347	0.1706
flesch_kincaid_grade	0.2365	0.1724	0.2353	0.1711	0.2338	0.1697
coleman_liau_index	0.2367	0.1725	0.2367	0.1706	0.2337	0.1697
automated_readability_index	0.2365	0.1724	0.2385	0.1734	0.2339	0.1697
dale_chall_readability_score	0.2365	0.1724	0.2386	0.1731	0.2336	0.1698
difficult_words	0.2323	0.1701	0.2241	0.1622	0.2239	0.1620
linsear_write_formula	0.2380	0.1732	0.2333	0.1687	0.2336	0.1696
gunning_fog	0.2366	0.1724	0.2456	0.1775	0.2338	0.1697
sentence_count	0.2361	0.1720	0.2353	0.1711	0.2352	0.1710
Wps	0.2366	0.1724	0.2385	0.1734	0.2330	0.1691
review_length	0.2333	0.1707	0.2258	0.1637	0.2241	0.1620
pos_no	0.2320	0.1698	0.2221	0.1597	0.2221	0.1597
neg_no	0.2362	0.1724	0.2347	0.1711	0.2347	0.1711
user_deviation	0.2248	0.1601	0.2517	0.1717	0.2244	0.1596
user_delay	0.2370	0.1726	0.2396	0.1708	0.2366	0.1723
no_of_reviews	0.2374	0.1734	0.2365	0.1740	0.2365	0.1739
reviewer_days	0.2369	0.1724	0.2389	0.1726	0.2362	0.1718
stem_sim_length	0.2345	0.1710	0.2320	0.1685	0.2319	0.1684
lem_sim_length	0.2345	0.1709	0.2320	0.1684	0.2319	0.1683
Rating	0.2102	0.1463	0.2086	0.1441	0.2086	0.1441

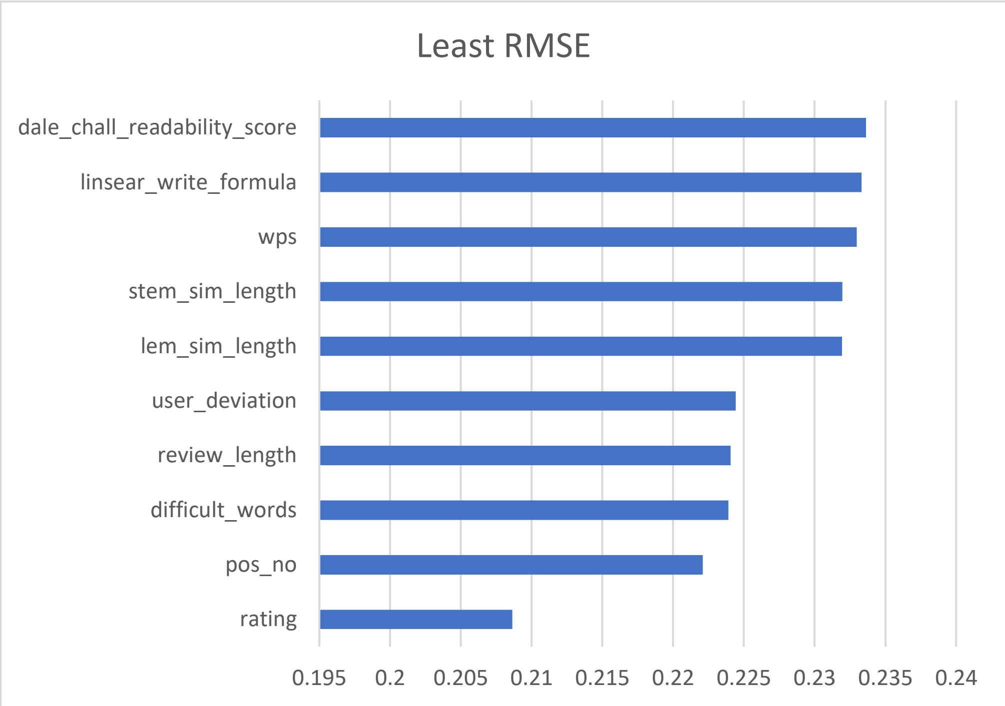

Figure 7.1 RMSE Top-10 Features

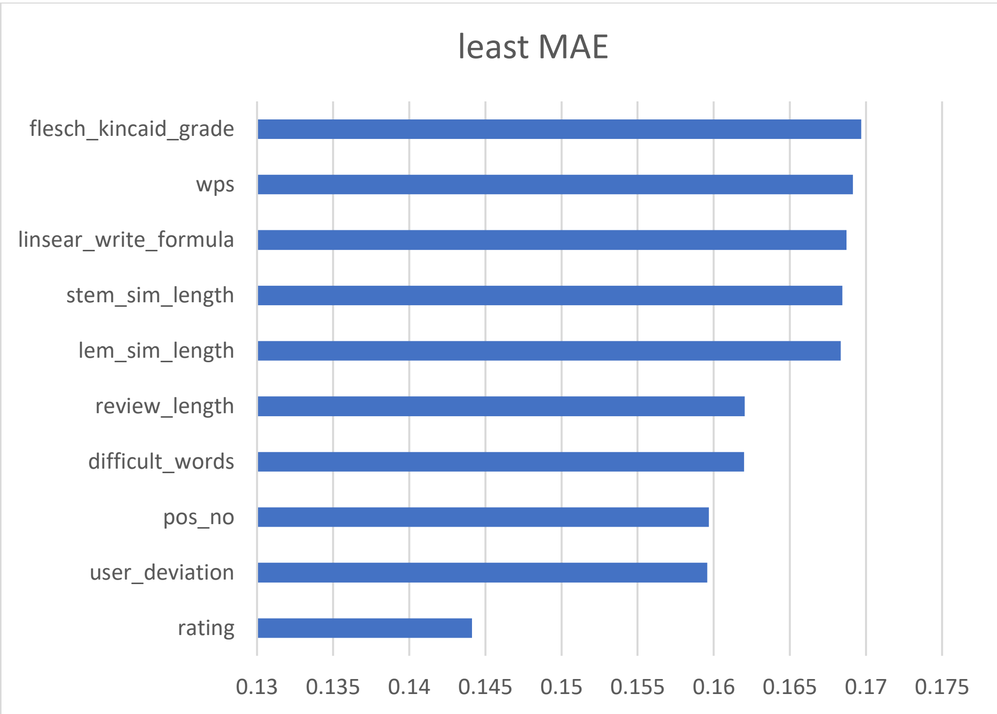

Figure 7.2 MAE Top-10 Features

7.3 Category-wise Results

This section covers the comparative analysis of helpfulness prediction results obtained for each category namely reviewer features, review text and review meta data and for machine learning algorithms linear regression, random forests and XGBoost for each of these categories. The RMSE and MAE for the machine learning models are depicted in table 7.3. In the graph represented by figure 7.3 the RMSE of least of linear regression, random forests and XGBoost of these categories is shown. In the graph represented by figure 7.4 the MAE for the same is depicted. In both the cases the meta data features showed the best performance with and RMSE of 0.2050 and MAE of 0.1417. This is followed by review text features.

Table 7.3 Category-wise Results

Category	**rmse-LR**	**mae-LR**	**rmse-RF**	**mae-RF**	**rmse-XGB**	**mae-XGB**
User Features	0.2235	0.1595	0.2364	0.1651	0.2220	0.1587
Review Text Features	0.2259	0.1652	0.2231	0.1627	0.2178	0.1561
Meta Data Features	0.2085	0.1457	0.2062	0.1427	0.2050	0.1417

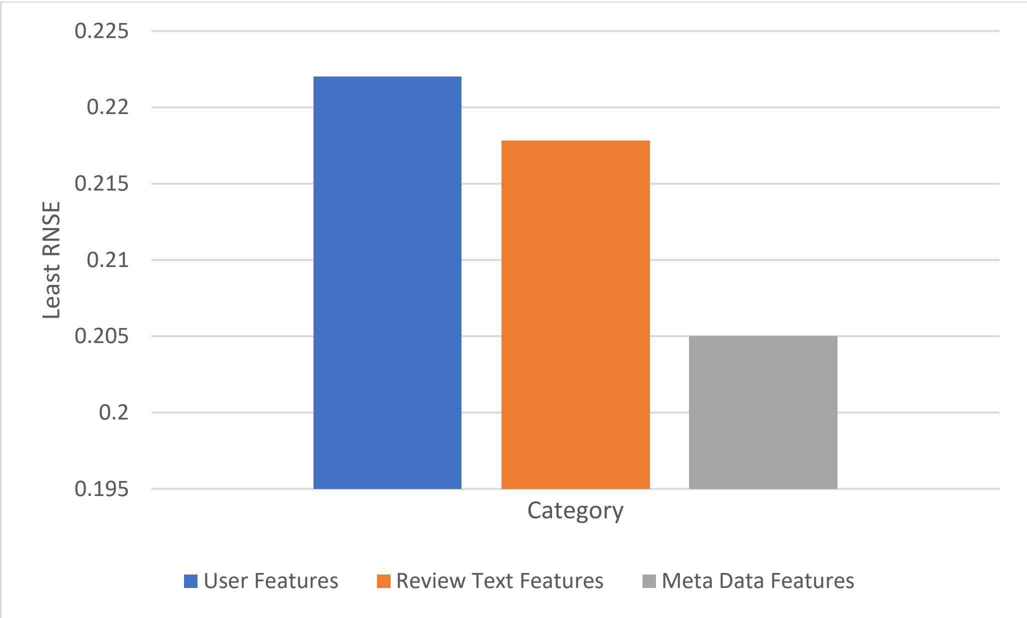

Figure 7.3 Category-wise RMSE

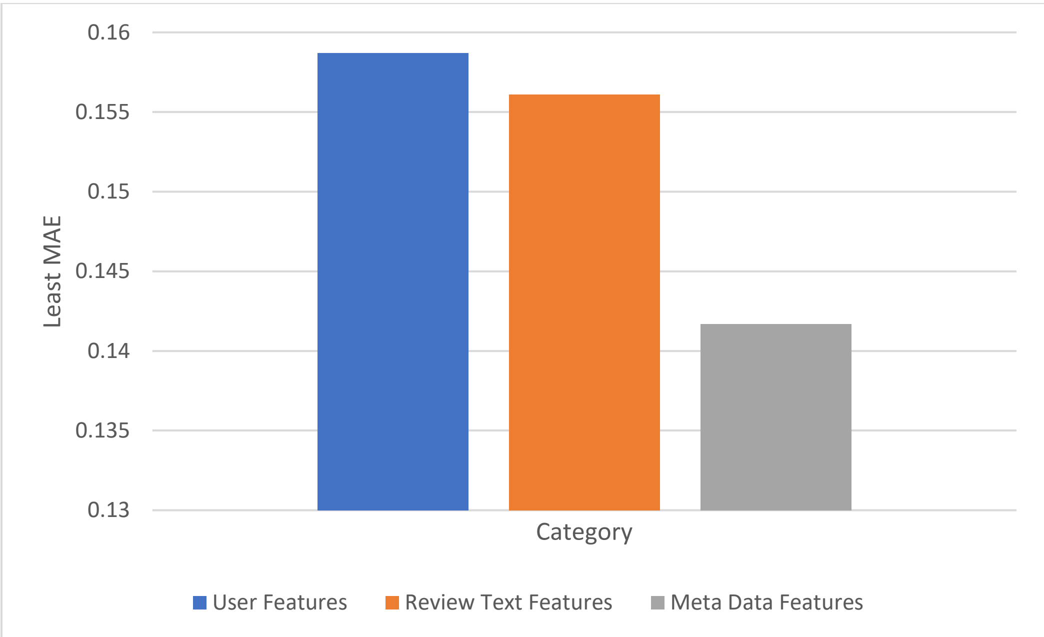

Figure 7.4 Category-wise MAE

7.4 Prediction with entire feature-set

Here all the features are considered for helpfulness prediction. Linear Regression, Random Forest, Extreme Gradient Boosting and the parameter tuned XGBoost using Grid Search are used as the prediction algorithms. The models are compared based on RMSE and MAE. This is depicted in table 7.4. XGBoost with Grid Search gives the best result of 0.1921 RMSE and 0.1319 MAE. A comparative visualization among the prediction models is shown in figure 7.5 for RMSE and figure 7.6 for MAE.

Table 7.4 Prediction Results with total features

Prediction Model	RMSE	MAE
Linear Regression	0.2010	0.1414
Random Forest	0.1944	0.1362
Extreme Gradient Boosting	0.1928	0.1329
XGBoost with Grid Search	0.1921	0.1319

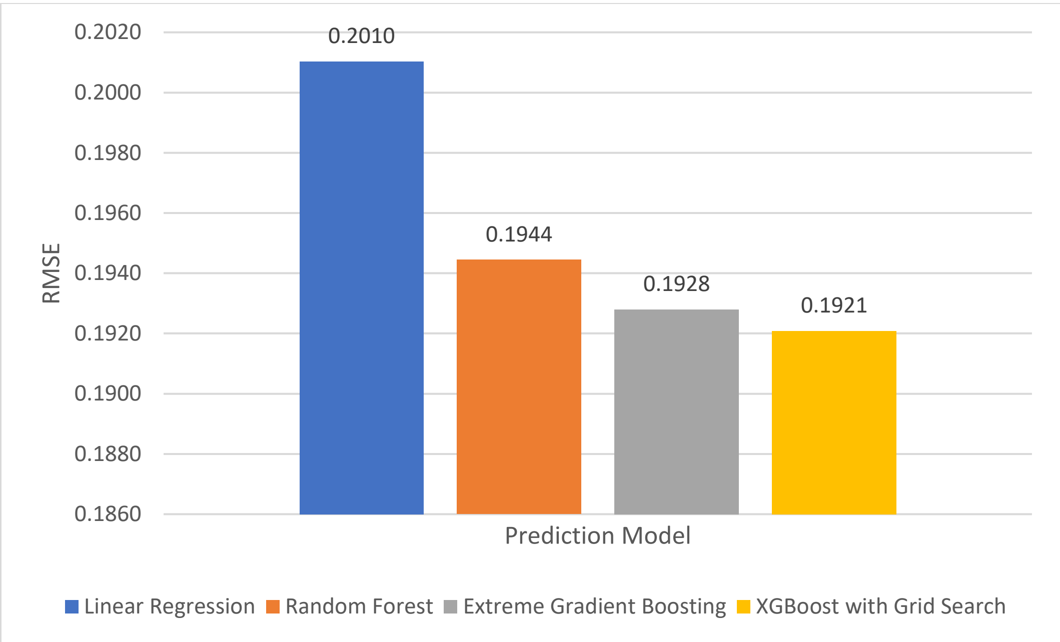

Figure 7.5 Machine Learning Algorithm-wise RMSE

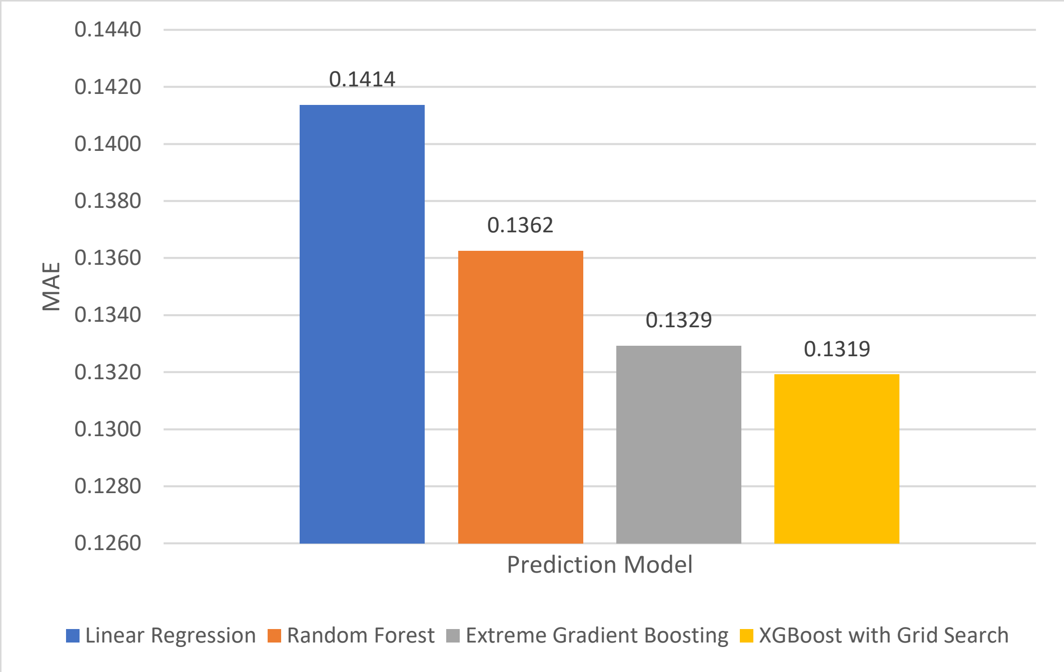

Figure 7.6 Machine Learning Algorithm-wise MAE

7.5 Inter-feature correlation analysis

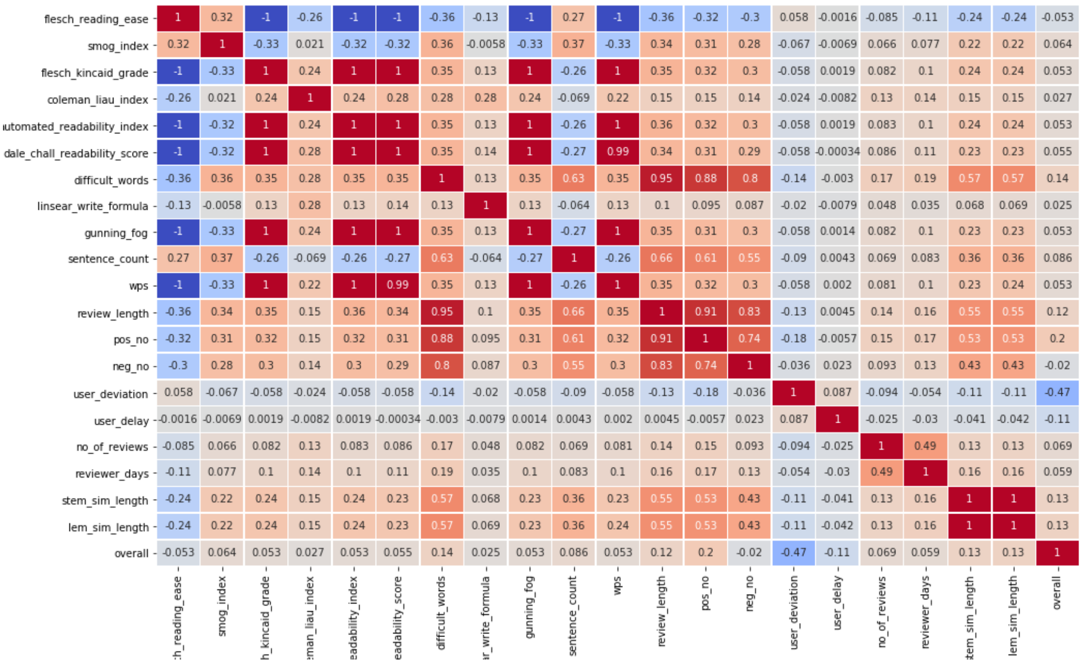

Figure 7.7 Heatmap of Inter-Feature Correlation

This section intends to analyze the relationship between the features themselves which are used to predict helpfulness. The heatmap depicting the correlation among all the features is shown in figure 7.7. Darker the shade of the cell, more is the correlation among the corresponding variables. A threshold value of 0.8 is considered and all the feature-pairs, absolute value of whose correlation is above this threshold are considered highly correlated. All the feature-pairs falling into this highly correlated category is shown in table 7.5. It can be observed that the highest correlation is among flesch kincaid grade and automated readability index which is 0.9999. The correlations with flesch Kincaid grade with certain readability measures is highest followed by other readability features, text features and meta data features.

Table 7.5 Top Highly Correlated Features

Sno	**Feature_1**	**Feature_2**	**Correlation**
1	flesch_reading_ease	flesch_kincaid_grade	-0.99947
2	flesch_reading_ease	automated_readability_index	-0.9993
3	flesch_reading_ease	dale_chall_readability_score	-0.99683
4	flesch_reading_ease	gunning_fog	-0.99933
5	flesch_reading_ease	Wps	-0.99868
6	flesch_kincaid_grade	automated_readability_index	0.999913
7	flesch_kincaid_grade	dale_chall_readability_score	0.995617
8	flesch_kincaid_grade	gunning_fog	0.999815
9	flesch_kincaid_grade	wps	0.999805
10	automated_readability_index	dale_chall_readability_score	0.99559
11	automated_readability_index	gunning_fog	0.99978
12	automated_readability_index	wps	0.999766
13	dale_chall_readability_score	gunning_fog	0.996795
14	dale_chall_readability_score	wps	0.994415
15	difficult_words	review_length	0.95466
16	difficult_words	pos_no	0.884
17	difficult_words	neg_no	0.804404
18	gunning_fog	wps	0.99958
19	review_length	pos_no	0.907989
20	review_length	neg_no	0.827676
21	stem_sim_length	lem_sim_length	0.995196

7.6 Feature-set Size Reduction using Inter-feature Correlation Graph

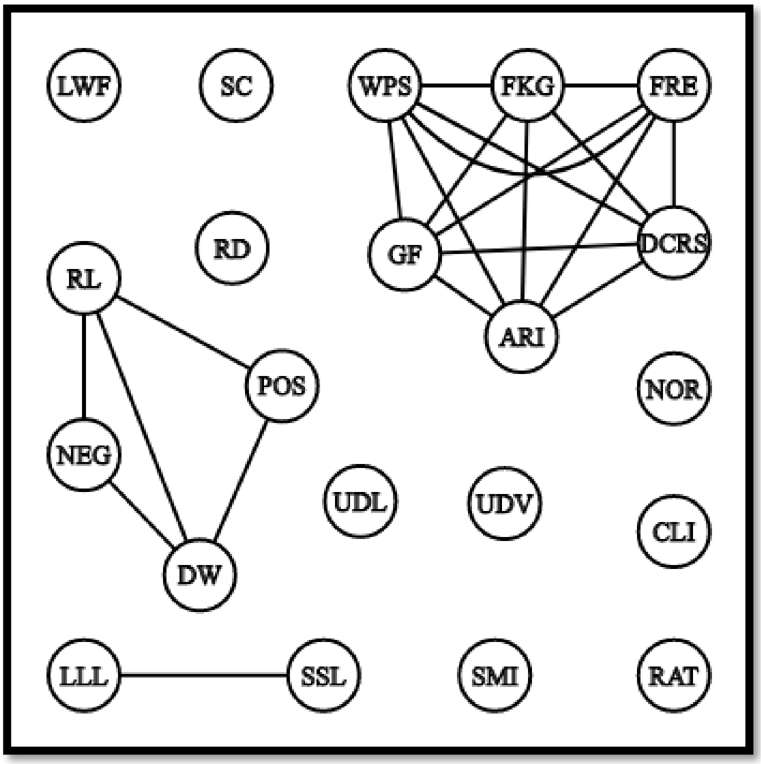

Figure 7.8 Inter-feature Correlation Graph

Based on the highly correlated features data, an inter-feature correlation graph is created as shown in figure 7.8. A total of 21 vertices corresponding to 21 features are established along with establishing edges between these vertices based on high-correlation. Here three connected components can be observed from the graph. Features from each connected component are grouped. These features are ordered based on their individual performance in predicting helpfulness using Extreme Gradient Boosting. The feature giving the least error is chosen from the group and finally the residual features are used for helpfulness prediction. The metric chosen here is RMSE. These features can be grouped into three groups. The first group consists of words per sentence, dale chall readability score, gunning fog, flesch Kincaid grade, automated readability index and flesch reading ease. This is shown in table 7.6. The feature with the least error is words per sentence. The second group consists of number of positive words, difficult words, review length and number of negative words. This is shown in table 7.7. The feature with the least error in this group is number of positive words. The third group consists of lem sim length and stem sim length. This is shown in table 7.8. The feature with the least error is lem sim length.

Table 7.6 Features under Group 1

Feature	rmse-XGB
wps	0.2330
dale_chall_readability_score	0.2336
gunning_fog	0.2338
flesch_kincaid_grade	0.2338
automated_readability_index	0.2339
flesch_reading_ease	0.2341

Table 7.7 Features under Group 2

Feature	rmse-XGB
pos_no	0.2221
difficult_words	0.2239
review_length	0.2241
neg_no	0.2347

Table 7.8 Features under Group 3

Feature	rmse-XGB
lem_sim_length	0.2319
stem_sim_length	0.2319

Taking only the features with the least errors from the respective groups and discarding rest of the features from those groups a revised feature-set is made. Now, the size of the revised feature-set becomes 13 with the features as shown in figure 7.9.

Reviewer	Review Text	Review Metadata
user_deviation user_delay no_of_reviews reviewer_days	smog_index wps coleman_liau_index linsear_write_formula sentence_count pos_no neg_no	rating lem sim length

Figure 7.9 Revised Feature-set

Using the revised feature-set, the helpfulness prediction task is conducted again. Since, Extreme Gradient Boosting was consistently performing well, this is used as the model. Along with that, customized Grid Search is used for adjusting the arguments for futher better performance. It is observed that an RMSE of 0.1924 and MAE of 0.1323 is achieved with this set-up.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusions

The purpose of this work is to develop a system which automatically predicts helpfulness of product reviews along with analyzing the factors that influence the helpfulness of reviews. Here, the features are divided into three categories reviewer, review text and review meta data and their predicting capabilities are evaluated individually and combined. The machine learning models Linear Regression, Random Forests and Extreme Gradient Boosting were utilized for predicting helpfulness. Customized Grid Search is developed for adjusting the arguments Subsample size, Learning Rate and No of Estimators in Extreme Gradient Boosting in order to extract further better results in all-feature helpfulness prediction.

The rating of a review shows immense impact on review helpfulness. This is followed by features such as Number of Positive Words, Number of Difficult Words, Lemmatized Description Similar Words, Stemmed Description Similar Words. Review meta data showed the best performance out of the three categories. Also, in most of the cases Extreme Gradient Boosting showed the best performance. Readability features showed high inter-feature correlation among themselves. This is followed by features such as Number of Difficult Words, Number of Positive Words and Length of Review. But Positive Word Number and Difficult Word Number showed better performance than Review Length individually in predicting helpfulness. It can be said that they are a better version of review length consisting only important segments of a review in the context of helpfulness prediction.

Furthermore, an Inter-feature Correlation Graph was developed based on the inter-feature correlation analysis done. This graph is utilized for feature-set size reduction which in turn leads to optimization without giving much effect on the error in prediction. Component-wise grouping from the graph leads to feature-set reduction.

The Reduced Feature-set obtained an Root Mean Squared Error (RMSE) of 0.1924 and Mean Absolute Error (MAE) of 0.1323. The reduced feature-set contained 13 features as compared to 21 of total feature-set. There is approximately a 40% reduction in feature-set size without significant increase in error through this technique.

The best performance of the system is achieved when all the 21 features are used together and Extreme Gradient Boosting is used along with Customized Grid Search as the machine learning algorithm to predict helpfulness score. The system achieved a Root Mean Squared Error (RMSE) of 0.1921, Mean Absolute Error (MAE) of 0.1319 in predicting helpfulness score of reviews.

8.2 Future Work

This work can be extended and enhanced in several possible ways. Firstly, further features can be explored in all three categories of reviewer, review text and review meta data and their influence in predicting helpfulness can be analyzed. Helpfulness received by previous reviews of the reviewer can be used to predict helpfulness of other reviews. Deep Learning can be used which can reduce the workload of feature extraction but giving an accurate result. Also, a weightage can be given to recent reviews since they may not initially attract large number of votes. This work focused on Electronics data. Similar analysis can be done on data of other domains. Reviews obtaining less than ten votes were filtered-out here. These reviews can also be considered for analysis. Further, this can also be taken as a classification problem or as a combination of classification and regression in order to get better results. The current problem can be considered as a Big Data problem since the number of reviews involved is large and a lot of processing is involved. Hence utilization of distributed and parallel computing can be explored for faster and efficient performance.

REFERENCES

[1] Malik, M. S. I. and Ayyaz Hussain. “An analysis of review content and reviewer variables that contribute to review helpfulness.” *Inf. Process. Manage.* 54 (2018): 88-104.

[2] Saumya, Sunil, Jyoti Prakash Singh, Abdullah M. Baabdullah, Nripendra P. Rana and Yogesh Kumar Dwivedi. “Ranking online consumer reviews.” Electronic Commerce Research and Applications 29 (2018): 78-89.

[3] Park, Yoon-Joo “Predicting the Helpfulness of Online Customer Reviews across Different Product Types”, Sustainability, (2018), 10. 1735.

[4] Y.K. Chua, Alton & Banerjee, Snehasish “Helpfulness of user-generated reviews as a function of review sentiment, product type and information quality”, Computers in Human Behavior, (2016) ,54. 547-554.

[5] Y. Zhang and D. Zhang, "Automatically predicting the helpfulness of online reviews," Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), Redwood City, CA, 2014, pp. 662-668.

[6] Mudambi, S. M., & Schuff, D. “What makes a helpful review? A study of customer reviews on Amazon. com.”, MIS Quarterly, 2010, 34(1), 185–200.

[7] Y. Liu, X. Huang, A. An and X. Yu, "HelpMeter: A Nonlinear Model for Predicting the Helpfulness of Online Reviews," 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, NSW, 2008, pp. 793-796.

[8] R. He, J. McAuley. Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016

[9] J. McAuley, C. Targett, J. Shi, A. van den Hengel. Image-based recommendations on styles and substitutes. SIGIR, 2015

[10] Merton, R.K., The Matthew effect in science: the reward and communication systems of science are considered. Science 159 (3810), 1968, 56–63.

[11] Kim, S.M.; Pantel, P.; Chklovski, T.; Pennacchiotti, M. Automatically assessing review helpfulness.In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Sydney, Australia, 22–23 July 2006; pp. 423–430.

[12] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" 2014.

[13] X. Li, L. Xie, F. Zhang and H. Wang, "Online Deceptive Product Review Detection Leveraging Word Embedding," 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology *Congress(DASC/PiCom/DataCom/CyberSciTech)*, Orlando, FL, 2017, pp. 867-870.

[14] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Seattle, Washington, USA, Aug 22-25, 2004.

[15] Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web conference (WWW-2005), Chiba, Japan, May 10-14, 2005.

[16] Pedregosa F. , Varoquaux G. , Gramfort A. , Michel V., Thirion B. , Grisel O. , Blondel M. , Prettenhofer P. , Weiss R. , Dubourg V. , V,erplas J. , Passos A.

, Cournapeau D. , Brucher M. , Perrot M. , Duchesnay E.,"Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, Volume 12, 2825-2830, 2011

[17] Wikipedia contributors. SMOG. In Wikipedia, The Free Encyclopedia. Retrieved 07:55, December 12, 2018, from https://en.wikipedia.org/w/index.php?title=SMOG&oldid=838363055, (April 26, 2018)

[18] Wikipedia contributors, Flesch–Kincaid readability tests. In Wikipedia, The Free Encyclopedia. Retrieved 07:58, December 12, 2018, from https://en.wikipedia.org/w/index.php?title=Flesch%E2%80%93Kincaid_readability_tests&oldid=872588092,(December 7, 2018).

[19] Wikipedia contributors, Coleman–Liau index. In Wikipedia, The Free Encyclopedia. Retrieved 07:59, December 12, 2018, from https://en.wikipedia.org/w/index.php?title=Coleman%E2%80%93Liau_index&oldid=758752161, (January 7, 2018).

[20] Wikipedia contributors, Gunning fog index. In Wikipedia, The Free Encyclopedia. Retrieved 08:00, December 12, 2018, from https://en.wikipedia.org/w/index.php?title=Gunning_fog_index&oldid=840475093, (May 10, 2018).

[21] Wikipedia contributors, Dale–Chall readability formula. In Wikipedia, The Free Encyclopedia. Retrieved 08:01, December 12, 2018, from https://en.wikipedia.org/w/index.php?title=Dale%E2%80%93Chall_readability_formula&oldid=872564618, (December 7, 2018).

[22] Wikipedia contributors, Automated readability index. In Wikipedia, The Free Encyclopedia. Retrieved 08:12, December 12, 2018, from https://en.wikipedia.org/w/index.php?title=Automated_readability_index&oldid=856199533,(August 23, 2018).

[23] Eltorai, Adam & S Naqvi, Syed & Ghanian, Soha & Eberson, Craig & Weiss, Arnold Peter & Born, Christopher & Daniels, Alan, "Readability of Invasive Procedure Consent Forms", Clinical and translational science,2015.

[24] Pandey S., M S., Shrivastava A., “Data Classification Using Machine Learning Approach”, In: Thampi S., Mitra S., Mukhopadhyay J., Li KC., James A., Berretti S. (eds) Intelligent Systems Technologies and Applications. ISTA 2017. Advances in Intelligent Systems and Computing, vol 683. Springer, Cham, 2018.

[25] V. Vishagini and A. K. Rajan, "An Improved Spam Detection Method with Weighted Support Vector Machine," 2018 International Conference on Data Science and Engineering (ICDSE), Kochi, pp. 1-5, 2018.