IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11356

Amazon Product Recommendation System

Md Zaid Ahmed[1], Abhay Singh[2], Abir Paul[3], Sayantani Ghosh[4], Avijit Kumar Chaudhuri[5]

[1,2,3]Undergraduate, Department of Computer Science and Engineering, Techno Engineering College,

Banipur, West Bengal, India

[4,5]Professor, Department of Computer Science and Engineering, Techno Engineering College,

Banipur, West Bengal, India

Abstract: This paper offers a detailed explanation of a system that uses sentiment analysis and machine learning algorithms to classify and recommend products on Amazon. Using the idea of Machine Learning, we developed a system that can be used by many e-commerce sites for better product recommendations. This system employs a machine learning model in which similar and superior products are offered to the customers in order of best to worst based on the product utilized in the past. The computer will compile a shortlist of all relevant items or products based on user-generated product reviews that meet the user's criteria, taking into account the product's quality and rating. The approach we employed was to create a system model that would analyze customer reviews for various products in the same category and then use Natural Language Processing to arrive at a conclusion where the system (model) would be able to assess whether the review is positive or negative. We've also used the ratings offered on various items to create a technique to combine ratings and reviews to improve the accuracy of the system (model). We employed the Collaborative Algorithm to improve the accuracy of product recommendations. During the creation of the system, we used the Amazon e-commerce site and its products to simulate a real-world implementation scenario (model). Our system uses cosine similarity to find the similarities between items on basis of the multiple user's ratings and form a matrix which helps to recommend items to other users.

Keywords: Review & Ratings, Machine Learning, Natural Language Processing, Collaborative Algorithm, Recommendation System, Accuracy.

1. INTRODUCTION

With the advancement of technology and day-by-day innovation, lives of people have become much more comfortable. Shopping on the internet has become an essential part of people's life. At present time people prefer to purchase products online because they have quick access to a wide range of different types of items on one platform. But they spent a lot of time on online commercial websites evaluating the quality of the products and reading the product reviews to get an insight into the particular product. Finding items that suits user's choice is very hectic choice as it takes lots of time and effort to manually go through all the products one by one.

In a digital environment, customers are also unable to physically touch or test products quality, and as a result, people rely on other customer's feedback to acquire knowledge about the product they are thinking about buying. However, people are having a difficult time finding their desired review due to unnecessary feedback and remarks.

As a result, there is a need for a system that can assist customer in quickly finding the relevant review, filtering thousands of comments, and helping users save their time and efforts by recommending them appropriate products.

The solution to above mentioned challenge can be provided using different machine learning algorithms and methods. People want to obtain valuable reviews as quickly as possible while viewing a product. As a result, algorithms that can predict the user rating based on the text review are needed. Getting insight from a textual evaluation as a whole could improve the consumers' experience. It can also help companies in growing their sales and improving their products by gaining a better understanding of their customers' demands.

The goal is to create a system that uses content-based filtering to predict user rating, review relevance, and recommend the precisely similar thing to users of their choices. This further helps the user to gain access to things quickly and efficiently and it saves a lot of their time.

1.2 LITERARY REVIEW

1) In [1], the authors have discussed a system that will help users to find some specific movie of their choice. The idea behind the implementation of the system that is discussed in this research paper is that a user will be recommended a movie, that will suit their interests, out of millions of movies present on online platforms nowadays. They have also

tried to explain the limitations that different such preexisting models possess together with algorithms based on which those models were built like Matrix Decomposition, Clustering, and Deep Learning Approach.

Finally, the authors have mentioned the techniques like Collaborative Content-based techniques and algorithms like the KNN algorithm that they have utilized to build the system. Content-based filtering is a technique that uses the previous log of the respective user to recommend something. While KNN is an algorithm that falls under the Supervised Machine Learning algorithm and can be used for both classification and regression.

2) In [2], it is examined as to how an information search system can be built which is quick-witted. Also, this paper focus on integrating the recommendation system feature with the intelligent information system to provide the user with all the search details and also recommends other useful information based on their historic search data. The paper at first explains all the aspects of building a search engine like maintaining a repository to store appropriate data, match and find the relevant data, and auto-correct misspelled words during the search. Then it describes methods and approaches needed to build a system that will recommend data related to the current search or it will recommend information using the user's previous preferences. To implement all the above-mentioned ideas, a proper algorithm is needed to be followed so that the searched result should be shown quickly and correctly together with relevant recommendations. For this reason, the authors have tried to provide a comparison between three widely use algorithms like Collaborative Filtering, Content-based Filtering, and Hybrid Filtering. Thewhole idea behind this research paper is to discuss in detail the best techniques and methods that can be followed to implement an intelligent search information system integrated with recommendation system.

3) In [3], the authors have described a system that they have proposed as a recommendation system, named APriori. It is a pervasive product recommendation system motivating people (specifical consumers) to actively share experiences about products at the point of use. The authors have also tried to explain the shortcomings between different approaches that were used to provide or get product recommendations. Mainly, product ratings and product reviews were used as the logic behind different approaches to build an effective product recommendation system. But these approaches seem not to work for mobile users because product reviews are not suitable for recommendation systems used on mobile phones, and also due to hardware restrictions, it is still difficult for users to enter longer texts on their mobile or even just read longer text on the small display of their mobile phone. Therefore, the authors proposed an innovative approach to help mobile users actively participate in the recommendation of products, which is Dynamic Rating Criteria.

The main objective behind this paper was to develop a system that will help people using mobile to get a better recommendation about a specific product and also motivate them to share their experiences about the products they have used without worrying about the device restrictions.

4) In [4], the authors have tried to give a detailed explanation of various methods, algorithms, and techniques used for recommending products to users. The paper also describes the shortcomings of all those methods and algorithms that were used to date in the product recommendation system. The authors have also mentioned the need for a recommendation system in this era because there can be found millions of information related to different products but going through each product will take a long time. Therefore, in such a situation, a recommendation system can prove to be handy by showing users the relevant product that they might like by utilizing some specific approaches. Content-based recommendation, Collaborative filtering, and Hybrid filtering are the three categories under which most of the recommendation systems fall. Limitation related to these categories of recommendation system is also discussed in this paper which are limited content analysis, cold-start problem, data-sparsity problem, scalability problem, and privacy issues. Then in the end the authors have provided analysis of different metrics needed during the development of recommendation systems.

1.3 STEPS OF OUR SYSTEM

1.3.1 Data Collection

The first step involved in developing any machine learning model, is to collect all the relevant data for training and testing purpose.
In our system, the musical instruments data set was compiled from Amazon reviews and product information. This data-set contains product details like descriptions, category information, price, brand, and image features, as well as reviews and ratings, like text, and useful votes.

1.3.1.1 Details about product's reviews data set

This data set contains product reviews for musical instruments, including ratings, text, and useful votes. The data was originally in JSON format. To convert JSON to CSV format, the JSON was imported and decoded.
Different fields of data-set are described as follows:

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)

1.3.1.2 Product Metadata

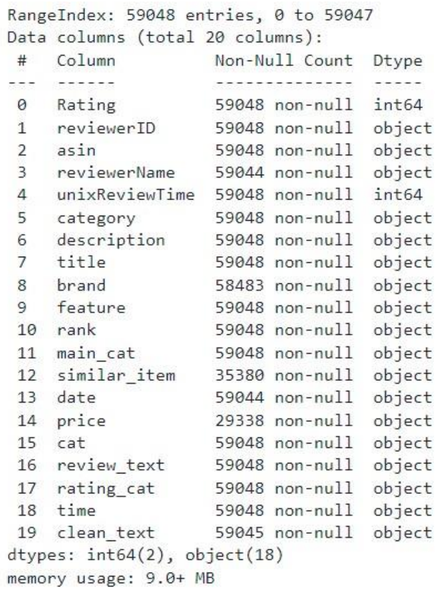
Descriptions, category information, price, brand, and picture features are all included in this collection for musical instruments. The JSON was imported and decoded to convert it to CSV format. Description of the data is given below:

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

1.3.2 Data Wrangling

1.3.2.1 Building Data frames

In JSON files, product reviews and meta information were saved in distinct data frames. The left join was used to combine two data frames, and "asin" was preserved as the common merger. The following is a description of the final merged data frame:



```
RangeIndex: 59048 entries, 0 to 59047
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Rating          59048 non-null  int64
 1   reviewerID      59048 non-null  object
 2   asin            59048 non-null  object
 3   reviewerName    59044 non-null  object
 4   unixReviewTime  59048 non-null  int64
 5   category        59048 non-null  object
 6   description     59048 non-null  object
 7   title           59048 non-null  object
 8   brand           58483 non-null  object
 9   feature         59048 non-null  object
 10  rank            59048 non-null  object
 11  main_cat        59048 non-null  object
 12  similar_item    35380 non-null  object
 13  date            59044 non-null  object
 14  price           29338 non-null  object
 15  cat             59048 non-null  object
 16  review_text     59048 non-null  object
 17  rating_cat      59048 non-null  object
 18  time            59048 non-null  object
 19  clean_text      59045 non-null  object
dtypes: int64(2), object(18)
memory usage: 9.0+ MB
```

Figure 1. Meta data of joined data frames

DOI: 10.17148/IJARCCE.2022.11356

The combined data has 1731703 rows and 30 columns. After sorting the data and reducing the product category to 'Musical Instruments,' we were able to reduce the data set to 1644354 rows. By limiting the data to guitar-related goods, we were able to reduce it to 59048 rows and 19 columns. We also removed the size and null values from the data set to decrease the amount of time it takes to run models.

1.3.2.2 Data Refining

The data-set is filtered by following some simple methods to reduce noise from the data and make it more efficient. This includes the following:

All the values that are found empty are replaced by null.
Reviews are then filtered based upon good and bad ratings, further above 3 was considered as good and below 3 was considered as a bad rating.
All the duplicate items are dropped out from the data-set
Columns were renamed for more clarity
Some of the missing values were also have been removed from the data-set.
HTML tags are removed as it doesn't add sense to the text making the data more understandable and easier to read.
Special characters are then removed. These special characters can be punctuation marks or any symbol or emojis. As this information is not required during Natural Language Processing (NLP).
The process of lemmatization is carried out. It is a process to remove word affixes together in a base form, which will act as a root word or lemma.

1.3.3 Exploratory Data Analysis

With the use of summary statistics and graphical representations, exploratory data analysis refers to the crucial process of doing first investigations on data to uncover patterns, spot anomalies, test hypotheses, and check assumptions.

Exploratory studies were conducted using our data-set to uncover the following findings.

(i) Rating prediction based on reviews
(ii) Rating vs number of reviews
(iii) The percentage of reviews vs. the rating
(iv) The top 20 goods with the most reviews
(v) The bottom 20 goods that have been reviewed
(vi) Positive and negative words
(vii) World cloud for different ratings, brand name, etc

The following observations aided us in answering the following crucial question, which will aid us in better feature engineering.

What changes occur to the most positively and negatively reviewed product?

From 2010 to 2018, the most popular product is 'Segll Enrge Rsic Gir ', which had an overall rating of more than 4. 'Sr 6 String Gir Right-Handed Ble Full (MG50-BL),' on the other hand, has an overall rating of less than 2 except in 2012, 2013 and 2014.

Which rating received the most reviews?

Customers have submitted a total of 59048 ratings and reviews ranging from 1 to 5 stars, as well as written reviews, from 2010 to 2018. When compared to other ratings, the number of reviews for rating 5 is significantly larger. The majority of customers were pleased with their purchases. Only roughly 12% of the reviews have a rating of less than 3.

All ratings less than 3 are considered 'negative,' while those more than 3 are considered'positive.' More than 50000 reviews are positive, according to the data.

DOI: 10.17148/IJARCCE.2022.11356

What is the year with the most reviews?

From 2000 to 2010, the number of reviews was low. Following that, there is a constant increase of reviews until 2015, when they begin to decline. The year with the most reviews is 2015.

What is the year with the most customers?

The highest number of unique customers is in 2015 according to the data. More than 10,000 unique customers purchased items from Amazon.

Which Year has the Highest Number of Product?

During the years 2000–2010, the number of unique items was low. In 2015, around 2500 unique products were sold. Except for 2003, the favorable rating percentage has increased to above 80% from 2000 to 2018. In 2003, there were 75 percent positive reviews, which was a rather low number. The greatest percentage was more than 90% between 2000 and 2002. Overall, the score ranged from 82 percent to 90 percent**.**

Which of the review length bins has the most positive rating?

The largest percentage of positive ratings (94%) is found in the 0-1000 word range, while the lowest (83%) is found in the 1100-1200 word range. As the length of the review grows longer, the positive ratings tend to grow. Customers that provide a detailed review are more likely to give a positive review.

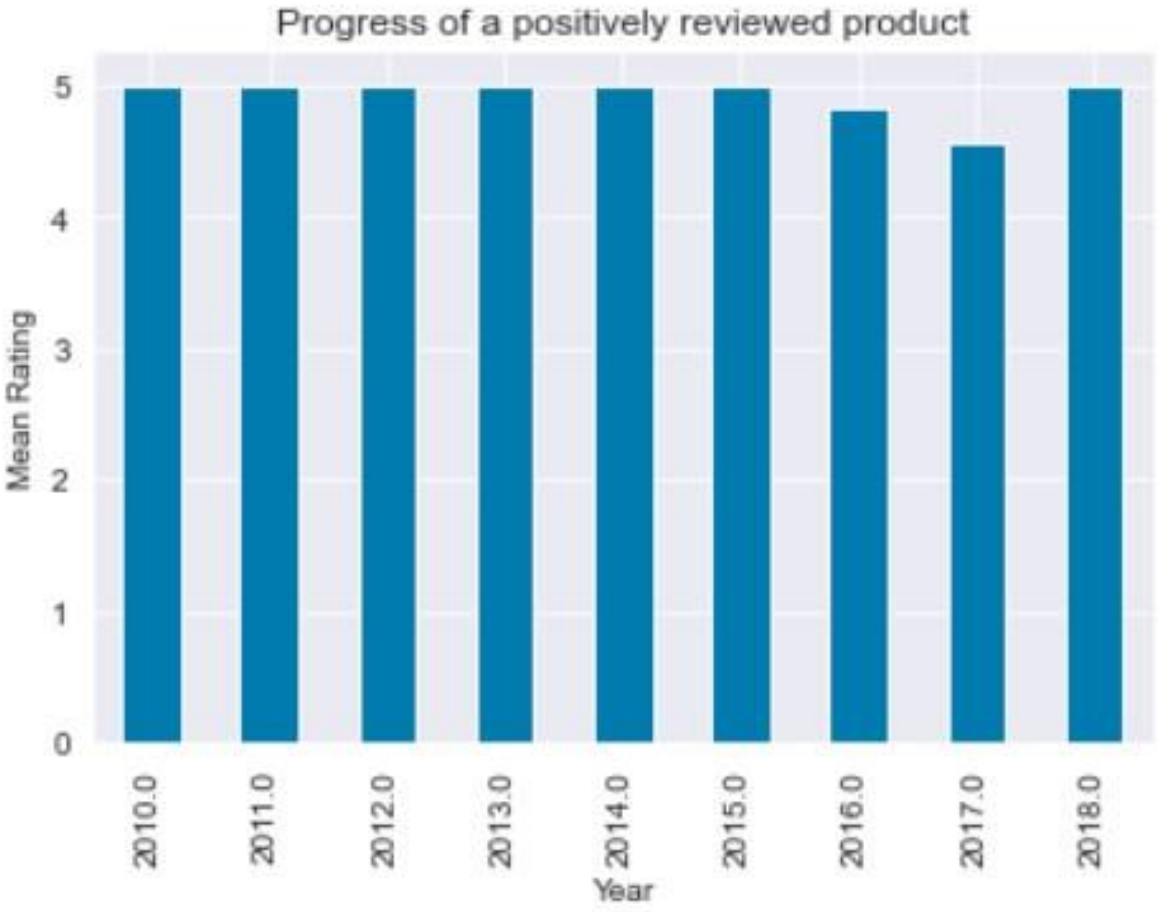


Figure 2. Mean Rating vs year of most positively reviewed product

DOI: 10.17148/IJARCCE.2022.11356

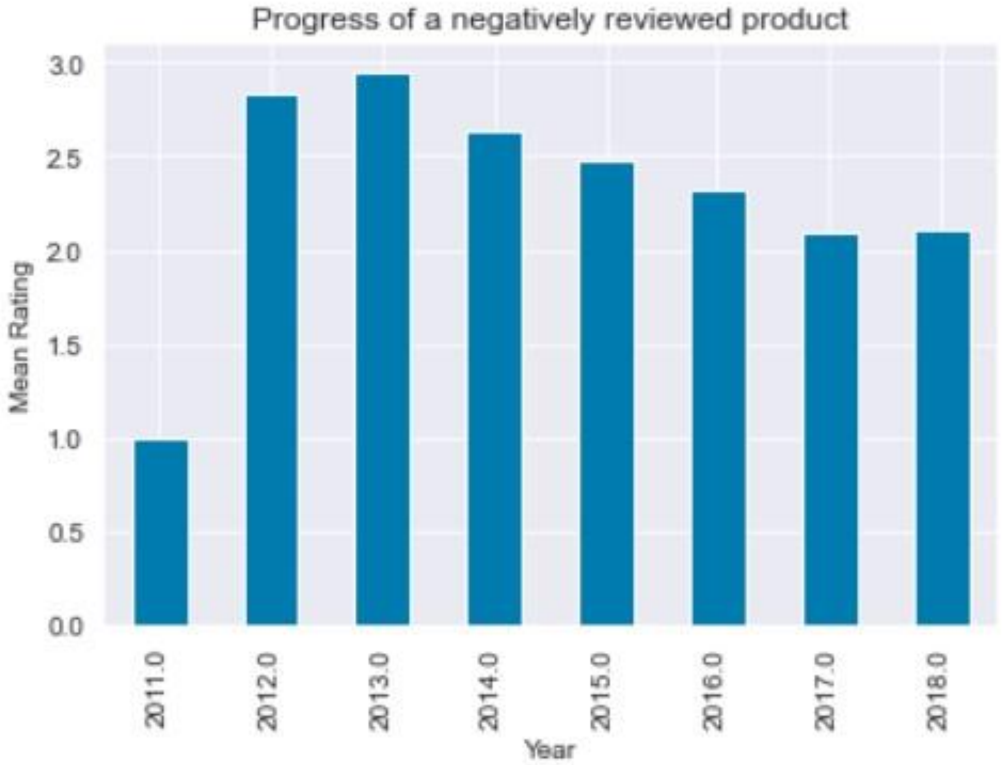


Figure 3. Mean Rating vs year of most negatively reviewed product

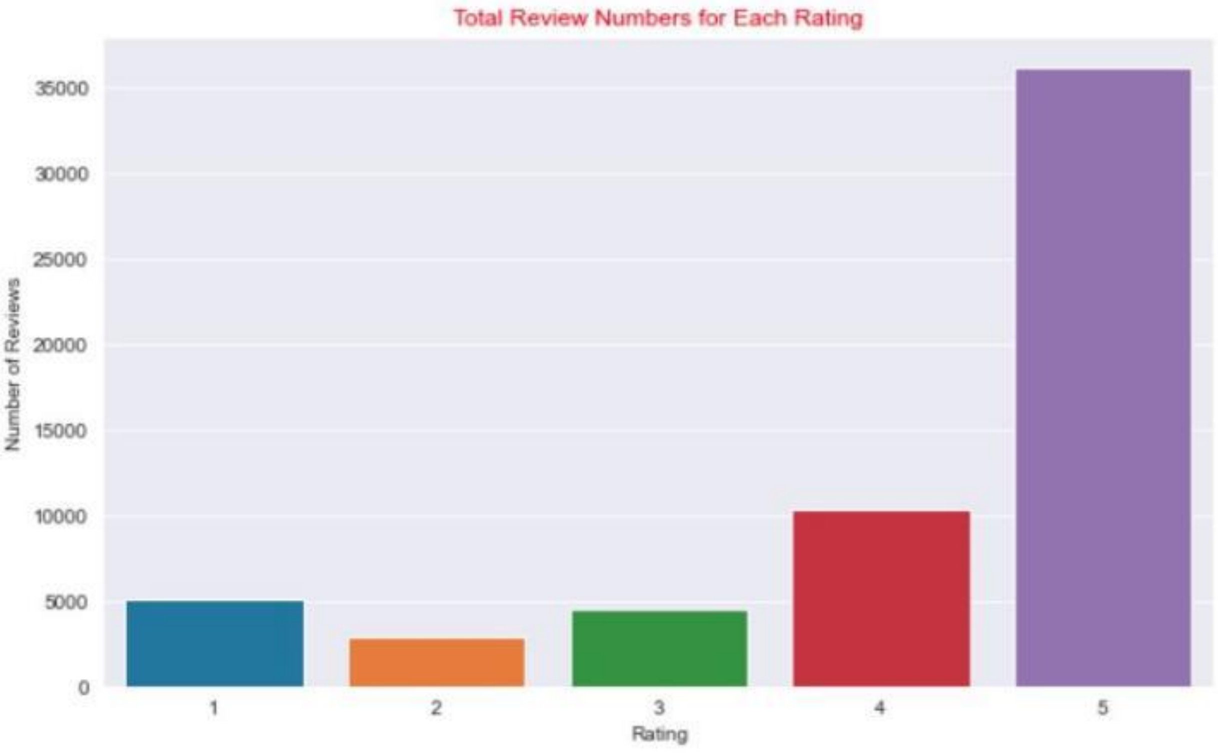


Figure 4. Number of reviews vs Rating count

DOI: 10.17148/IJARCCE.2022.11356

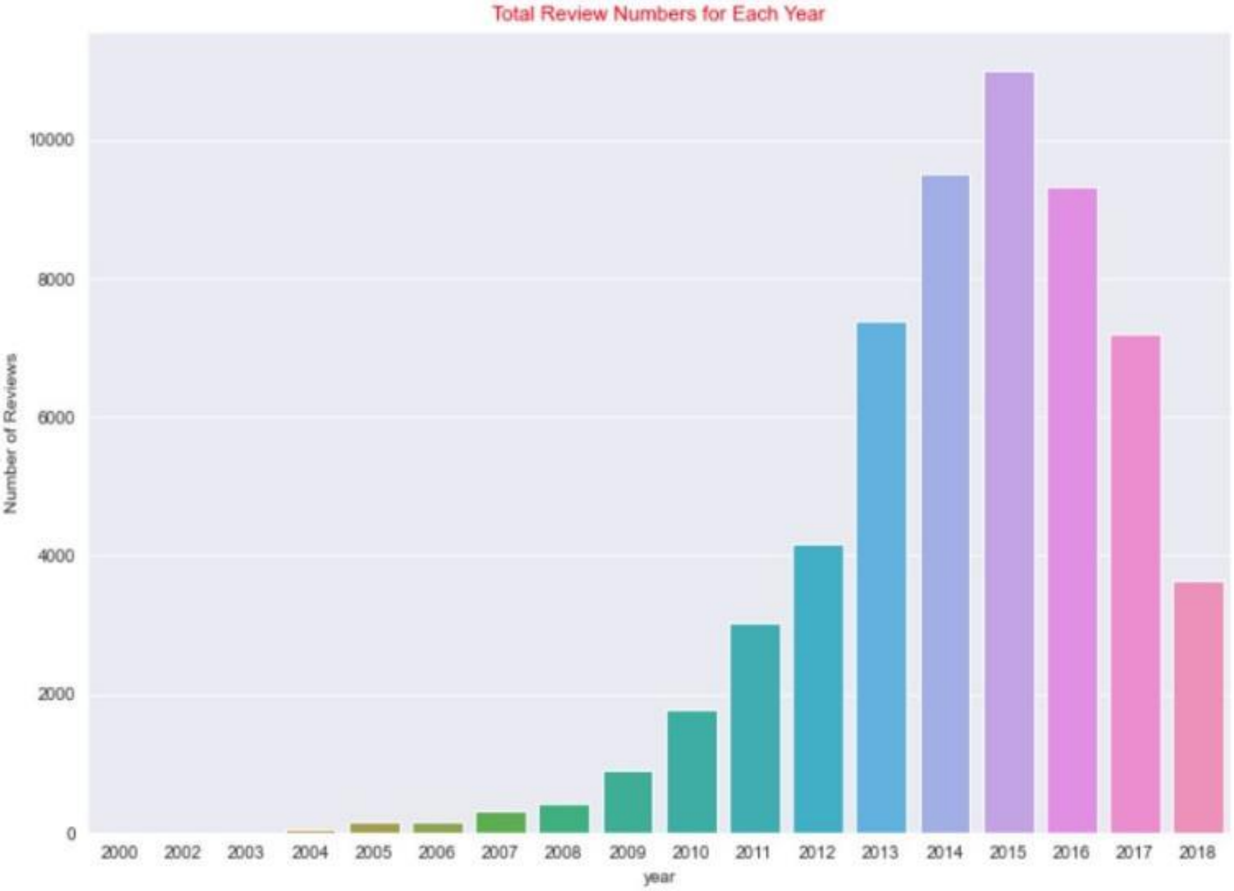


Figure 5. Number of reviews per year

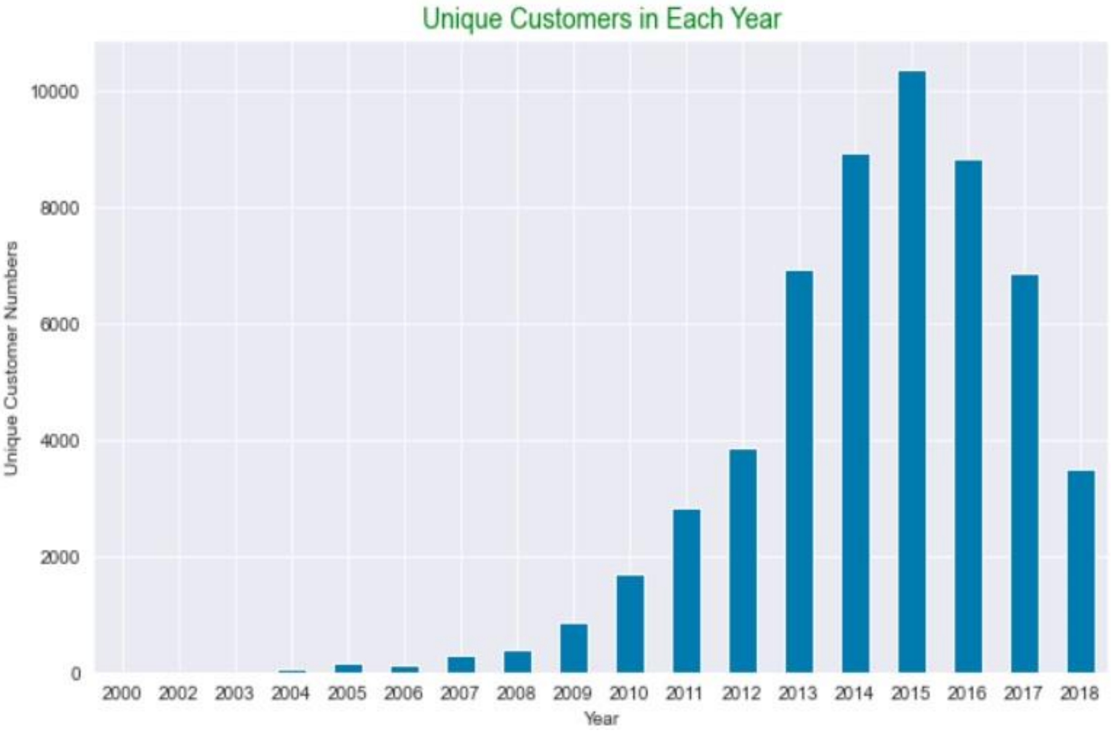


Figure 6. Unique customers each year

DOI: 10.17148/IJARCCE.2022.11356

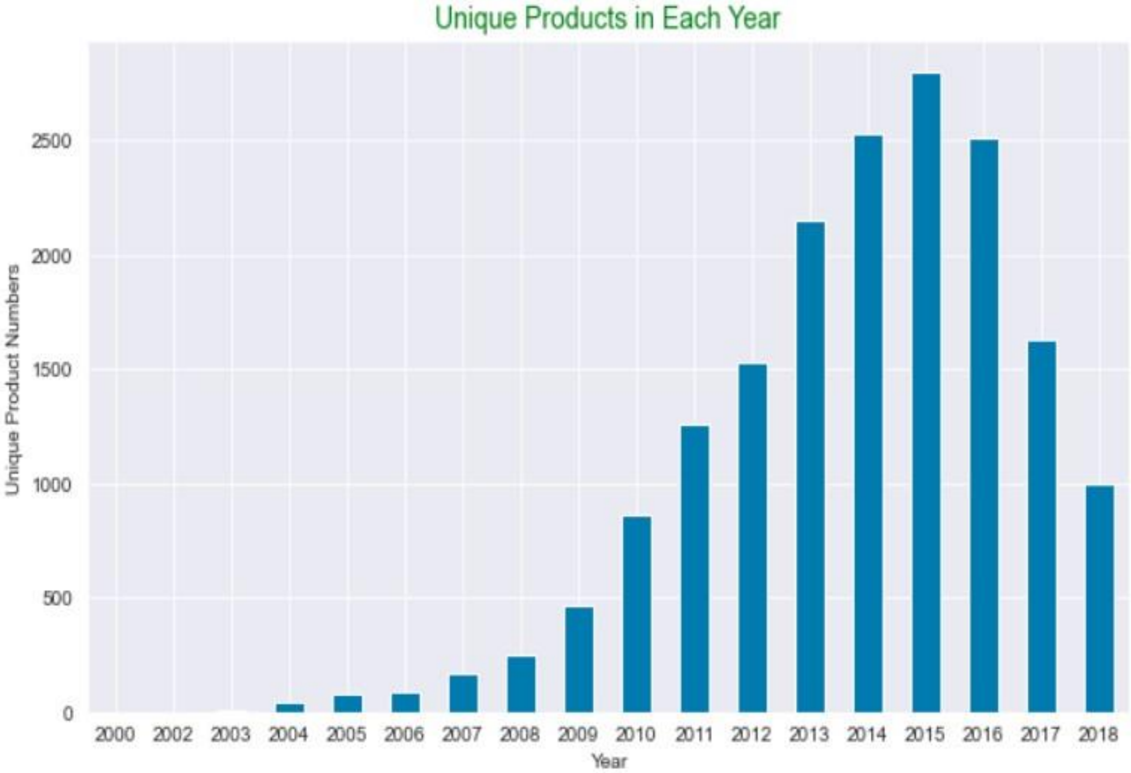


Figure 7. Unique products each year

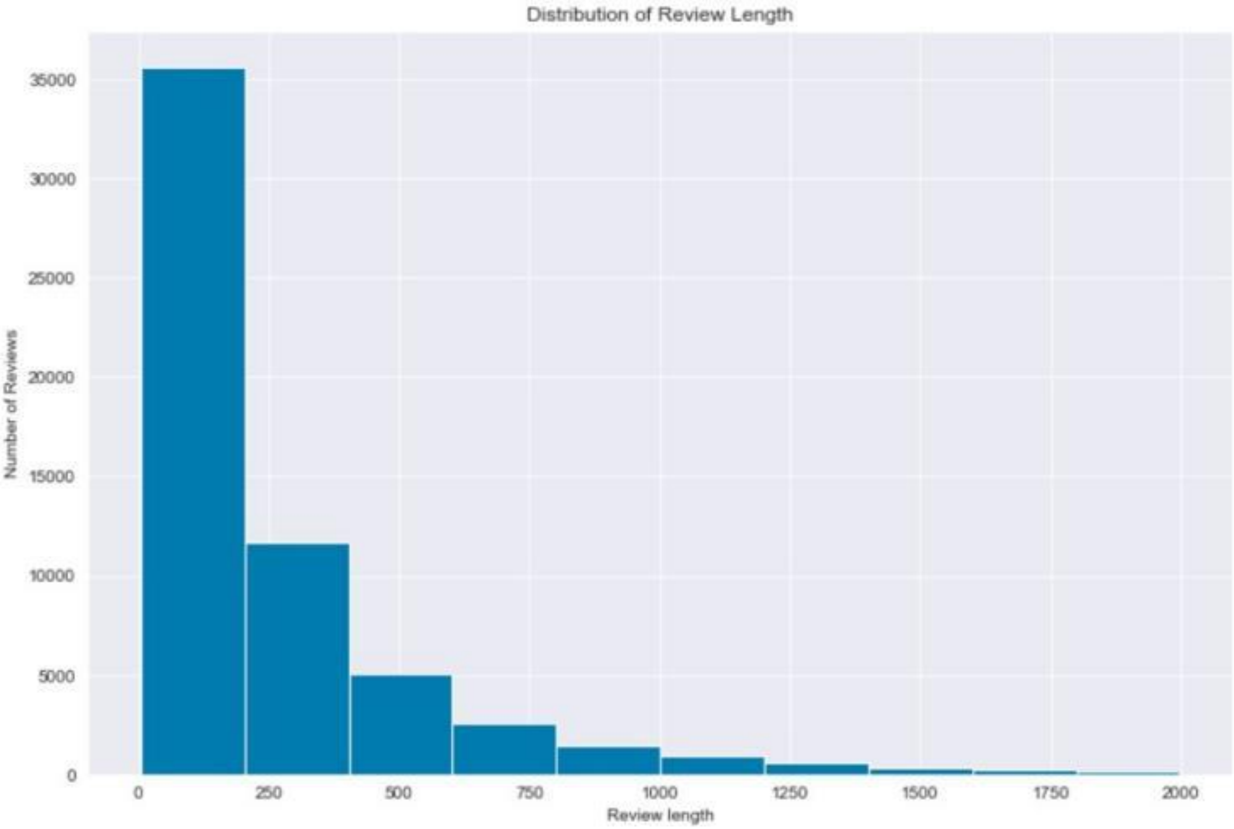


Figure 8. Number of reviews vs Review length

DOI: 10.17148/IJARCCE.2022.11356

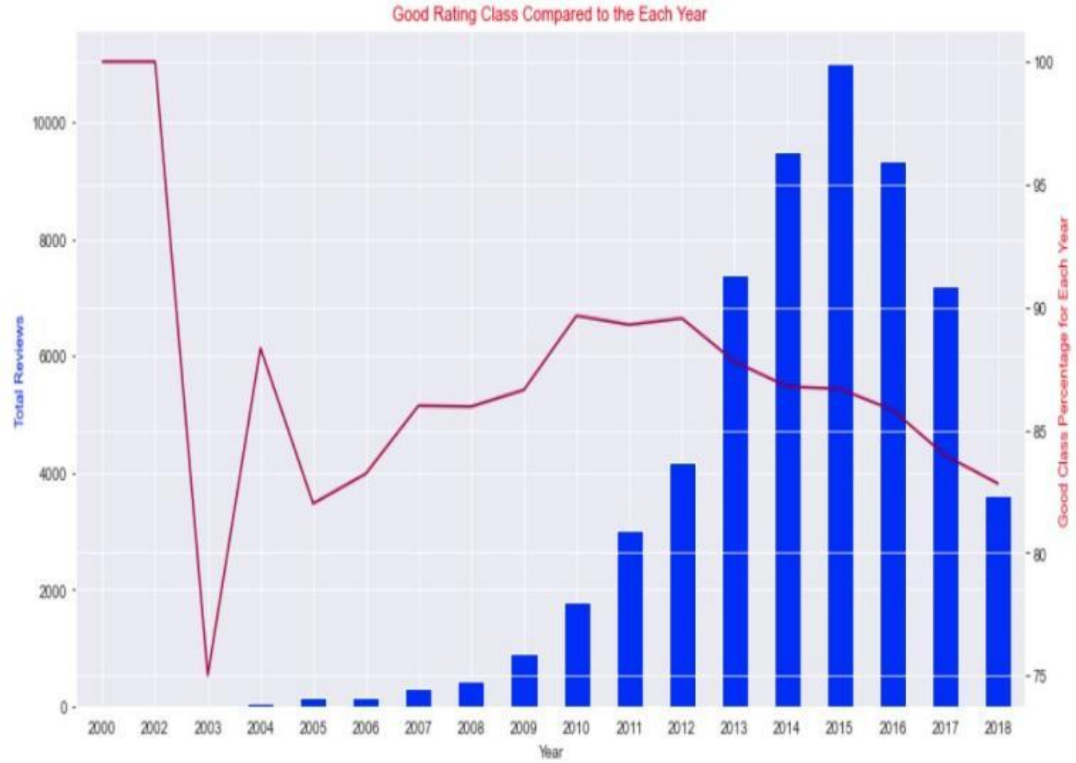


Figure 9. Positive rating compared in each year

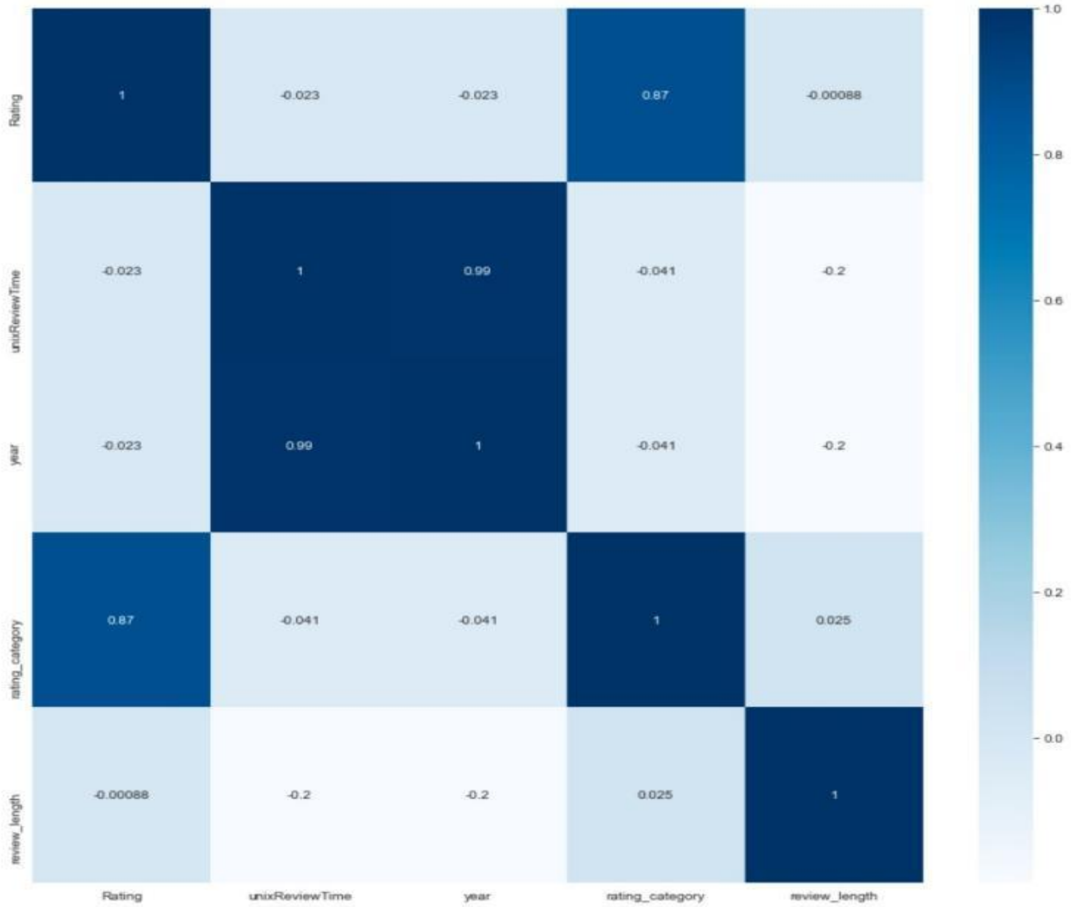


Figure 10. Correlation heat-map of all the features

DOI: 10.17148/IJARCCE.2022.11356

1.3.4 Sentiment Analysis

Sentiment analysis is a technique for determining whether data is good, negative, or neutral using natural language processing (NLP).

Interpreting consumer feedback through product reviews assists businesses in determining how pleased customers are with their goods/services and in recommending more appealing products.

1.3.4.1 Data Pre-processing

In Machine Learning, data pre-processing refers to the practice of preparing (cleaning and organizing) raw data to develop and train Machine Learning classifiers.

To conduct the sentiment analysis on the reviews, responsive variables and features are separated from the data-set. We further decreased the data-set size to 20771 rows by deleting all null values and removing positive reviews longer than 150 words from before 2010. 'clean text' and 'rating cat' are chosen from the data-set for X(feature) and Y (variable) respectively. The data-set was separated into two parts: 75 percent training and 25 percent test.

1.3.4.2 Feature engineering

Characteristics are generally quantitative and can be absolute numeric values or categorical features that can be encoded as binary features for each category in the list utilizing a one-hot encoding procedure. The act of choosing, editing, and converting raw data into features that can be utilized in supervised learning to make machine learning algorithms operate is known as feature engineering.

The threshold for word occurrence was calculated using min df/max df, PCA, and Singular Value Decomposition for feature selection. We implemented Count Vectorizer, TF-IDF, and Hashing Vectorizer to generate a collection of text documents into numerical feature vectors for feature engineering. We implemented five machine learning algorithms for each of these models to find the optimal combination. Let's discuss these models in detail.

Bag-of-Words Model

The Bag of Words is a Natural Language Processing approach to text modeling. The Bag of Words model is one of the most basic yet effective methods for extracting characteristics from text texts. A bag of words is a text representation that describes the frequency with which words appear in a document. We only keep track of word counts and don't pay attention to grammatical subtleties or word arrangement. With this model, XGBoost has the best accuracy of 0.933179.

TF-IDF

The TF-IDF (term frequency-inverse document frequency) statistic examines the relevance of a word to a document in a collection of documents. Because the TF-IDF weights words according to their significance, it may be used to discover which words are the most essential. This may be used to more quickly summarize articles or just identify keywords (or even tags) for a document. XGBoost has the best accuracy of 0.931831 with this model.

Hashing Vectorizer

The hashing vectorizer is a vectorizer that employs the hashing method to discover the token string name to feature integer index mapping. This vectorizer converts text documents into matrices by converting the collection of documents into a sparse matrix including the token occurrence counts. Again, XGBoost has the highest accuracy of 0.928943 out of all the algorithms.

Out of all the models, XGBoost with CountVectorizer Bag of Words(f1-score is 0.931442) or XGBoost with TF-IDF(f1-score is 0.930078) is the top models to predict the sentiment of the reviews.

1.3.5 Machine Learning Models

The algorithm must forecast sentiment based on reviews made by Amazon consumers who purchased guitar-related goods. This is a problem of supervised binary classification. This challenge was solved using Python's Scikit library. The machine learning methods listed below were implemented.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁒ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11356

Logistic Regression

Logistic Regression is a Machine Learning technique that is used for classification issues; it is a predictive analytic approach that is based on the probability notion. Because the nature of the target or dependent variable is dichotomous, there are only two viable classes.

Naive Bayes

Naive Bayes is a classification algorithm that may be used to solve binary (two-class) and multi-class classification issues. When stated using binary or categorical input values, the approach is simplest to grasp. It employed conditional probability to categorize future objects by giving class labels to instances/records.

Random Forest Classifier

Random forest is a Supervised Machine Learning Algorithm frequently utilized in Classification and Regression applications. The random forest classifier is a flexible classification method that provides aggregated predictions on multiple sub-samples of the data-set using some decision trees and averaging to increase predictive accuracy and control over-fitting.

XGBoost Classifier

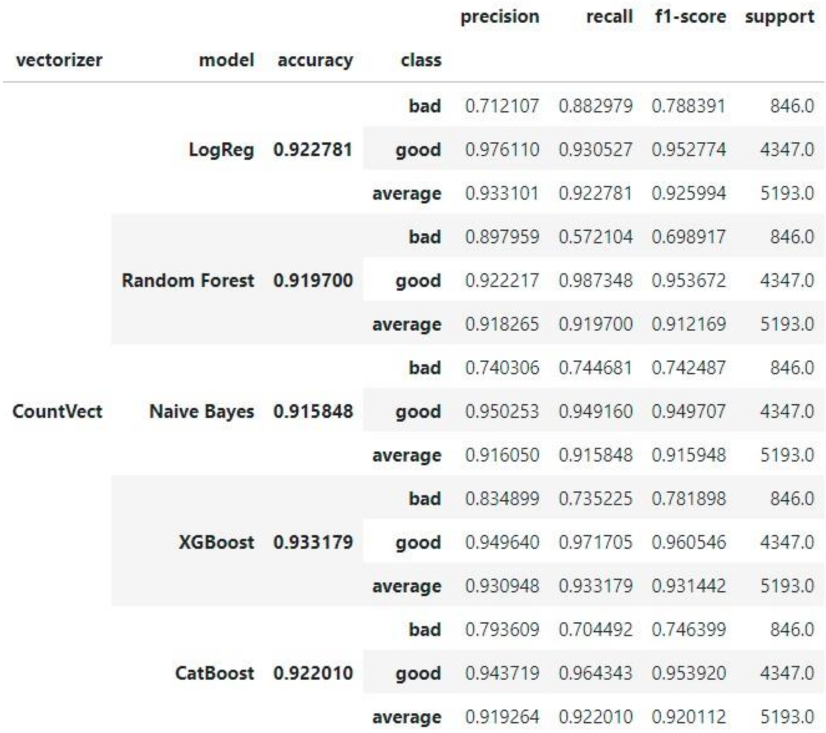
XGBoost is an ensemble Machine Learning technique based on decision trees that use a gradient boosting framework. XGBoost offers parallel tree boosting (also known as GBDT, GBM) to handle numerous data science issues quickly and accurately, as well as access to a range of model hyperparameters designed to provide you control over the model training process.

CatBoost Classifier

Yandex's CatBoost is an open-source enhanced decision tree machine learning technique. It operates in the same way as XGBoost, except it supports categorical variables out of the box and has a greater degree of accuracy.

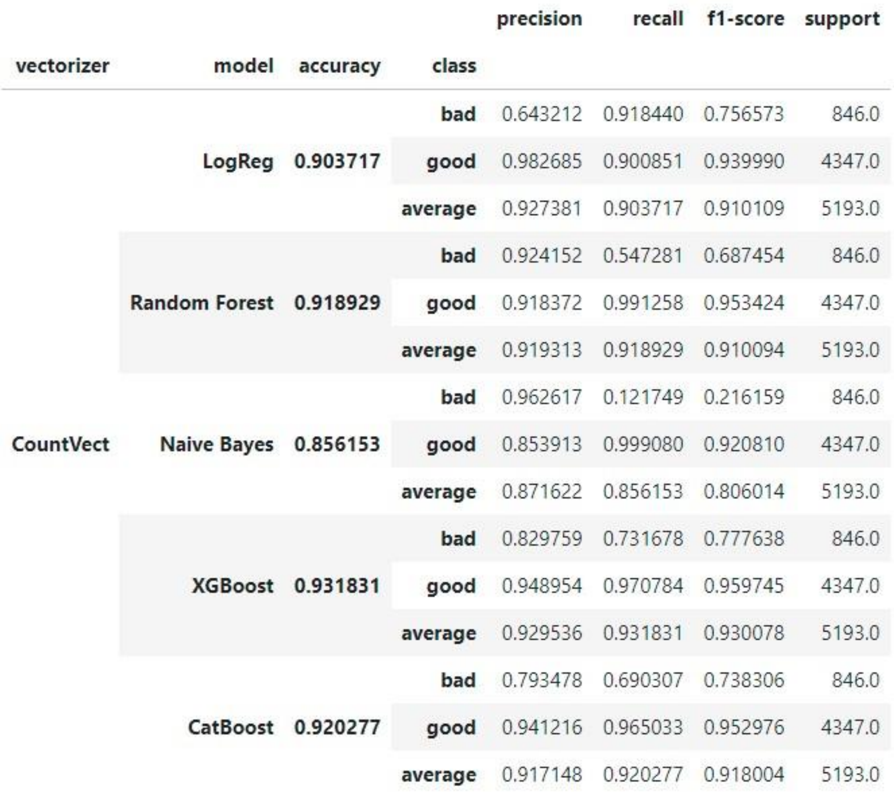
To compare all of the data from the algorithms, each method must be evaluated using appropriate metrics that take into account the class distribution and pay special attention to the minority class. As a result, we employed the f1-score, which is the harmonic mean of accuracy and recall values for a classification issue.

We also implemented a confusion matrix to describe the performance of a classification model on the test data.



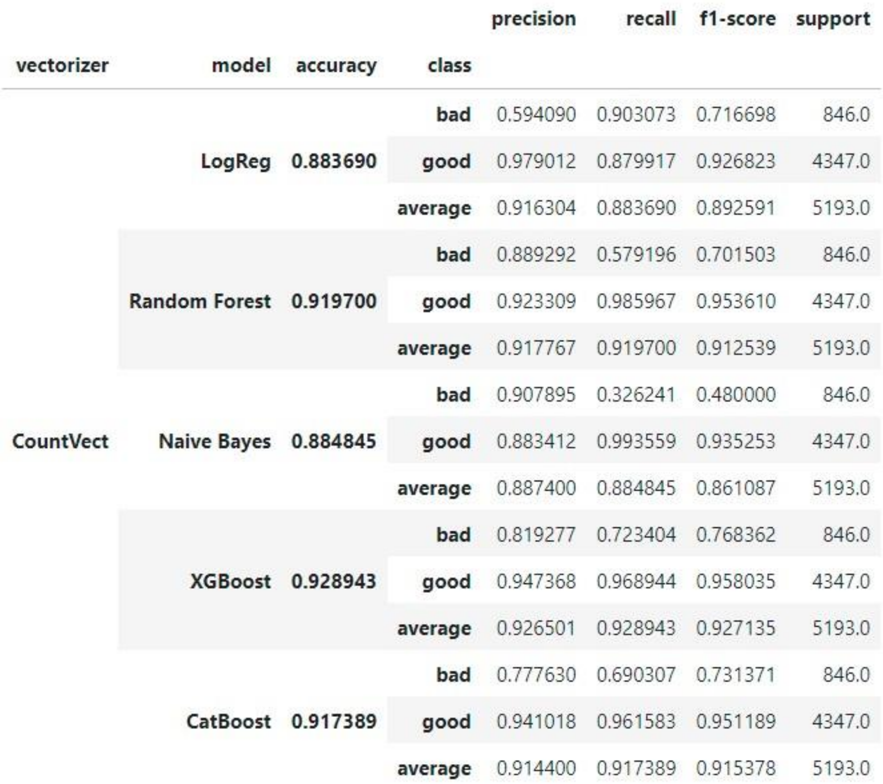
				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.922781	bad	0.712107	0.882979	0.788391	846.0
			good	0.976110	0.930527	0.952774	4347.0
			average	0.933101	0.922781	0.925994	5193.0
	Random Forest	0.919700	bad	0.897959	0.572104	0.698917	846.0
			good	0.922217	0.987348	0.953672	4347.0
			average	0.918265	0.919700	0.912169	5193.0
	Naive Bayes	0.915848	bad	0.740306	0.744681	0.742487	846.0
			good	0.950253	0.949160	0.949707	4347.0
			average	0.916050	0.915848	0.915948	5193.0
	XGBoost	0.933179	bad	0.834899	0.735225	0.781898	846.0
			good	0.949640	0.971705	0.960546	4347.0
			average	0.930948	0.933179	0.931442	5193.0
	CatBoost	0.922010	bad	0.793609	0.704492	0.746399	846.0
			good	0.943719	0.964343	0.953920	4347.0
			average	0.919264	0.922010	0.920112	5193.0

Figure 11. Comparison of different models using Count Vectorizing



				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.903717	bad	0.643212	0.918440	0.756573	846.0
			good	0.982685	0.900851	0.939990	4347.0
			average	0.927381	0.903717	0.910109	5193.0
	Random Forest	0.918929	bad	0.924152	0.547281	0.687454	846.0
			good	0.918372	0.991258	0.953424	4347.0
			average	0.919313	0.918929	0.910094	5193.0
	Naive Bayes	0.856153	bad	0.962617	0.121749	0.216159	846.0
			good	0.853913	0.999080	0.920810	4347.0
			average	0.871622	0.856153	0.806014	5193.0
	XGBoost	0.931831	bad	0.829759	0.731678	0.777638	846.0
			good	0.948954	0.970784	0.959745	4347.0
			average	0.929536	0.931831	0.930078	5193.0
	CatBoost	0.920277	bad	0.793478	0.690307	0.738306	846.0
			good	0.941216	0.965033	0.952976	4347.0
			average	0.917148	0.920277	0.918004	5193.0

Figure 12. Comparison of different models using TF-IDF



				precision	recall	f1-score	support
vectorizer	model	accuracy	class				
CountVect	LogReg	0.883690	bad	0.594090	0.903073	0.716698	846.0
			good	0.979012	0.879917	0.926823	4347.0
			average	0.916304	0.883690	0.892591	5193.0
	Random Forest	0.919700	bad	0.889292	0.579196	0.701503	846.0
			good	0.923309	0.985967	0.953610	4347.0
			average	0.917767	0.919700	0.912539	5193.0
	Naive Bayes	0.884845	bad	0.907895	0.326241	0.480000	846.0
			good	0.883412	0.993559	0.935253	4347.0
			average	0.887400	0.884845	0.861087	5193.0
	XGBoost	0.928943	bad	0.819277	0.723404	0.768362	846.0
			good	0.947368	0.968944	0.958035	4347.0
			average	0.926501	0.928943	0.927135	5193.0
	CatBoost	0.917389	bad	0.777630	0.690307	0.731371	846.0
			good	0.941018	0.961583	0.951189	4347.0
			average	0.914400	0.917389	0.915378	5193.0

Figure 13. Comparison of different models using Hash Vectorizer

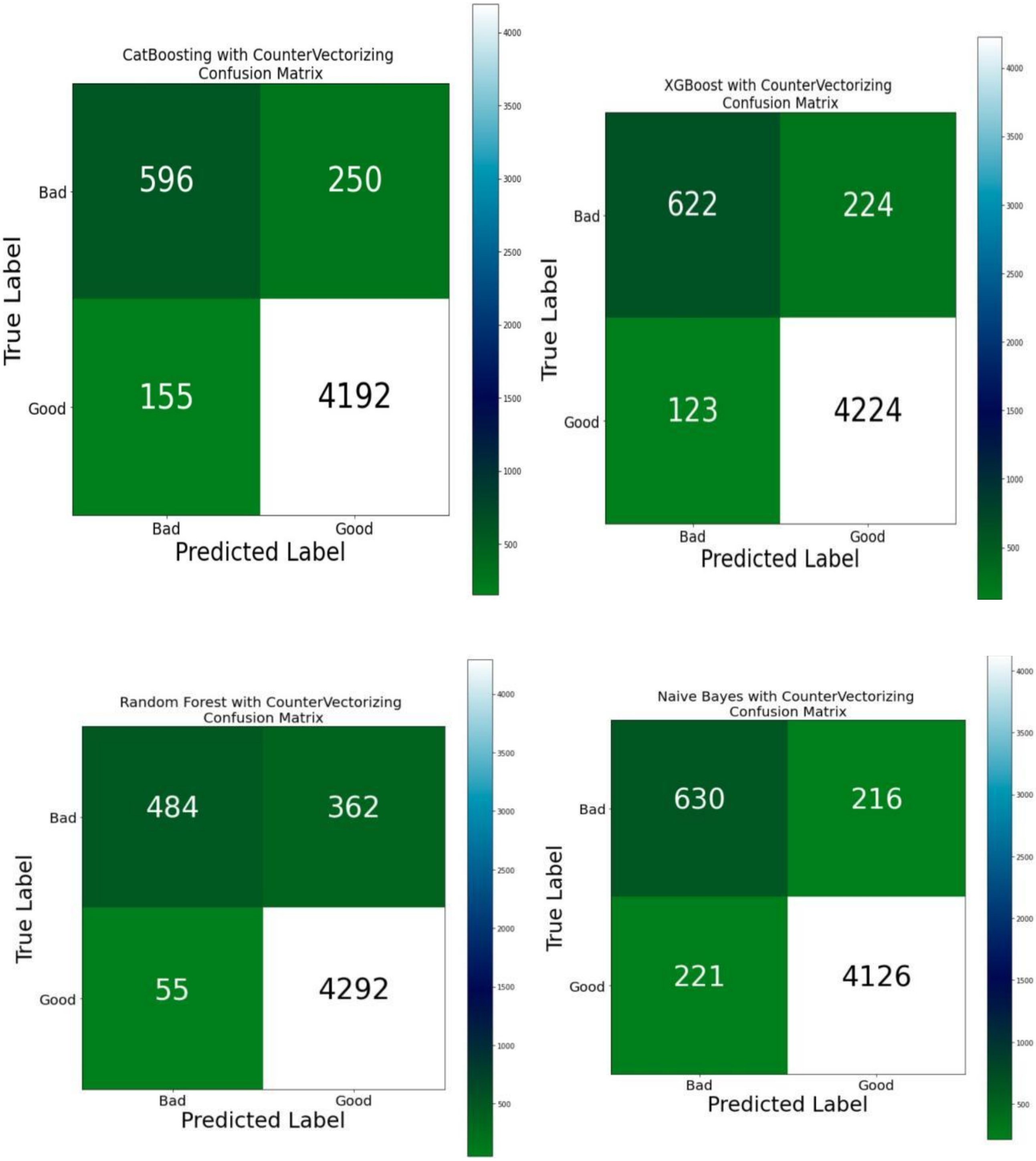
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

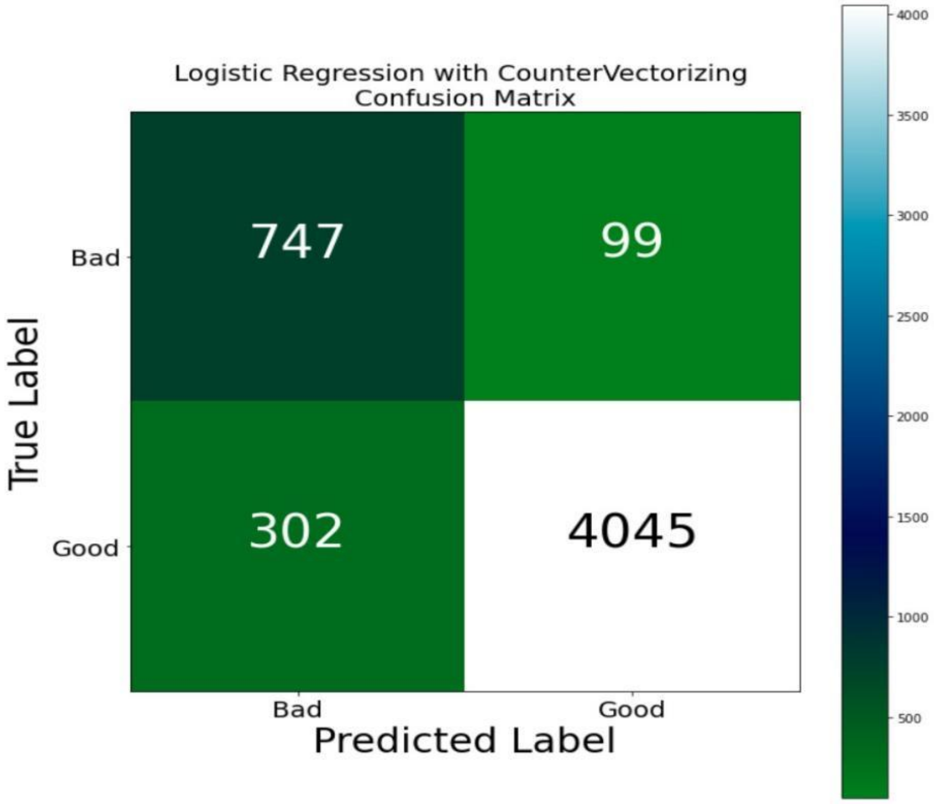
Impact Factor 7.39 ⁜ Vol. 11, Issue 3, March 2022

DOI: 10.17148/IJARCCE.2022.11356

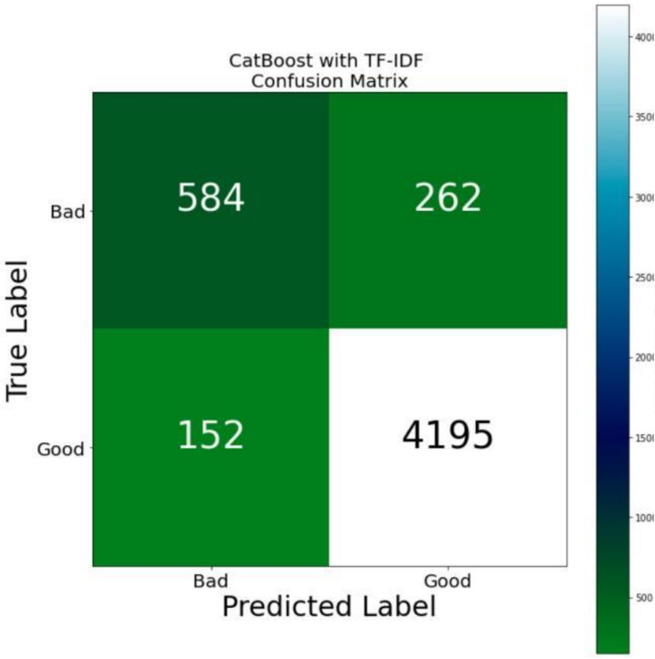
Confusion Matrix of different models with Counter Vectorizing

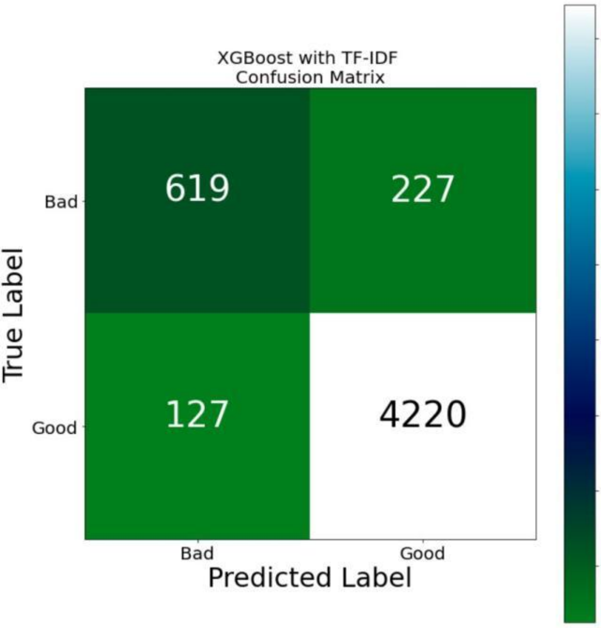


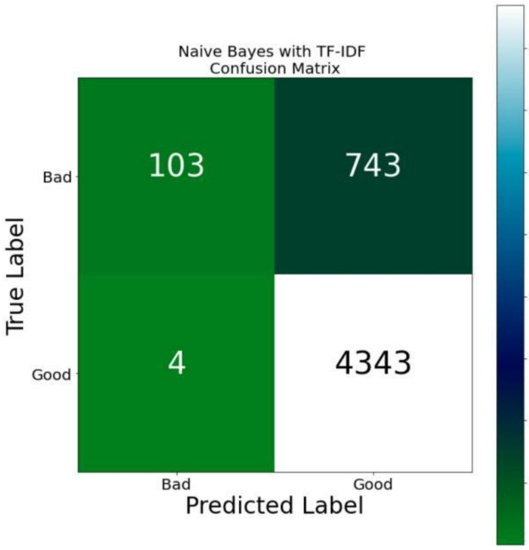
DOI: 10.17148/IJARCCE.2022.11356

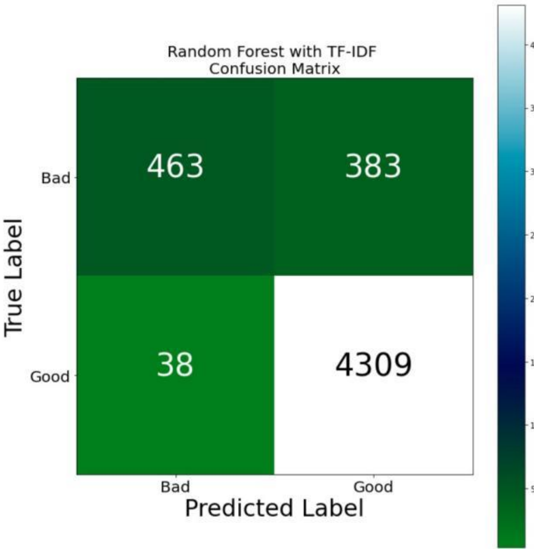


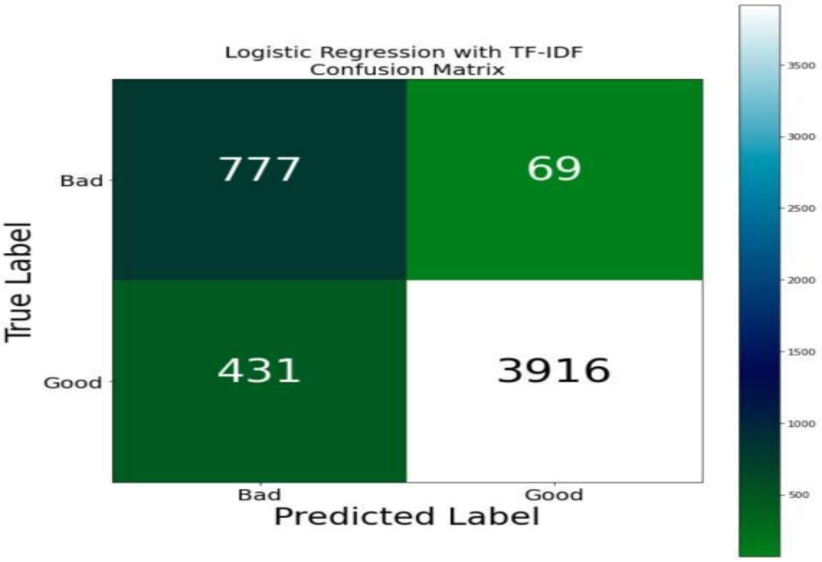
Confusion Matrix of different models with TF-IDF



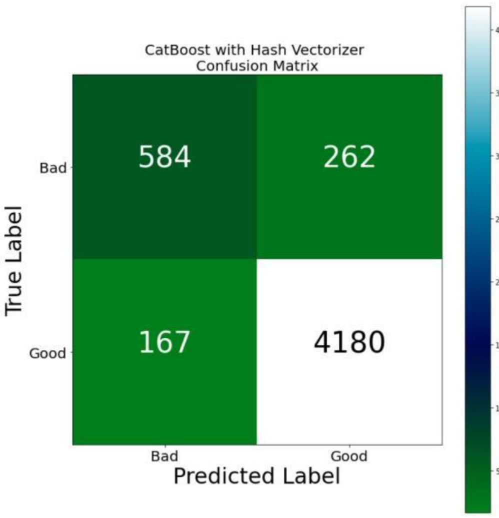


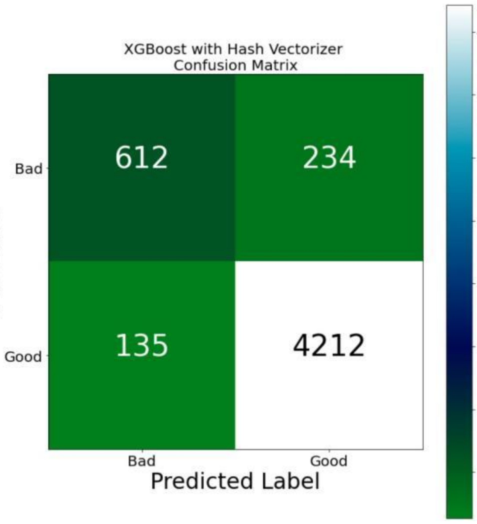




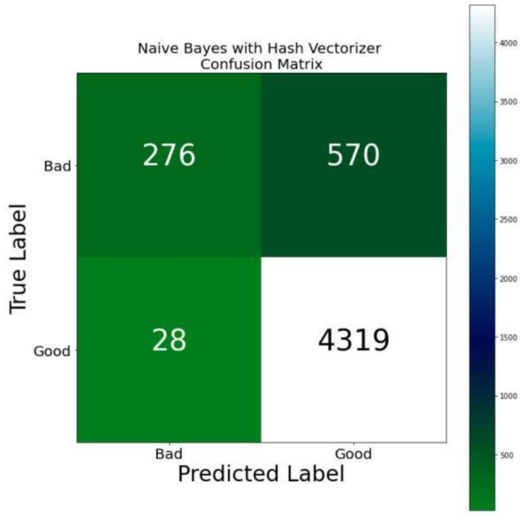


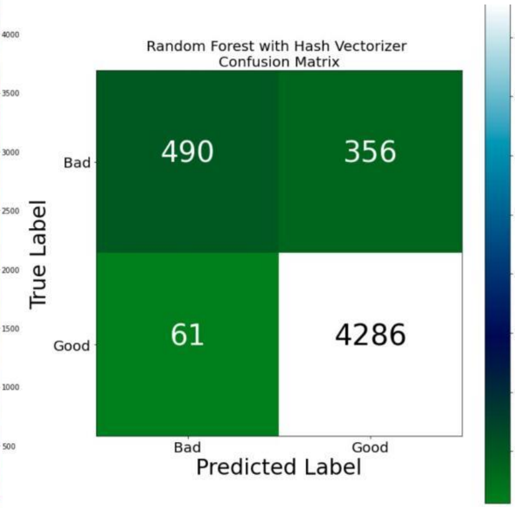
Confusion Matrix of different models with Hash Vectorizer

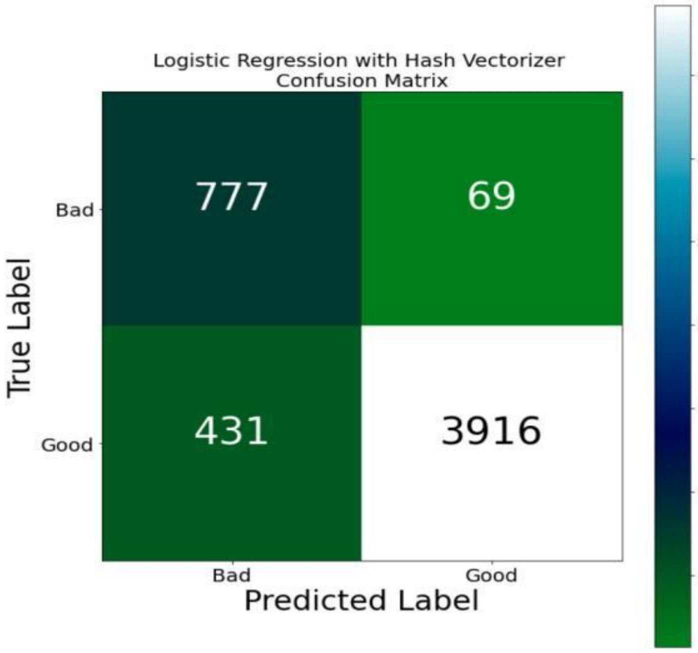




DOI: 10.17148/IJARCCE.2022.11356







1.4 RECOMMENDATION SYSTEM

Whether it's for e-commerce or online advertising, recommender systems are now an indisputable part of our daily online lives. A recommendation engine filters data using various algorithms, determining whether a user and an item are compatible, and inferring similarities between users and items in order to make recommendations.

1.4.1 Types of Recommendation System

I. Content based

The system learns to suggest goods that are similar to those that the user has previously purchased. It made its decision by analyzing user profiles and product information such as features and category [5].

II. Collaborative Filtering

It is predicated on the assumption that individuals prefer things that are similar to other things they like, as well as items that other people with similar tastes like. Instead than focusing on the attributes of other items, this form of recommender system takes a more social approach based on other people's tastes [6].

DOI: 10.17148/IJARCCE.2022.11356

III. Demographic

This system makes suggestions based on the demographic characteristics of the user [5]. It categorizes people based on their characteristics and suggests items based on their demographic information. It's simple to set up and doesn't require user feedback [7].

IV. Knowledge-Based

The system is based on the domain knowledge about how a user will be given product recommendations based on their needs and utility [5]. In this form of system, the behaviour of other users does not take centre stage.As a result, it may be utilized to overcome the shortcomings of typical recommendation methods. When employing the knowledge-based technique, no huge data collection is required, and the suggestions are more credible since the domain knowledge on which they are based is noise-free [8].

V. Hybrid Systems

These systems are built using a mix of the strategies listed above. A hybrid system that combines content-based and collaborative elements, with the goal of using content-based to compensate for collaborative system's inadequacies. Several studies have compared the effectiveness of traditional approaches to hybrid methods, concluding that adopting hybrid methods results in more accurate suggestions [5].

1.4.2 Item - Item Collaborative Filtering

We'll look at the relationship between the pair of items. The item-based technique examines the collection of things assessed by the target user, calculates how similar they are to the target item 'i' and then chooses the 'm' most similar items.

	User 1	User 2	User 3	User 4
i1	4	2	NAN	NAN
i2	1	-	3	4
i3	NAN	4	1	NAN
i4	4	2	2	NAN

Let's look at User 2 as an example of how to anticipate a user's rating. To begin, we must locate User 2's co-rated goods, which are items i1, i3, and i4. We've now discovered the similarity between i2 and item i1, as well as i3 and i4. Finally, based on the similarity and co-rated rating, we must make a prediction.

$$\text{Prediction} \;=\; \frac{\sum_n s_{i,n} \times r_{u,n}}{\sum_n |s_{i,n}|}$$

where w is the similarity, and r is the rating value [9].

1.4.3 Machine Learning Algorithms

1. Cosine Similarity

Cosine similarity is a statistic that compares the size of two things or documents to see how similar they are. Two things are thought of as two vectors in the m-dimensional user-space in this scenario. The cosine of the angle between these two vectors is used to determine their similarity.

The cosine of the angle between two vectors is calculated by dividing the dot product of the two vectors by the magnitude product of the two vectors. Similarity between items i and j, denoted by sim(i, j) is given by [10]

$$\text{sim}(i,j) \;=\cos(\vec{i},\vec{j}) \;=\frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\,\sqrt{\sum_{i=1}^{n} q_i^2}}$$

2. Correlation Similarity

The Pearson Correlation Coefficient (PCC) is one of the most commonly used similarity metrics in collaborative filtering recommender systems to determine how closely two users are linked. This similarity metric is based on how much common users' ratings for a pair of things differ from the average ratings for those items [10].

$$\mathrm{sim}(i,j) = \frac{\sum_{u\epsilon U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u\epsilon U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u\epsilon U}(R_{u,j} - \bar{R}_j)^2}}$$

3. Adjusted Cosine Similarity

When using the cosine similarity measure for item-based CF, differences in user ratings are ignored. Adjusted cosine similarity compensates for this flaw by removing each co-rated pair's average user rating, as shown below.

$$\mathrm{sim}(i,j) = \frac{\sum_{u\epsilon U}(R_{u,i} - \overline{R_u})(R_{u,j} - \overline{R_u})}{\sqrt{\sum_{u\epsilon U}(R_{u,i} - \overline{R_u})^2}\sqrt{\sum_{u\epsilon U}(R_{u,j} - \overline{R_u})^2}}$$

Because our data was sparse, we utilized cosine similarity to determine the similarities between all of the products and create a final similarity matrix.

1.4.4 Evaluation Metrics

Individuals can more successfully identify their interests from a range of options with the aid of a recommendation system that incorporates the opinions of different users. In its particular arena, each algorithmic technique is superior. As a result, various measures should be used to evaluate the optimal technique. Our standard ML evaluation metrics to evaluate ratings and predictions that are Precision, Recall, F1-measure, False-positive rate, Mean average precision, Mean absolute error, and The area under the ROC curve (AUC) [11].

We used RMSE as our metrics to find out the accuracy of our system as it is the most used and easiest to execute. It is calculated by taking the average of all squared differences between the true and projected scores and then taking the square root of the result. As a result, large errors can have a considerable impact on the RMSE rating, making the RMSE statistic most useful when considerably large errors are undesirable. The root mean square error between true and anticipated ratings is calculated as follows [12] :

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(r_i - \hat{r}_i)^2}{n}}$$

Where r_i is the actual rating, $\hat{r}_i$ is the predicted rating and n is the amount of ratings.

1.5 CONCLUSION

Recommendation systems are computer programs that provide recommendations to users based on a variety of parameters. It is based on the discovery of patterns in consumer behaviour data, which may be obtained both implicitly and overtly. These algorithms predict which products people are most likely to purchase and are most interested in. Using recommender systems, both consumers and suppliers profit. In an online shopping environment, they reduce the costs of looking for and selecting items.

We tested a collaborative filtering recommender system in this article by comparing the ratings of each product to their text reviews using sentiment analysis. Our findings show that item-based collaborative filtering is an effective method for obtaining high-quality recommendations and it can be used for large data-set as well. We implemented cosine similarity to create the similarity matrix to recommend items to user. Using RMSE as our evaluation metric, we found out our system has a accuracy of 0.52 which is better than average.

DOI: 10.17148/IJARCCE.2022.11356

1.6 REFERENCES

1. Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. International journal of research in industrial engineering, 9(1), 84-98.
2. Balush, I., Vysotska, V., & Albota, S. (2021). Recommendation System Development Based on Intelligent Search NLP and Machine Learning Methods. In CEUR Workshop Proceedings (Vol. 2917, pp. 584-617).
3. Von Reischach, F., Guinard, D., Michahelles, F., & Fleisch, E. (2009, March). A mobile product recommendation system interacting with tagged products. In 2009 IEEE international conference on pervasive computing and communications (pp. 1-6). IEEE.
4. Kumar, P., & Thakur, R. S. (2018). Recommendation system techniques and related issues: a survey. International Journal of Information Technology, 10(4), 495-501.
5. Francesco Ricci, Lior Rokach, and Bracha Shapira (2015). Chapter 1 Recommender Systems: Introduction and Challenges. Recommender Systems Handbook, DOI 10.1007/978-1-4899-7637-6_1
6. Yen-Yao Wang, Andy Luse, Anthony M. Townsend, Brian E. Mennecke. Understanding the moderating roles of types of recommender systems and products on customer behavioral intention to use recommender systems.Information Systems and e-Business Management 13, 769-799 (2015) https://doi.org/10.1007/s10257-014-0269-9
7. M.Sridevi, Dr .R.Rajeswara Rao (2017).DECORS: A Simple and Efficient Demographi collaborative Recommender System for Movie Recommendation ISSN 0973-6107 Volume 10, Number 7 (2017) pp. 1969-1979.
8. Sarah Bouraga,Ivan Jureta1, St´ephane Faulkner, and Caroline Herssens (2014). Knowledge-Based Recommendation Systems: A Survey. International Journal of Intelligent Information Technologies (IJIIT) 10(2) 10.4018/ijiit.2014040101
9. Jeffery chiang (2021). Overview of Item-Item Collaborative Filtering Recommendation System. MEDIUM
10. Badrul Sarwar, George Karypis, Joseph Konstan, and John Ried. Item-Based Collaborative Filtering Recommendation Algorithms. WWW '01: Proceedings of the 10th international conference on World Wide Web (May 2001) Pages 285–295 https://doi.org/10.1145/371920.372071
11. Denis Parra, Shaghayegh Sahebi. Recommender Systems: Sources of Knowledge and Evaluation Metrics. Advanced Techniques in Web Intelligence-2 pp 149-175. Part of the Studies in Computational Intelligence book series (SCI, volume 452)
12. SAFIR NAJAFI, ZIAD SALAM. Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems. DEGREE PROJECT IN TECHNOLOGY, FIRST CYCLE, STOCKHOLM, SWEDEN 2016.
13. Burke, R. (1999, July). Integrating knowledge-based and collaborative-filtering recommender systems. In Proceedings of the Workshop on AI and Electronic Commerce (pp. 69-72).
14. Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. J. Softw., 5(7), 745-752.
15. Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. Now Publishers Inc.