# PRODUCT RECOMMENDATION USING SENTIMENT ANALYSIS OF PRODUCT'S REVIEW IN E-COMMERCE

**Abstract**

Recommendation is very crucial part of human life; we rely on recommendation for almost every stuff in our daily routine. In this research work we worked on sentiment analysis by applying frequent pattern mining on the reviews of products provided by the customers on e-commerce platform. It is very easy to recommend a product based on its rating which is usually out of 5, but to deal with the reviews written in few thousands then it would be not feasible for any individual to go through and then select the best product out of available. It is very essential to identify the polarity of the given review by analyzing the sentiment of that particular review. We are applying BoW model and extract the frequent pattern by filtering positive keyword to check the polarity of the review and finally recommending the product to the customer.

**Keywords**:MachineLearning; RecommendationSystem;Artificial Intelligence; NLP; Sentiment Analysis

**Authors**

**Dr. Bhrantav Vora**
Assistant Professor
LDRP Institute of Research and Technology
MCA Department, KSV University
Gandhinagar,India.

**Dr. Meghna Patel**
Associate Professor
A. M. Patel Institute of Computer Studies
GanpatUniversity, Kherava, India.

**Dr.Ronak B. Patel**
Associate Professor
Shrimad Rajchandra Institute of Management and Computer Application
UkaTarsadiaUniversity, Maliba Campus
Barodli,Gujarat.

**Prof. Megha Shah**
Assistant Professor
B P College of Computer Studies
KSV University, Gandhinagar
India.

## I. INTRODUCTION

Sentiment Analysis also known as Opinion Mining plays a vital role in recommending products online through e-commerce platforms. The huge advancement in E-Commerce worldwide avails this opportunity to mine the frequent pattern and extract the insights from the product's review through opinion mining or sentiment analysis [1]. Fig. 1 depicts the process of sentiment analysis on review given for a product. While selecting any product in an online e-commerce marketplace, customers always rely on ratings given to that product generally out of 5. This is an attempt to provide extra support to the customer in their decision making to finalize the product selection through recommendation [1].

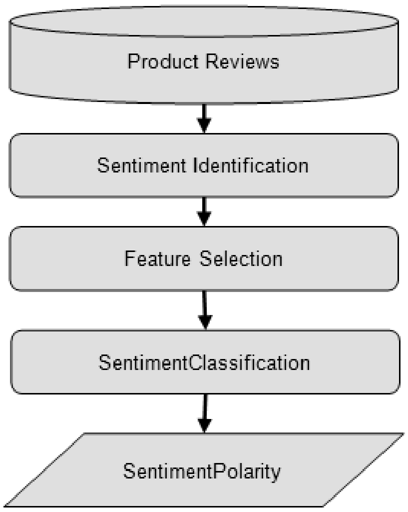


**Figure 1:** SA Process

## II. RELATED WORK

Recommendation System (RS) is a very crucial component for e-commerce giants, for an individual in his/her daily routine he/she heavily depends on recommendation whether it is selecting a school for their kids, buying new car/property, consulting a doctor/lawyer, or even purchasing items online. There are three types of recommendation system 1. Content Based Filtering, 2. Collaborative Filtering and third is the mixing for first two that is 3. Hybrid filtering [2].

1. **Content based filtering:** Content based filtering technique uses features of the product to work with and recommend similar other products to the customer who is most likely to buy the product, It works on observing previous actions of the customer or by using explicit feedback from the customer.

2. **Collaborative filtering**: Collaborative filtering technique uses demographic data of the customer and prepares a customer profile, then it would filter out the product for other customers whose profile matches and based on that recommendation can be done.

3. **Hybrid filtering:** Collaborative filtering and content-based filtering both comes together in Hybrid filtering approach by combining the features of both the techniques.

Recommendation is a three-phase process which includes information collection Phase, learning Phase and finally, recommendation/prediction [3].

- **Information collection phase:** In this phase of recommendation all the necessary information is being collected through various means which include web scraping, APIs, surveys, and avail huge datasets from portals like Kaggle.

- **Learning phase:** Learning phase is very important in the recommendation system, where ML algorithms take place in order to learn the frequent pattern and further help in making predictions in later stages.

- **Prediction/Recommendation phase:** This is the final phase of the recommendation system. In this phase RS will try to predict or recommend the best suitable option from the available options based on the learning using ML algorithms in the previous phase. If needed feedback is also provided to the information/data collection phase to improve the prediction. Fig. 2 depicts the recommendation phases [3].

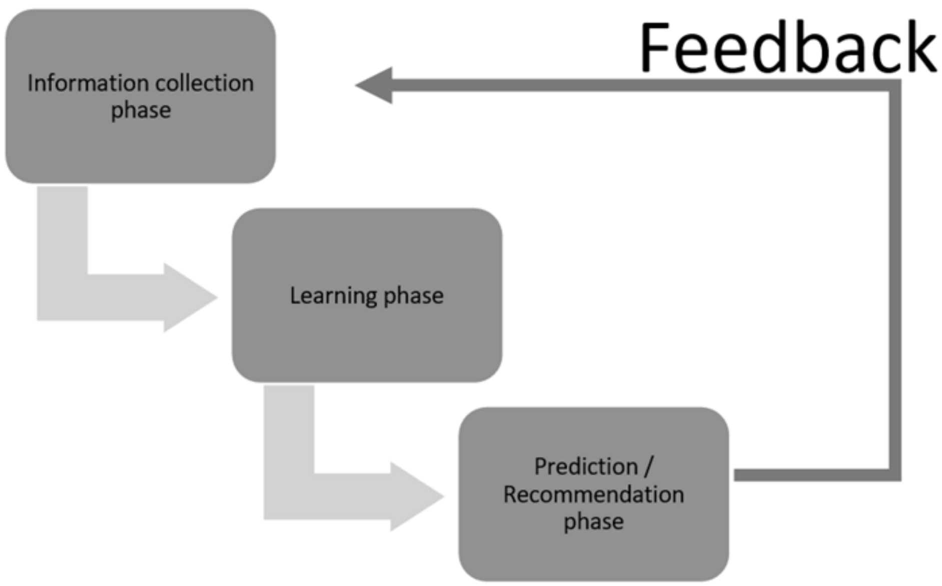


**Figure.2 :** Recommendation Phases.

In this research study, a recommendation model was designed, developed and implemented which analyzes using patterns which are extracted from the reviews of the individual customer and rating given from the individual customer. The recommendation algorithm is used in the proposed system to recommend the product at different levels to the customer. In this research our proposed model accepts users' review of more than one product and gives a level of recommendation i.e. most likely to purchase, likely to purchase, recommended, less recommended and not recommended at all.

The principle thought behind the Recommendation frameworks for E-Commerce is to fabricate a connection between the products, customers, and settle on the choice to choose the most suitable item to-a particular customer[1]. Gone ahead time, consistently in a hurry, current customers shop in erupts from numerous gadgets, as opposed to set aside the effort to take part in a careful shopping long-distance race[4]. Brands conveying fast, on-point offers to catch the most advantages. As indicated by the Personalization Consumer Survey did by 48% of customers go through additional with an E-Commerce organization conveying customized shopping knowledge[2]. ML frameworks enable you to catch information from past and current shopping sessions and change them into dynamic offers[13]. ML can pinpoint which product is in stock to feature[14]. Keen recommender frameworks can look over your whole item index and line up the best items for individual purchasers. You can make it a stride further and show the closest blocks[3]. E-Commerce defines the scope of products is accessible, to peruse the information, focusing on the prospect with nearby stock promotions later in the day[15]. A few out of every unique possibility is prepared to purchase from you at this very moment[16]. When a particular product is absent on an E-Commerce site, many customers will go looking somewhere else. Indeed, even graphic inquiries like 'a white shirt with brilliant catches' or 'nutrients for winter' may not lead customers to the items they need. On location, web search tools are somewhat inadequate when given random questions[17].

AI calculations can manage this issue. Aside from being prepared to perceive a more extensive scope of equivalent words, they can likewise help you naturally arrange your items dependent on their highlights[18]. Profound learning calculations are as of now equipped for dissecting item pictures and separating them into specific traits, for example, Slipover, A-line skirt, knee-length, and so on[19]. Recommender frameworks are utilized by E-business locales to propose items to their consumers. The items could be prescribed depending on top dealers online, in light of the socioeconomics of the consumers, or dependent on the investigation of the previous purchasing conduct of the consumersas an expectation for upcoming purchasing conduct[20]. Extensively, these procedures are a piece of personalization on a site since they help the site adjust to every customer. Recommendation Systems give the experience of personalization on the Web,which engages the customer for personalization experience[21]. To understand as very private to this degree is individual approach to understand author's thoughts on the internet. In this manner, author would presumably concur with the CEO of Amazon online superstores, when he spoke: "On the off chance that I have 2 million consumers online, I thought to have 2 million online stores[22]. " Recommender frameworks upgrade E-business deals in three different methods: Browsers into purchasers: Visitors to a website regularly investigate the webpage while never obtaining anything[23]. Recommender frameworks can enable customers to find items they want to buy. Strategically pitch: Recommendation frameworks improve strategically pitch by proposing extra items for the customer to buy. If the proposals are great, the normal request size should increment. For example, a site may suggest extra items in the checkout procedure, in light of those items as of now in the shopping basket. Dedication:

In reality, as we know it where a site's rivals are just a tick or two away, picking up customer faithfulness is a basic business system[24]**.** Recommender frameworks improve reliability by making a worth included connection between the site and the customer. Destinations put resources into finding out about their customers, use recommender frameworks to operation that learning, and present custom interfaces that match customer needs. Customers reimburse these destinations by coming back to the ones that best match their needs. The more a customer utilizes the proposal framework – instructing it what they need – the more faithful they are to the site. "Regardless of whether a contender was to assemble precisely the same capacities, a customer would need to invest an unnecessary measure of time and vitality instructing the contender what the organization knows" [25]. A clever framework like recommender takes care of the data over-burden issue on the Web by offering decisions that consider customers' needs or interests[26]. The job of recommender frameworks isn't irrelevant in different significant level sites, for example, Amazon, YouTube, Netflix, Yahoo, and so on[27].

## III. PROPOSED FRAMEWORK

We have organized proposed framework with four different phases in which first phase is to identify the products available from the web portal of e-commerce. Where as in second phase we are working with the user's review by extracting reviews and rating. While in third phase machine learning algorithms are implemented on the information collected in previous phase. Lastly in fourth phase we are applying an algorithm known as steepest ascent hill climbing to recommend the product. Fig. 3 depict the framework to be proposed.

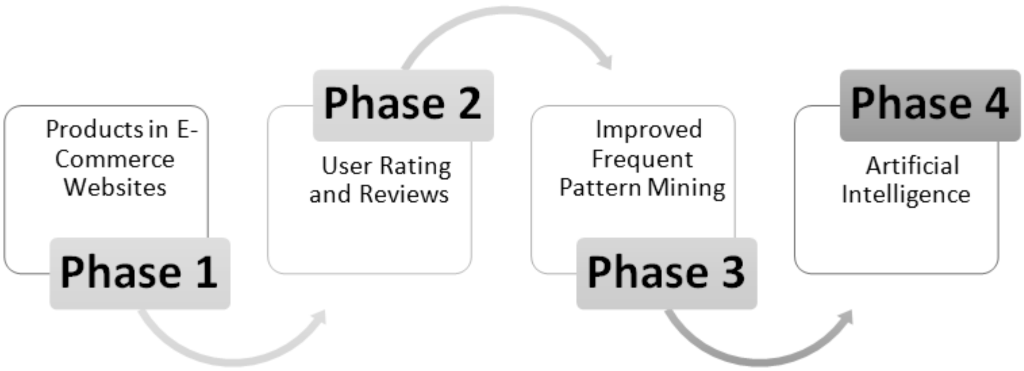


**Figure 3:** Proposed framework

1. **Phase1. Products in e-commerce:** In the first phase of the proposed framework, we are selecting six products from electronic segment, because by comparing with other products we found more number of reviews were there for the products available in electronic segment. As we required enough amount of data to work with, we opted for this option. In this phase we have selected laptop, air condition, washing machine, fridge and television to collect sample data.

2. **Phase2. User Rating and Reviews:** In this phase of user rating and review we have prepared a set of code to scrap details of product details using python script. In this attempt we have extracted few details of products i.e. Model, Review, Rating, Brand, Price. PCA could be implemented to select the features if number of features were too large. But in our research we have selected features manually which include reviews and rating of the product. This extracted information will be helpful in next phase of the proposed framework**.**

3. **Phase3. Improved Frequent Pattern Mining: In this stage, we apply NLP on gathered** surveys in past stage to deal with the regular language, to find proper data from the gathered audits we have applied Sack of Words model and attempt to track down continuous example in audits and changed over all out data into mathematical to handle it further in AI calculation.

4. **Phase4. Artificial intelligence:**In this period of suggestion system, Man-made reasoning is utilized to prescribe the item to the client in view of the got data from the past stage. Steepest rising slope climbing calculation is utilized to find the best adjoining hub to meet the objective state and that item is being suggested, further in progress we have developed steepest climb slope climbing calculation to a higher level and give level of proposal for example Generally suggested, In all probability suggested, Suggested, Less suggested and not suggested by any means.

## IV. EXPERIMENTAL PROCEDURE

In our examination work, we can separate our trial strategy into three significant stages to get an unmistakable comprehension of carried out research. Stage 1 is Information or Data assortment work in which we made a python script in Google colab and carried out web scratching to gather designated information from the online business sites. In Stage 2 we are performing pre-handling and learning errands, and again we have made a different python content to separate information from the information that we have scratched from the past python script made in stage 1. Ultimately in stage 3 which is the proposal or expectation stage we have made another python script in which we have executed the steepest rising slope climbing calculation of computerized reasoning and accomplished the suggestion

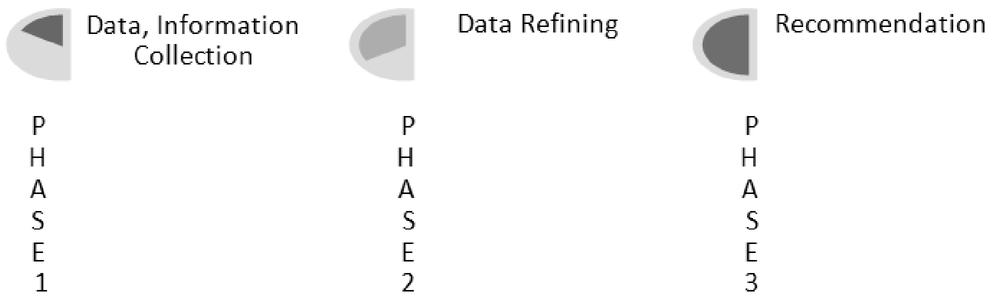


**Figure 4 :** Phases in Experimental Procedure

1. **Data Collection:** Information assortment is one of the most fundamental and vital errands to be finished. To push ahead in research work, a specialist needs to gather information, and information can be accessible from different sources. In our exploration work we have gathered information from online business sites.

    - **Web Scraping:** Web scratching is the method to extricate a lot of organized information from the web. In the time of data and information on the web, removing information from the web is profoundly able. To execute information mining, design acknowledgment, estimating through data set boring one should require adequate measure of information to chip away at and it ought to likewise be accessible in delicate structure to effectively carry out strongly exact and result arranged AI and man-made brainpower calculations and for a similar web is the most reasonable stage that anyone could hope to find. Separating information from the web through various sites utilizing extraordinarily strong prearranging language for example python is web scratching. Fig. 5 shows the manner in which we scrap the information from online business sites.

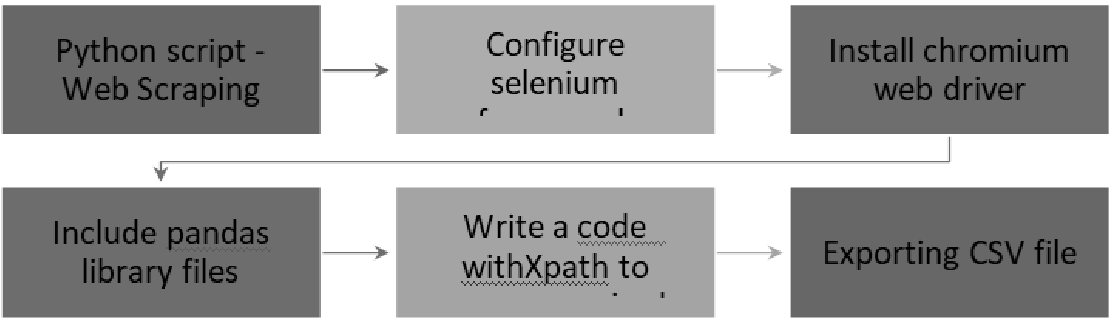


**Figure 5 :** Data collection flow

    - **Python Script:** In our examination work to get clients' survey and rating of item from various online business sites, we have fostered a python script in Google colab which is a cloud based information science work area comparable like Jupyter journal, which additionally give 13 GB of slam and processors going from computer chip, GPU to TPU.

    - **Selenium Framework: In** python script we have consolidated selenium structure which is an open-source online mechanization apparatus with python it engages to associate with the program and can send python orders to various programs.


**Figure 6 :** Selenium configuration in Python notebook

- **Chromium Driver:** We have then designed the ChromeDriver, which is a different executable that Selenium WebDriver uses to control Chrome. X - Way for XML way language is utilized for choosing hubs from a XML record of designated website page.

- **Pandas Library:** We have likewise utilized pandas library records in python, it is a NumFOCUS supported project and an open source BSD-permit library, giving superior execution, simple to-utilize information designs and information examination devices for the Python programming language.

By utilizing previously mentioned instruments we have then created a strong python content to extricate wanted information from the web. Figure shows the code execution of bringing in pandas library document and chromium driver.

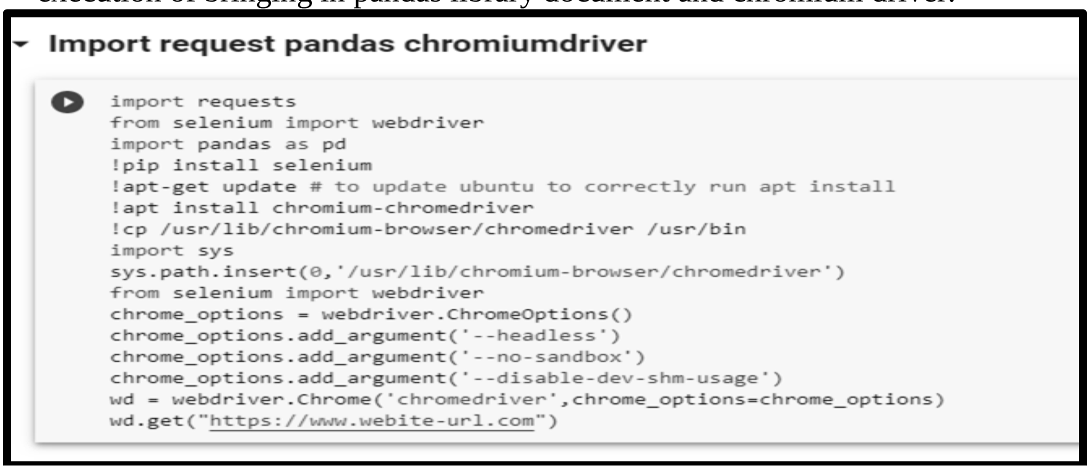


**Figure 7:** Importing Pandas &Chromiumdriver

- Programming with Xpath to scrape data: In the wake of adding vital library records, introducing driver and designed system it is expected to take advantage of the XML produced from the designated page from the chose web based business site. Utilizing the GET demand capability of the web driver expected connect is taken advantage of and to additional mine XML document and get the vital information from it, a Xpath has been utilized in created rationale to circle through rehashed information in XML record. To choose the hub from the XML report, Xpath is the suitable inquiry language. Besides, to figure values from the XML record content XPath is helpful. The Internet Consortium has characterized the Xpath. Given figure referenced how rationale was carried out with Xpath.

Code with logic and Xpath to scrap data from e-commerce website

```
[ ] rating=[]
    review=[]
    for page in range(1,6):
      wd.get('https://w[redacted]ikon-d5600-dslr-camera-body-single-lens-af-s-dx-nikkor-18-140-mm-f-3-5-5-6g-ed-vr-16-gb-sd-
      dat=wd.find_element_by_class_name('_1AtVbE')
      for i in range(3,1300):
        record=dat.find_elements_by_xpath("//*[@id='container']/div/div[3]/div/div/div[2]/div[" + str(i) + "]")
        a=dat.find_elements_by_xpath("//*[@id='container']/div/div[3]/div/div/div[2]/div["+ str(i) + "]/div/div/div/div[1]/div")
        b=dat.find_elements_by_xpath("//*[@id='container']/div/div[3]/div/div/div[2]/div["+ str(i) + "]/div/div/div/div[2]/div")
        for a_rating in a:
          rating.append(a_rating.text)
        for b_review in b:
          review.append(b_review.text)
    table = pd.DataFrame(review,columns=['Review'])
    table ["Rating"] = rating
    # table
```

**Figure 8 :** Xpath usage to scrape data

- Exporting extracted data : Subsequent to applying rationale with the assistance of Xpath, and separating the expected information, it is exceptionally expected to change over that information into the even structure through which further handling should be possible. For that we utilized a component of pandas library document and traded information into csv design for the utilization in the following stage. Fig. 9 portrays the trading information into CSV document design.

▾ Exporting CSV File

```
[ ] table.to_csv("NikonD5600DSLRCameraBodywithSingleLens67900.csv")
```

**Figure 9:** Exporting scraped data in csv file

- Primary Data Collection: Execution assessment of the framework, which takes contribution of a document containing 75026 records of various items like Cell Phones, Cooler, TV, Forced air system, Clothes washer, and PC which has the accompanying credits Survey, Rating. Grouping of the separated information is given underneath.

**Table 1: Summary of scrapped data**

| [redacted] | | | | |
|---|---|---|---|---|
| 1 | AC | 2636 | 6 | 18 |
| 2 | Laptop | 4570 | 2 | 6 |
| 3 | MobilePhone | 31880 | 9 | 27 |
| 4 | Refrigerator | 11090 | 7 | 21 |
| 5 | TV | 13930 | 7 | 21 |
| 6 | WashingMachine | 10920 | 7 | 21 |
| Total | 6 | 75026 | 38 | 114 |

**2. Data Refining:** Information preprocessing is vital, any step pushing ahead without clear comprehension and doing legitimate pre-handling might prompt monstrous mistake or we might place ourselves into a crush. Consequently it is exceptionally expected to perform information pre-handling which incorporates information cleaning and afterward carrying out a pack of word model to track down catchphrases for the further use. Following figure portrays the manner in which we accomplish pre-handling undertakings.

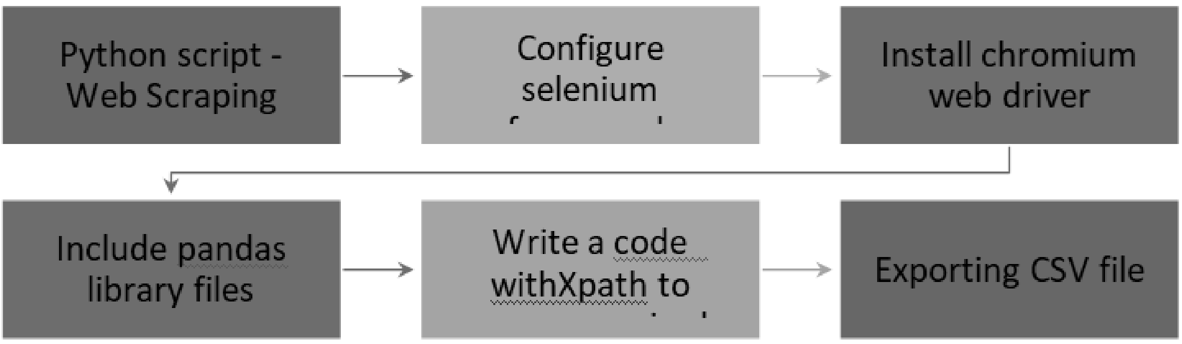


**Figure 10:** Pre-processing and Learning workflow

- **Feature selection**To apply design mining on scratched information, it is expected to have exec:muted highlight determination which is one of the center ideas in ML, it influences tremendously the model execution. The information includes that we use to prepare our proposed model incredibly affect the exhibition we could accomplish. Highlights that are not important may impact the model presentation adversely. Thus it is strongly prescribed to clean the information and select fitting highlights prior to moving further.

By applying highlight choice, we could stay away from the issue of Over fitting and could be benefited by Further develops Precision and Decreases Preparing Time. In this cycle we can naturally or
physically select those elements which contribute most to our expectation variable or result wherein we are keen on. We have chosen include physically and we have chosen two vital element which could profoundly affect our model those highlights are appraising and surveys.

- **Import File from previous phase:** Subsequent to performing highlight choice, we are presently prepared to perform important errand on the assembled information gathered in the information data assortment stage. Subsequent to bringing in important library records for example pandas, google-collab records and matplotlib.pyplot, we then imported the csv document which was traded from the past stage. Fig. 11 shows the essential code to carry out something very similar.

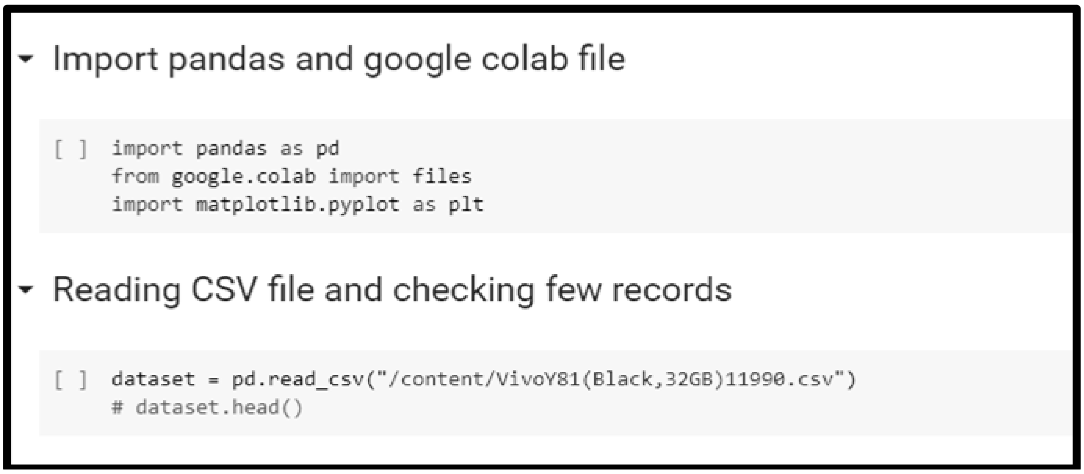


**Figure 11** : Importing CSV file

- **Data Cleaning:** Our work is on clients' survey and positioning of items got from web based business locales. By carrying out a fruitful python script, we assembled an adequate number of information to manage, while managing positioning it is easy to deal with numeric information however while dealing with surveys, which is text including unique images, emoticon and extraordinary characters which are not needed in additional means. Subsequently suitable information cleaning code in python script was created and we dispose of undesirable substance from the audits in information cleaning. Following two figures Fig. 12 and Fig. 13 show how cleaning processes happened through python script.

▾ Data count , count Null record

```
[ ] dataset.count()
```

```
[ ] dataset['Review'].isnull().sum() # to count null data or missing values
```

▾ Removing null records

```
[ ] dataset=dataset.dropna(axis=0, how='any') # remove null data
    dataset.count()
```

**Figure 12:** Removing null records

▾ Cleaning

```
[ ] dataset["Review"]=dataset["Review"].str.replace("."," ") # remove dots
    dataset["Review"]=dataset["Review"].str.replace("\n"," ") # remove \n
    dataset["Review"]=dataset["Review"].str.replace("#"," ") # remove #
    dataset["Review"]=dataset["Review"].str.replace("❤"," ") # remove ❤
    dataset["Review"]=dataset["Review"].str.replace("😎"," ") # remove 😎
    dataset["Review"]=dataset["Review"].str.replace("👍"," ") # remove 👍
    dataset["Review"]=dataset["Review"].str.replace("□"," ") # remove □
    dataset["Review"]=dataset["Review"].str.replace("😍"," ") # remove 😍
    dataset["Review"]=dataset["Review"].str.replace("😲"," ") # remove 😲
    dataset["Review"]=dataset["Review"].str.replace("💯"," ") # remove 💯
    dataset["Review"]=dataset["Review"].str.replace(":"," ") # remove :
    dataset["Review"]=dataset["Review"].str.replace("□"," ") # remove □
    dataset["Review"]=dataset["Review"].str.replace("😛"," ") # remove 😛
```

**Figure 13:** Data Cleaning

- **Text Transformation and NLP:**Subsequent to carrying out highlight choice and information cleaning, in our examination work at this stage we are promptly accessible with the specific substance we need to manage. The substance accessible with us are clients' appraising and surveys. It is very simple to manage appraisals which are mathematical information and could be fitted in AI calculations effectively, yet the equivalent isn't true with audits. It is expected to change the text from all out to mathematical with the goal that it very well may be fitted into an AI calculation and produce the necessary outcome. We have handled the regular language utilizing a pack of word models and changed the text from all out to mathematical. To accomplish wanted yield from the text we expected to gather all singular surveys into a solitary string object. Fig. 14 portrays the execution of the equivalent.

▾ Combining review in one single object for frequent word count

```
[ ] strobj = ""
    for index, row in dataset.iterrows():
      strobj = strobj + row['Review']
```

**Figure 14:** Combining all reviews in one object

- **Bag of Words model:** Pack of words model is a data recovery through regular language handling (NLP) to improve on portrayal. A message record is addressed as a pack of words it contains and by disregarding the syntax and word request it gives the variety of words in the report given as a contribution to it.

  In our python script all text content gathered in one article called pack of word and afterward by applying the Sack of Words model to it gives the most successive word in it, which further assists in finding with keying words to change over key data from unmitigated to mathematical to execute further proposal calculations on it. Following figure Fig. 15 shows the execution of the "Pack of Word model" in our python script.

▾ Bag of Word

```
[ ] from collections import Counter
    split_it = strobj.split()
    Counter = Counter(split_it)
    most_occur = Counter.most_common(50)
    ForGraph = []
    for i in most_occur:
      if(i[0]=='awesome' or i[0]=='best' or i[0]=='good' or i[0]=='nice' or i[0]=='like' or i[0]=='great' or i[0]=='like' or i[0]=='amazing'):
        ForGraph.append(i)
    print(most_occur)
```

**Figure 15 :** Bag of Word Model

- **Improved Frequent Pattern Mining:** Continuous examples are sets of information or thing sets which regularly show up in the accessible informational index which has recurrence more noteworthy than or equivalent to the predetermined limit. Mining continuous example is really information mining which has a target to extricate incessant information or thing sets from the given informational index. Finding continuous examples from the informational index is vital and it assumes an extremely huge part in various information mining errands. By applying a pack of word model we have distinguished habitually seeming positive words from the surveys and in view of the recurrence we have chosen the positive catchphrase to additionally dole out rank to every positive word.

- **Selecting Keyword:** From the Sack of Word model we could distinguish the continuous example showing up in the information, however it means a lot to choose suitable catchphrases to relegate rank. To choose the right catchphrase we have then utilized a diagram from the matplotlib and attempt to view which watchword is as chosen. Then we get the unmistakable plan to best choose the accompanying five watchwords to be specific "great", "pleasant", "like", "amazing" and". We likewise keep the counter refreshed by refuting the positive word utilized in bad things. For instance, great is a positive word yet when it is gone before by not, it becomes bad. We dealt with such events and afterward we arranged and chose the catchphrase. Following figure Fig. 16 portrays diagram execution and how we select the watchword.

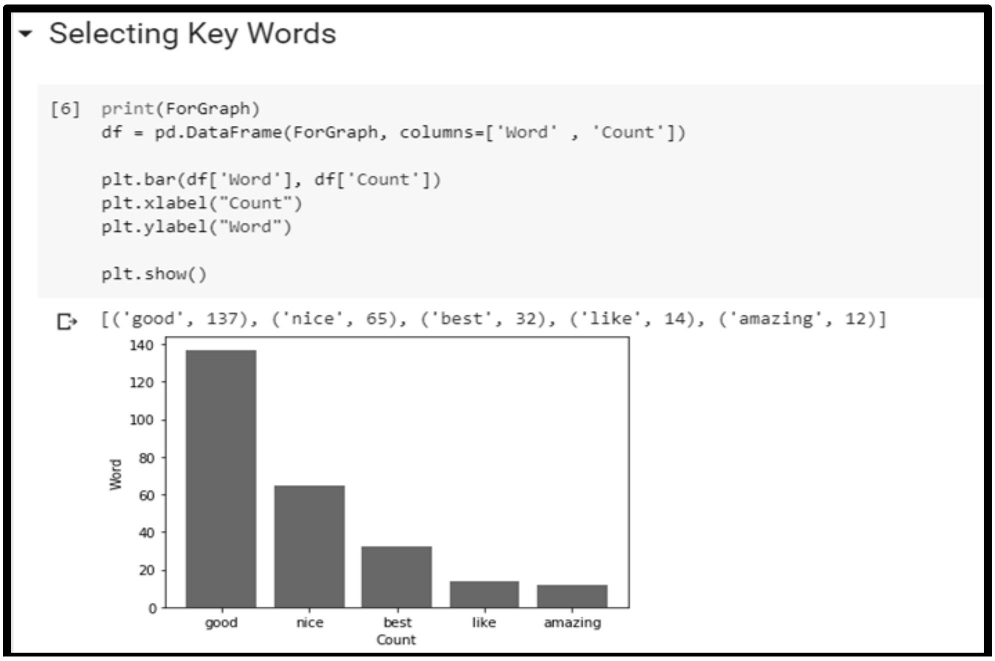


**Figure 16: Selecting** Keywords

- **Finding Rank of Keywords:** In our examination function as talked about in the past section, we found a couple of catchphrases which are habitually utilized in the greater part of the clients' surveys by applying a sack of word model, to change over key data from clear cut to mathematical it is profoundly essential to keep up with suitable weightage of every item audit we accumulated. As we would see it while separating information from the web one item could have 80 quantities of audits and another item could have 2200 surveys and subsequently to give equivalent weightage to each catchphrase, we took normal mean of the quantity of audits and afterward positioned watchword likewise, which at last proselyte downright information to mathematical. Figure portrays the code execution in python script for separating the positive word count by taking out the positive word went before by natation.

▾ Filtering positive word count

```
[ ] awesomeCount = 0
    bestCount = 0
    goodCount = 0
    niceCount = 0
    likeCount = 0
    pcount =[]
    for item in most_occur:
      if ((item[0]) == 'awesome'):
        awesomeCount = (item[1]-strobj.count("not awesome"))/dataset['Review'].count()
      if ((item[0]) == 'best' ):
        bestCount = (item[1]-strobj.count("not best"))/dataset['Review'].count()
      if ((item[0]) == 'good'):
        goodCount = (item[1]-strobj.count("not good")+strobj.count("not bad"))/dataset['Review'].count()
      if ((item[0]) == 'nice'):
        niceCount = (item[1]-strobj.count("not nice"))/dataset['Review'].count()
      if ((item[0]) == 'like'):
        likeCount = (item[1]-strobj.count("dont like"))/dataset['Review'].count()

    pcount = [('awesome',awesomeCount), ('like',likeCount), ('good',goodCount), ('best',bestCount), ('nice',niceCount)]
    # pcount
```

**Figure 17:** Filtering positive word count

- Following is the pictorial representation of find keyword and given them rank form the row data that is reviewed.

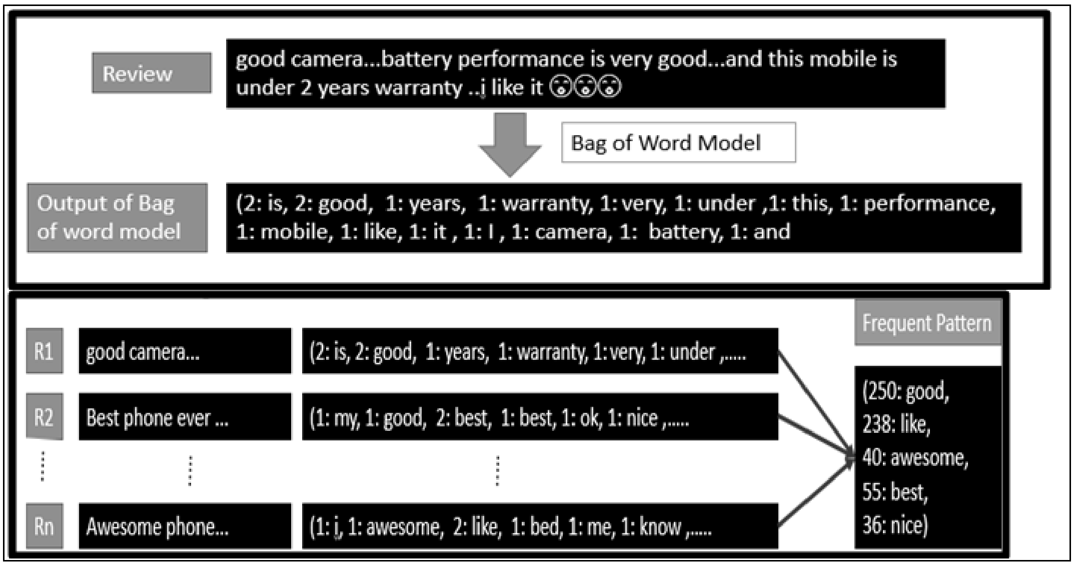


**Figure 18 :** Process of finding Word Rank

- **Data Transformation:** We are planning and organizing an expert table for every item which contains chosen catchphrases with its weightage and rate with its weightage. This expert table for every item will be a contribution for the following period of suggestion and expectation. Following figure shows the coding work to create the expert document and afterward send out it into a csv design.

Forming master table

```
[ ]  Rating = [1,2,3,4,5]
     RatingCount = [31,9,59,234,527]
     MasterTable= pd.DataFrame(wordCountTable)
     MasterTable["Rating"] = Rating
     MasterTable["RatingCount"] = RatingCount

     MasterTable
```

| | Word | Count | Rating | RatingCount |
|---|---|---|---|---|
| 0 | awesome | 0.000000 | 1 | 31 |
| 1 | like | 0.009333 | 2 | 9 |
| 2 | good | 0.087333 | 3 | 59 |
| 3 | best | 0.021333 | 4 | 234 |
| 4 | nice | 0.043333 | 5 | 527 |

**Figure 19:** Forting Master Table

Export Final Matix of Product

```
[ ]  MasterTable.to_csv("/content/Whirlpool7kgSemiAutomaticTopLoad8999_FinalMatrix.csv")
```

**Figure 20:** Exporting final matrix of product in CSV file

- **Recommendation Algorithm :** In experimental procedure, after data collection and preprocessing, it is predicting or recommending the product a user will be interested in or most likely to purchase. For the final recommendation we have prepared a product final matrix from the learning or pre-processing phase and given it as an input to the steepest ascent hill climbing algorithm to predict the final product recommendation. Fig. 21 illustrates the steps of including the recommendation algorithm

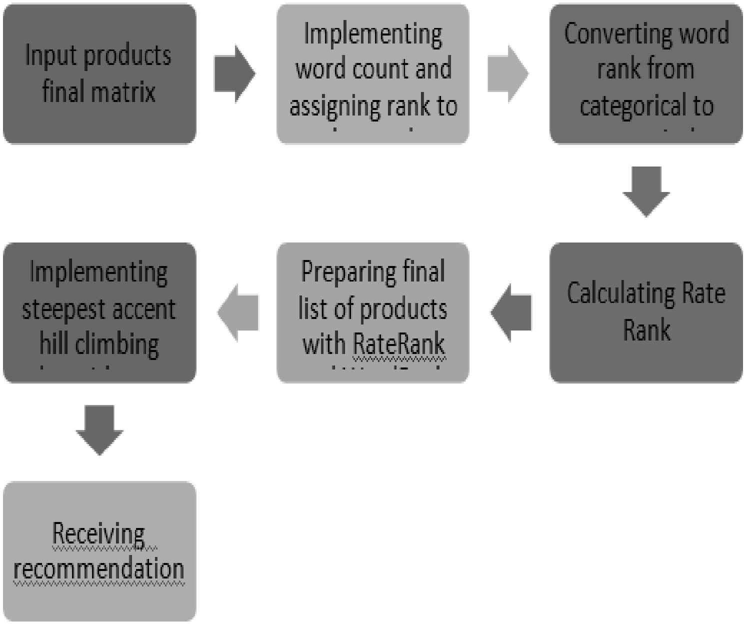


**Figure 21 :** Recommendation/ Prediction Workflow

- **Importing necessary library files:** To work with the information produced in past stage we are making new python script in which we are bringing in pandas and numpy library documents to create a diagram and structure information in plain organization, we are likewise utilizing and bringing in records Google colab to manage documents in cradle region which save document in support for next 12 hours. Figure Fig. 22 shows the panda, numpy and Google collab records getting imported.


**Figure 22:** Importing pandas and Google File

- **Reading data from the CSV files:** In the proposal stage, we took more than one different item's subtleties created in the past stage and chipped away at them for additional handling and emerging with the most appropriate suggested item out of given items. Figure Fig. 23 shows the perusing of three distinct models' information perusing.

## Reading CSV file

```
[2] m1 = pd.read_csv("/content/Whirlpool6.5kgFullyAutomaticTopLoad14690_FinalMatrix.csv")
    m2 = pd.read_csv("/content/Whirlpool7.2kgSemiAutomaticTopLoad11490_FinalMatrix.csv")
    m3 = pd.read_csv("/content/Whirlpool7kgSemiAutomaticTopLoad8999_FinalMatrix.csv")

    print(m1)
    print(m2)
    print(m3)
```

```
   Unnamed: 0     Word     Count  Rating  RatingCount
0           0  awesome  0.000000       1           40
1           1     like  0.000000       2           17
2           2     good  0.193333       3           94
3           3     best  0.000000       4          334
4           4     nice  0.047500       5          715
   Unnamed: 0     Word     Count  Rating  RatingCount
0           0  awesome  0.000000       1           11
1           1     like  0.000000       2            5
2           2     good  0.146341       3           29
3           3     best  0.000000       4          118
4           4     nice  0.070732       5          247
   Unnamed: 0     Word     Count  Rating  RatingCount
0           0  awesome  0.000000       1           31
1           1     like  0.000000       2            9
2           2     good  0.140698       3           59
3           3     best  0.026744       4          234
4           4     nice  0.053488       5          527
```

**Figure 23:** Reading CSV containing final product matrix

- **Giving Rank to the word:** It is vital to give rank to the word to do bound together examination among the various items. In the wake of tracking down watchwords by applying a pack of word model and example distinguishing and finding weighted normal mean in the past stage, we are currently giving fitting position to the catchphrase. This cycle assists in finding with phrasing rank, which changes over straight out information into mathematical information. Given figure Fig. 24 shows the code of tracking down position of the watchword and changing over it from clear cut to mathematical

## Filtering positive word count

```
[ ] awesomeCount = 0
    bestCount = 0
    goodCount = 0
    niceCount = 0
    likeCount = 0
    pcount =[]
    for item in most_occur:
      if ((item[0]) == 'awesome'):
        awesomeCount = (item[1]-strobj.count("not awesome"))/dataset['Review'].count()
      if ((item[0]) == 'best' ):
        bestCount = (item[1]-strobj.count("not best"))/dataset['Review'].count()
      if ((item[0]) == 'good'):
        goodCount = (item[1]-strobj.count("not good")+strobj.count("not bad"))/dataset['Review'].count()
      if ((item[0]) == 'nice'):
        niceCount = (item[1]-strobj.count("not nice"))/dataset['Review'].count()
      if ((item[0]) == 'like'):
        likeCount = (item[1]-strobj.count("dont like"))/dataset['Review'].count()

    pcount = [('awesome',awesomeCount), ('like',likeCount), ('good',goodCount), ('best',bestCount), ('nice',niceCount)]
    # pcount
```

**Figure 24 :** Ranking word

## Function to find Word Rank and convert it from categorial to numerical

```
[ ] def WordRank(word):
      if(word == WR_Word1):
        return(5)
      if(word == WR_Word2):
        return(4)
      if(word == WR_Word3):
        return(3)
      if(word == WR_Word4):
        return(2)
      if(word == WR_Word5):
        return(1)
```

**Figure 25:** Function to find word rank

- **Preparing Product Final Matrix :** All the while, further we amass the nitty gritty audit and rating of the item, we orchestrate the rating by computing the amount of the specific rating given by the clients which is a simple assignment and afterward we work on the survey. As referenced in the above clarification, by changing straight out data over completely to mathematical we appoint rank to every catchphrase and organize them appropriately. This will create the last lattice of the singular item containing its significance by rating and audit wise in the eventual outcome framework. Following figure is only a gander at the pre-arranged information which will create the eventual outcome network on which the Steepest-Rising Slope climbing calculation is applied to track down the best reasonable item to suggest.

```
********************************
Model 1 Rank List  R1 (5, 4, 3, 1, 2)
Model 1 Word List  R2 (5, 4, 1, 2, 3)
********************************
Model 2 Rank List  R1 (5, 4, 3, 1, 2)
Model 2 Word List  R2 (5, 4, 1, 2, 3)
********************************
Model 3 Rank List  R1 (5, 4, 3, 1, 2)
Model 3 Word List  R2 (5, 4, 3, 1, 2)
********************************
```

**Figure 26 :** Product Final Matrix

- **Steepest-Ascent Hill climbing algorithm:** Heuristic hunt utilized by the Slope Moving for numerical improvement tangles in the field of Man-made brainpower. In Steepest-Rising Slope climbing procedure, prior to choosing any hub, it initially reviews every one of the adjoining hubs and afterward chooses the hub closest to the objective state as of next hub. In our calculation we utilized a similar strategy to find the objective state

```
[ ]  ans=p[0]
     print(ans.Name)
     for i in range(1,3):
       for j in range(0,5):
         if (ans.r1[j] > p[i].r1[j]):
           break
         if (ans.r1[j] < p[i].r1[j]):
           ans=p[i]
           break
         else:
           if(ans.r2[j] > p[i].r2[j]):
             break
           if(ans.r2[j] < p[i].r2[j]):
             ans = p[i]
             break
     print(ans.Name)

     m1
     m3
```

**Figure 27:** Steepest Ascent Hill Climbing Algorithm – I

In additional progression, when we have the eventual outcome framework of beyond what one item we can then give them as a contribution to our Steepest-Rising Slope climbing calculation. This model is an upgraded rendition of a man-made brainpower calculation named steepest rising slope climbing calculation. In steepest rising slope climbing calculation we can find for the best adjoining hub which is closer to the objective state lastly we get the most ideal that anyone could hope to find hub, this is the way we could suggest the best item out of given items, in upgrade our created Superior Steepest-Climb Slope climbing calculation, we give level of proposal and give opportunity to the client to choose in light of the degree of given suggestion. Given figure portrays the code execution for the Steepest-Rising Slope climbing calculation - II.

```
[ ]  for i in range(0,2):
       for j in range(i+1,3):
         for k in range(0,5):
           if (p2[i].r1[k] > p2[j].r1[k]):
             break
           if (p2[i].r1[k] < p2[j].r1[k]):
             temp = p2[i]
             p2[i] = p2[j]
             p2[j] = temp
             break
           else:
             if(p2[i].r2[k] > p2[j].r2[k]):
               break
             if(p2[i].r2[k] < p2[j].r2[k]):
               temp = p2[i]
               p2[i] = p2[j]
               p2[j] = temp
               break

     for i in range (0,3):
       print(p2[i].Name)
```

**Figure 28:** Improved Steepest Ascent Hill Climbing Algorithm – II

## V. RESULT

In the previously mentioned trial method we have recognized six unique items in the electronic portion and we have isolated items under the class of various brands, consequently under each brand we have gathered information for something like three distinct models and work on that. In the last endeavor we gathered information for example clients' survey and rating of various 114 items which can be isolated into 38 distinct models and absolute audits and appraisals gathered was 75026. In the wake of doing pre-handling on accumulated information and further passing it to the proposal or expectation stage we got results to suggest the item at various levels which further passed as contribution to Further developed Steepest Rising Slope Climbing calculation to track down the verification of certainty.

1. **Evaluation Process:** The suggestion framework created has been assessed by two unique Confirmations of Idea Evidences of Idea were tried on the created proposal framework with various speculations and tried against the genuine outcome [28].

- **Proof of Concept**

  - POC 1: On the off chance that rating and positioning of the item copies in the last level while playing out the suggestion calculation as in at the same time removing information from the web based business gateway making the calculation arrive at a bottleneck circumstance.

The rating position of two item copies then the algorithmic iterated, the following outcome will upgrade the proposal followed by the powerful hunt

- POC2: On the off chance that positioning and rating deception perseveres in all degrees of looking through the most noteworthy position, while playing out the suggestion calculation as in at the same time removing information from the web based business entryway making the calculation arrive at a bottleneck circumstance. In the event that the item is copied in all levels, the calculation is moved for the following element extraction.


**Figure 29:** Proof of Concept Justification

2. **Performance evaluation:** Extensive variety of accessible measurements are utilized to gauge the precision of the created proposal framework. Two characterized classifications of measurements used to gauge the proposal quality and assess the exactness of the suggestion framework. RMSE, MAE, RAE are among the couple of famous expectation measurements to assess the quality and precision of the created suggestion framework.

Random tree algorithm has been tested against the generated result from the recommendation system

**Table 2 : Performance Evaluation**

| Criteria | Unit / Result |
|---|---|
| Cross validation fold | 10 |
| Correctly classified instances | **94.8148%** |
| Incorrectly classified instances | 5.1852% |
| Kappa statistic | 0.9222 |
| MAE | 0.0352 |
| RMSE | 0.1868 |
| RAE | 7.9189% |
| RRSE | 39.6186% |

Root Mean Square Mistake (RSME) is a variation of Mean Square Blunder (MSE), and prior to adding a mistake it squares the blunders. Root Mean Square Mistake technique has been utilized oftentimes as a presentation evaluator. The suggested and noticed rating distinction is handled through this technique. In our noticed outcome in Weka 3.8.4 the RSME is 0.1868. MEA is Mean Outright Blunder gauges, the contrast among suggestion and real evaluating and its outright standard deviation is estimated by MEA. The understanding and execution of this strategy is extremely simple and consequently it is utilized most often. As we would see it the MEA given is 0.0352 [29].

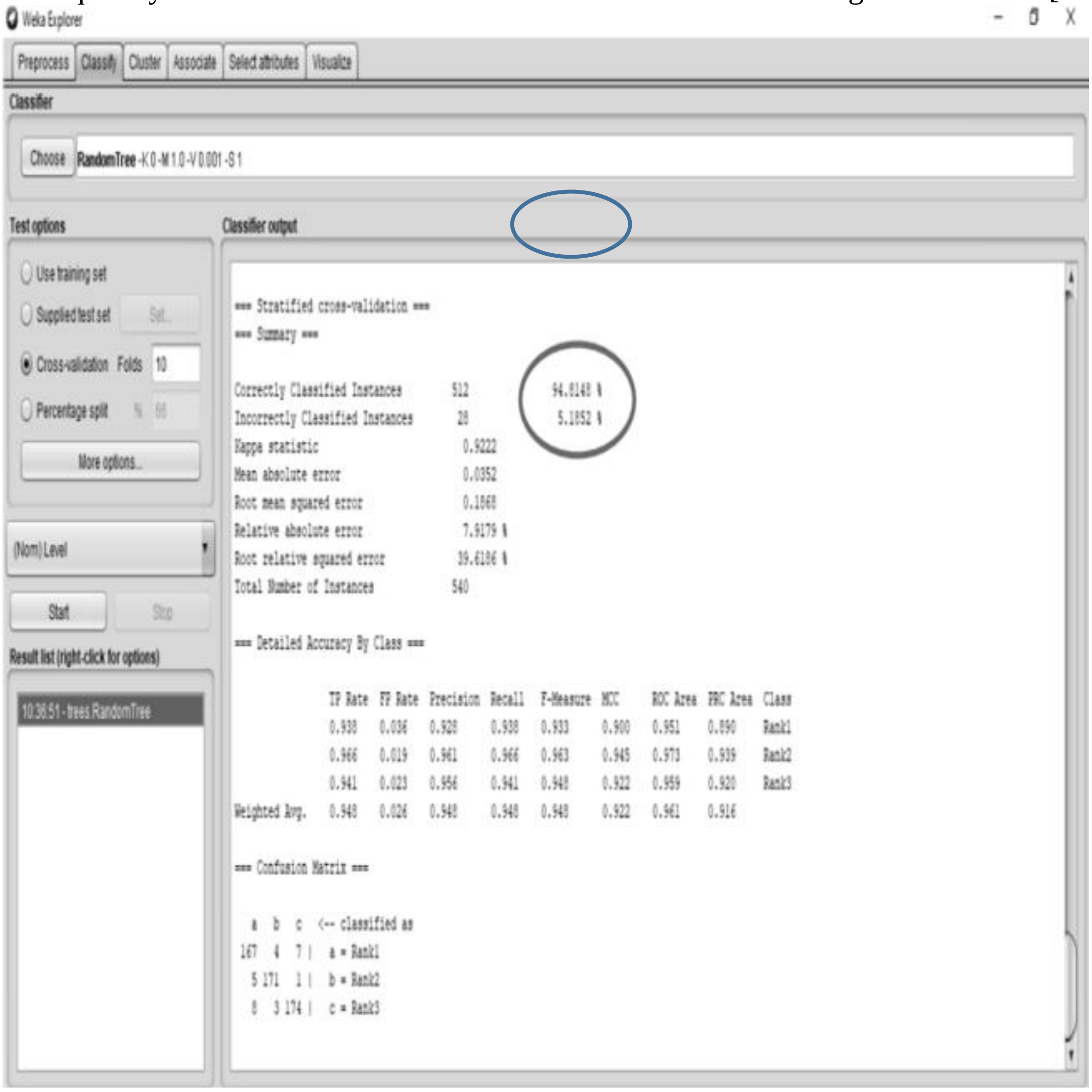


**Figure 30:** Weka screen shot - performance evaluation

## VI. CONCLUSION AND FUTURE SCOPE

This examination impelled the item suggestion framework for the internet business site, which prescribes the probably going to buy item or item at various degree of proposal to the client utilizing Steepest-Rising Slope climbing and Further developed Steepest-Rising Slope climbing calculation separately and content-based sifting, in light of the clients' evaluating and survey. For the clients' comfort we created scripts and scratched information straightforwardly from the online business sites. NLP is utilized to handle clients' audits and concentrate applicable data from the surveys, further in view of the rating and surveys both on the whole we have created two calculations specifically Steepest-Rising Slope climbing and Further developed Steepest-Rising Slope climbing. Our calculation has a restriction of the virus start issue in the suggestion framework. For the recently sent off item or the item which has no surveys and rating given from the client isn't canvassed in this examination. For the future extension we could zero in on the virus start issue in the suggestion framework with our proposed Superior Steepest-Rising Slope climbing calculation.

## REFERENCES

[1] E. Turban, D. King, J. K. Lee, T.-P. Liang, and D. Turban, Electronic Commerce: a Managerial and social Networks Perspective 2012. 2012.
[2] T. Garín-Muñoz, R. López, T. Pérez-Amaral, I. Herguera, and A. Valarezo, "Models for individual adoption of eCommerce, eBanking and eGovernment in Spain," Telecomm. Policy, vol. 43, no. 1, pp. 100–111, 2019, doi: 10.1016/j.telpol.2018.01.002.
[3] J. Jooa, S. Bangb, and G. Parka, "Implementation of a Recommendation System using Association Rules and Collaborative Filtering," Procedia - Procedia Comput. Sci., vol. 91, no. Itqm 2016, pp. 944–952, 2019, doi: 10.1016/j.procs.2016.07.115.
[4] M. Narayanan and A. Kumar, "IMR-DCAL SPECIAL PAPER A study and analysis of recommendation systems for location-based social network ( LBSN ) with big data," IIMB Manag. Rev., vol. 28, no. 1, pp. 25–30, 2016, doi: 10.1016/j.iimb.2016.01.001.
[5] L. M. De Campos, J. M. Fernández-luna, J. F. Huete, and M. A. Rueda-morales, "International Journal of Approximate Reasoning Combining content-based and collaborative recommendations : A hybrid approach based on Bayesian networks q," Int. J. Approx. Reason., vol. 51, no. 7, pp. 785–799, 2010, doi: 10.1016/j.ijar.2010.04.001.
[6] C. Y. Li and Y. C. Ku, "The power of a thumbs-up: Will e-commerce switch to social commerce?," Inf. Manag., vol. 55, no. 3, pp. 340–357, 2018, doi: 10.1016/j.im.2017.09.001.
[7] N. Boysen, R. de Koster, and F. Weidinger, "Warehousing in the e-commerce era: A survey," Eur. J. Oper. Res., vol. 277, no. 2, pp. 396–411, 2019, doi: 10.1016/j.ejor.2018.08.023.
[8] P. D. I. Hernando, "Electronic commerce," Comput. Law Secur. Rev., vol. 19, no. 5, pp. 363–374, Sep. 2003, doi: 10.1016/S0267-3649(03)00504-1.
[9] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Knowledge-Based Systems Recommender systems survey," Knowledge-Based Syst., vol. 46, pp. 109–132, 2013, doi: 10.1016/j.knosys.2013.03.012.
[10] R. Katarya and O. P. Verma, "An effective collaborative movie recommender system with cuckoo search," Egypt. Informatics J., vol. 18, no. 2, pp. 105–112, 2017, doi: 10.1016/j.eij.2016.10.002.
[11] F. O. Isinkaye, "Recommendation systems : Principles , methods and evaluation," pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.
[12] B. Vora Assit, N. J. Patel Professor, A. Motibhai Patel, N. N. Jani Dean, and G. Dave Assit, "Review Paper introducing E-commerce, Data Mining and Pattern Recognition to recommend the product," Res. HUB-International Multidiscip. Res. J., no. 03, 2018.
[13] S. Redhu, S. Srivastava, B. Bansal, and G. Gupta, "Sentiment Analysis Using Text Mining : A Review," Int. J. Data Sci. Technol., vol. 4, no. 2, pp. 49–53, 2018, doi: 10.11648/j.ijdst.20180402.12.
[14] Y. Zhao, G. Kou, Y. Peng, and Y. Chen, "Understanding influence power of opinion leaders in e-commerce networks: An opinion dynamics theory perspective," Inf. Sci. (Ny)., vol. 426, pp. 131–147, 2018, doi: 10.1016/j.ins.2017.10.031.
[15] P. D. I. Hernando, "Electronic commerce," Comput. Law Secur. Rev., vol. 19, no. 5, pp. 363–374, Sep. 2003, doi: 10.1016/S0267-3649(03)00504-1.
[16] L. Labuschagne and J. Eloff, "E-commerce strategy formulation."
[17] Y. Guo, M. Wang, and X. Li, "Application of an improved Apriori algorithm in a mobile e-commerce recommendation system," Ind. Manag. Data Syst., 2017, doi: 10.1108/IMDS-03-2016-0094.
[18] P. Mehta et al., "A high-bias , low-variance introduction to Machine Learning for physicists," Phys. Rep., vol. 810, pp. 1–124, 2019, doi: 10.1016/j.physrep.2019.03.001.
[19] X. M. T. Vu et al., "A Shared Vision for Machine Learning in Neuroscience," J. Neurosci., vol. 38, no. 7, pp. 1601–1607, 2018, doi: 10.1523/JNEUROSCI.0508-17.2018.

[20] F. Hutter, Automated Machine Learning. .
[21] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," Expert Syst. Appl., vol. 92, pp. 507–520, 2018, doi: 10.1016/j.eswa.2017.09.058.
[22] P. Alencar and D. Cowan, "The use of machine learning algorithms in recommender systems : A systematic review," Expert Syst. Appl., vol. 97, pp. 205–227, 2018, doi: 10.1016/j.eswa.2017.12.020.
[23] T. Cunha, C. Soares, and A. C. P. L. F. De Carvalho, "Metalearning and Recommender Systems : A literature review and empirical study on the algorithm selection problem for Collaborative Filtering," Inf. Sci. (Ny)., vol. 423, pp. 128–144, 2018, doi: 10.1016/j.ins.2017.09.050.
[24] S. Sivapalan, A. Sadeghian, H. Rahnama, and A. M. Madni, "Recommender systems in e-commerce," World Autom. Congr. Proc., pp. 179–184, 2014, doi: 10.1109/WAC.2014.6935763.
[25] C. S. Echt and C. D. Nelson, "Linkage mapping and genome length in eastern white pine (Pinus strobus L.)," Theor. Appl. Genet., vol. 94, no. 8, pp. 1031–1037, 1997, doi: 10.1007/s001220050511.
[26] K. Zhang, R. P. Lively, C. Zhang, W. J. Koros, and R. R. Chance, "Investigating the Intrinsic Ethanol/Water Separation Capability of ZIF-8: An Adsorption and Diffusion Study," J. Phys. Chem. C, vol. 117, no. 14, pp. 7214–7225, Apr. 2013, doi: 10.1021/jp401548b.
[27] R. Bell, Y. Koren, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer (Long. Beach. Calif)., vol. 42, no. 08, pp. 30–37, 2009, doi: 10.1109/MC.2009.263.
[28] C. K. Pereira, F. Campos, V. Ströele, J. M. N. David, and R. Braga, "BROAD-RSI – educational recommender system using social networks interactions and linked data," J. Internet Serv. Appl., vol. 9, no. 1, 2018, doi: 10.1186/s13174-018-0076-5.
[29] D. Agalya and V. Subramaniyaswamy, "Group-Aware Recommendation using Random Forest Classification for Sparsity Problem," Indian J. Sci. Technol., vol. 9, no. 48, pp. 1–10, 2016, doi: 10.17485/ijst/2016/v9i48/107960.