**A Project Report on**

# Product Recommendation using Feature-Level Analysis

submitted in partial fulfillment for the award of

**Bachelor of Technology**

in

**Computer Science & Engineering**

by

**B. Pranavi (Y20ACS535)** **R. Likhitha (L21ACS417)**

**Sk. Abdul Gouse (Y20ACS557)** **R. Mahathi (Y20ACS551)**

Under the guidance of
**Mr. P. Nanda Kishore.**
Assistant Professor
Department of Computer Science and Engineering
**Bapatla Engineering College**
(Autonomous)
(Affiliated to Acharya Nagarjuna University)
**BAPATLA – 522 102, Andhra Pradesh, INDIA**
**2023-2024**

# Department of

# Computer Science and Engineering

## CERTIFICATE

This is to certify that the project report entitled **Product Recommendation using Feature-Level Analysis** that is being submitted by B. Pranavi (Y20ACS535), R. Likhitha (L21ACS417), Sk. Abdul Gouse (Y20ACS557), R. Mahathi(Y20ACS551) and in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

| | |
|---|---|
| **Signature of the Guide** | **Signature of the HOD** |
| **P. Nanda Kishore** | **Dr. M. Rajesh Babu** |
| **Assistant. Prof** | **Assoc. Prof. & Head** |

# DECLARATION

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

B. Pranavi (Y20ACS535)

R. Likhitha (L21ACS417)

Sk. Abdul Gouse (Y20ACS557)

R. Mahathi (Y20ACS551)

# ACKNOWLEDGEMENT

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **P. Nanda Kishore**, Assistant. Prof. Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Assoc. Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

B. Pranavi (Y20ACS535)

R. Likhitha (L21ACS417)

Sk. Abdul Gouse (Y20ACS557)

R. Mahathi (Y20ACS551)

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

In this project, we present a systematic approach to recommending products based on a comprehensive assessment of customer preferences and feedback. The methodology involves a multi-step user interface where customers answer boolean questions to determine their interest, select desired product features, and provide opinions through reviews or ratings. The collected data is organized into a structured dataset, which undergoes preprocessing to address missing values and prepare textual reviews for sentiment analysis using the VADER algorithm. A novel combined metric is derived, integrating sentiment scores and feature ratings, to classify product features into categories (strong, moderate, weak) using K-means clustering. The final recommendation decision is made based on the balance of strong and moderate features relative to the total selected features. This approach offers a systematic framework for personalized product recommendations informed by customer sentiment and feature preferences

# 1 INTRODUCTION

In today's competitive marketplace, personalized product recommendations play a pivotal role in enhancing customer satisfaction and driving sales. This project aims to develop an intelligent recommendation system that leverages customer feedback and preferences to suggest products tailored to individual needs. The methodology involves a structured interface where customers engage in a series of interactions: from initial assessment of interest through boolean questions, to selection of desired features, and finally, providing opinions via reviews or ratings.

The core of this project lies in the effective utilization of user-generated data. By collecting and organizing customer inputs into a dataset, we enable systematic analysis and processing to derive valuable insights. A key challenge addressed in this project is the handling of missing data and the preprocessing of textual reviews to prepare them for sentiment analysis.

To gauge customer sentiment effectively, the VADER (Valence Aware Dictionary and sEntiment Reasoner) algorithm is employed to assign sentiment scores to textual reviews. These scores, combined with feature ratings, form a novel metric used to categorize product features into distinct groups (strong, moderate, weak) using K-means clustering. This categorization provides a nuanced understanding of feature significance based on customer sentiment and feedback.

Ultimately, the recommendation decision is driven by the balance and strength of features selected by the customer. If the combined count of strong and moderate features surpasses a certain threshold relative to the total features selected, a product is recommended as strong; otherwise, it is categorized as weak.

This project not only showcases a practical application of data-driven decision-making in product recommendations but also highlights the value of customer-centric design in enhancing the overall user experience.

## 1.1. Problem Statement and objective

Generating recommendation for products based on a combination of customer reviews and feature ratings presents challenges in data collection, processing, and analysis. Ensuring accurate classification of product strength and providing meaningful recommendations requires effective handling of missing data, sentiment analysis of customer reviews, and feature classification.

## 1.2. Technology Background

### 1.2.1. Machine Learning using Python

Machine learning enables systems to learn from data and make predictions or decisions without being explicitly programmed. Python has emerged as one of the most popular programming languages for machine learning due to its simplicity, extensive libraries, and vibrant community support.

### 1.2.2. Libraries and Algorithms of project

**Pandas**

Pandas is a powerful Python library used for data manipulation and analysis. It offers data structures and functions to work with structured data, making it easier to clean, transform, and analyze datasets.

### Tkinter

Tkinter is the standard GUI (Graphical User Interface) toolkit for Python. It provides a set of built-in widgets and functions for creating desktop applications with graphical interfaces.

### Scikit-learn (sklearn)

Scikit-learn is a popular machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It offers a wide range of algorithms for classification, regression, clustering, dimensionality reduction, and more.

### NLTK (Natural Language ToolKit)

NLTK is a leading platform for building Python programs to work with human language data. It provides tools and resources for tasks such as tokenization, stemming, tagging, parsing, and semantic reasoning.

### VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER is a rule-based sentiment analysis tool specifically designed for analyzing sentiment in text data. It quantifies the sentiment of text documents into positive, negative, or neutral categories.

### K-means Algorithm

K-means is a popular clustering algorithm used to partition a dataset into a predetermined number of clusters. It aims to minimize the sum of squared distances between data points and their respective cluster centroids.

## 1.3. Runtime Environment – Jupyter notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It supports over 40 programming languages, including Python, R, Julia, and Scala, making it a versatile tool for interactive computing and data analysis.

It is a versatile tool that facilitates interactive computing, data analysis, and collaborative work, making it popular among data scientists, researchers, educators, and developers.

# 2 LITERATURE SURVEY

**Feature-level Rating System using Customer Reviews and Review Votes** Publication year: 2020, Authours: Koteswar Rao Jerripothula, Member, IEEE, Ankit Rai, Kanu Garg, and Yashvardhan Singh Rautela. This work studies how we can obtain feature level ratings of the mobile products from the customer reviews and review votes to influence decision making, both for new customers and manufacturers. Such a rating system gives a more comprehensive picture of the product than what a product-level rating system offers. While product-level ratings are too generic, feature-level ratings are particular; we exactly know what is good or bad about the product. There has always been a need to know which features fall short or are doing well according to the customers perception. It keeps both the manufacturer and the customer well-informed in the decisions to make in improving the product and buying, respectively. Different customers are interested in different features. Thus, feature-level ratings can make buying decisions personalized. This work analyze the customer reviews collected on an online shopping site (Amazon) about various mobile products and the review votes. Explicitly, it carry out a feature-focused sentiment analysis for this purpose.Eventually, this analysis yields ratings to 108 features for 4k+ mobiles sold online. It helps in decision making on how to improve the product (from the manufacturers perspective) and in making the personalized buying decisions (from the buyers perspective) a possibility. This analysis has applications in recommender systems, consumer research, etc.

# 3 PROPOSED SYSTEM

The proposed system is an intelligent product recommendation platform that leverages customer feedback and sentiment analysis to deliver personalized product suggestions. The system comprises several key components designed to engage customers, analyze their preferences, and generate tailored recommendations:

## 3.1 Various modules in the project

### 3.1.1User Interface and Engagement

Through the proposed system, we design and implement an intuitive interface to collect Boolean - type responses from customers to assess their authenticity. Once the credibility assessment is success, the customer will be redirected to next interface

### 3.1.2 Feature Selection and Feedback Collection

Customers will be able to select desired product features from a curated list and then they can provide feedback for selected features through textual reviews or numerical ratings. Once customer provides his feedback the data is replicated into a new dataset for further analysis.

### 3.1.3 Data Processing and Sentiment Analysis

Robust data preprocessing techniques will be implemented to handle missing data and clean textual inputs (reviews) .Sentiment analysis using the VADER algorithm will be employed to analyze textual reviews and derive sentiment scores that reflect customer sentiment towards specific product features

### 3.1.4 Feature Importance Assessment and Categorization

Sentiment scores from reviews will be integrated with feature ratings to derive a combined metric that quantifies feature importance based on customer sentiment. Clustering algorithms such as K-means will categorize product features into groups (strong, moderate, weak) based on the combined metric, facilitating data-driven decision-making in recommendations.

### 3.1.5 Recommendation Generation

The system will generate recommendation utilizing categorized feature groups and customer preferences to generate personalized product recommendations. Decision rules will be implemented to determine recommendation outcomes (e.g., strong recommendation vs. weak recommendation) based on the balance and strength of categorized features selected by the customer.

## 3.2 Dataset

Initially to enable feature selection through customers, we have formed a datastet which simply contains various features of a product.

Here, we have considered around 40 features of a mobile phone to be incorporated in the dataset.

**Table 3.1 Initial Dataset**

| S.no. | Feature | S.no | Feature |
|---|---|---|---|
| 1 | charger | 21 | touch |
| 2 | camera | 22 | price |
| 3 | battery | 23 | sim |
| 4 | screen | 24 | permissions |
| 5 | apps | 25 | notifications |
| 6 | android | 26 | updates |
| 7 | services | 27 | software |
| 8 | size | 28 | multitasks |
| 9 | appearance | 29 | brightness |
| 10 | calls | 30 | music |
| 11 | sound | 31 | network |
| 12 | picture | 32 | buttons |
| 13 | warranty | 33 | features |
| 14 | ram | 34 | heat |
| 15 | design | 35 | waterproof |
| 16 | storage | 36 | video |
| 17 | speed | 37 | games |
| 18 | hardware | 38 | recordings |
| 19 | compatibility | 39 | gestures |
| 20 | sensors | 40 | gps |

After module 2, we'll be provided with actual dataset that can be used for further analysis. The dataset formed consists of the selected features along with their corresponding reviews or ratings.

**Table 3.2 Generated Dataset After Feedback Collection**

| Feature | Review | Rating |
|---|---|---|
| camera | | 4 |
| battery | takes lot of time to charge. not recommended | |
| screen | | 5 |
| services | | 3 |
| sound | Good audio quality | |
| picture | | 5 |
| ram | High gb ram at this cost | |
| design | | 5 |
| storage | | 3 |
| speed | High speed. Best for general use | |
| hardware | Good hardware is used may not get easily damaged. | |
| updates | | 4 |
| software | Better software compatible for many applications | |

## 3.3 Preprocessing

As the dataset comprises of missing data, it need to be handled.To handle the missing data, pandas libraries functions 'fillna' is used accordingly. Also to clean the textual reviews and to prepare them for sentiment analysis, the preprocessing is done with the help of nltk library in python. After preprocessing, the preprocessesd reviews are augumented to the dataset.

## 3.4 Sentiment Analysis

### 3.4.1 Overview

Sentiment analysis, also known as opinion mining, is the process of extracting and analyzing sentiment or emotion from text data. The goal of sentiment analysis is to determine the overall sentiment expressed in a piece of text, whether it's positive, negative, or neutral. This analysis is valuable for understanding public opinion, customer feedback, and social media sentiment.

#### 3.4.1.1 Key Components of Sentiment Analysis

**Text Preprocessing:**

Before sentiment analysis, text data undergoes preprocessing steps such as tokenization (breaking text into words or phrases), removing stop words (commonly used words that do not contribute much meaning), and stemming (reducing words to their base or root form).

**Sentiment Lexicons and Dictionaries:**

Sentiment analysis often relies on lexicons or dictionaries containing words annotated with sentiment scores (e.g., positive, negative, neutral). These lexicons assign polarity values to words based on their sentiment.

**Machine Learning Models:**

Machine learning techniques, such as supervised learning (classification) and unsupervised learning (clustering), can be employed for sentiment analysis. Supervised learning uses labeled data to train models to predict sentiment, while unsupervised learning identifies patterns and clusters in text data.

**Rule-based Approaches:**

Rule-based approaches use predefined rules and patterns to identify sentiment in text data. These approaches may include grammatical rules, linguistic patterns, and syntactic analysis.

#### 3.4.1.2 Applications of Sentiment Analysis

**Social Media Monitoring:** Sentiment analysis is widely used to analyze public opinion and sentiment expressed on social media platforms. It helps businesses understand customer perceptions and brand sentiment.

**Customer Feedback Analysis:** Sentiment analysis is used to analyze customer reviews, surveys, and feedback to gauge customer satisfaction and identify areas for improvement.

**Market Research:** Sentiment analysis provides insights into market trends, consumer preferences, and sentiment towards products or services, aiding in market research and decision-making.

### 3.4.2 VADER algorithm for sentiment analysis

#### 3.4.2.1 Overview

The VADER algorithm is a lexicon and rule-based approach designed specifically for sentiment analysis of text data. It was developed by C.J. Hutto and Eric Gilbert, VADER is widely used for its simplicity and effectiveness in capturing sentiment nuances in natural language.

### 3.4.2.2 Key components of VADER

**Lexicon-based Approach:**

VADER utilizes a sentiment lexicon that contains words annotated with sentiment scores (positive, negative, neutral). Each word in the lexicon is assigned a polarity value based on its sentiment intensity, ranging from -4 (extremely negative) to +4 (extremely positive), with 0 representing neutrality.

**Rule-based Sentiment Analysis:**

In addition to sentiment scores, VADER incorporates grammatical rules and linguistic features to analyze sentiment in text. It considers punctuation marks, capitalization, degree modifiers (e.g., "very", "extremely"), and emoticons to infer sentiment intensity.

**Handling Context and Negation:**

VADER is equipped to handle contextual nuances and negation in text. It can recognize changes in sentiment due to modifiers or qualifiers (e.g., "not good") and adjust sentiment scores accordingly, capturing the sentiment shift accurately.

**Scalability and Real-time Analysis:**

One of the key advantages of VADER is its computational efficiency and scalability. It can perform sentiment analysis in real-time on large volumes of text data, making it suitable for applications such as social media monitoring and customer feedback analysis.

#### 3.4.2.3 VADER Architecture

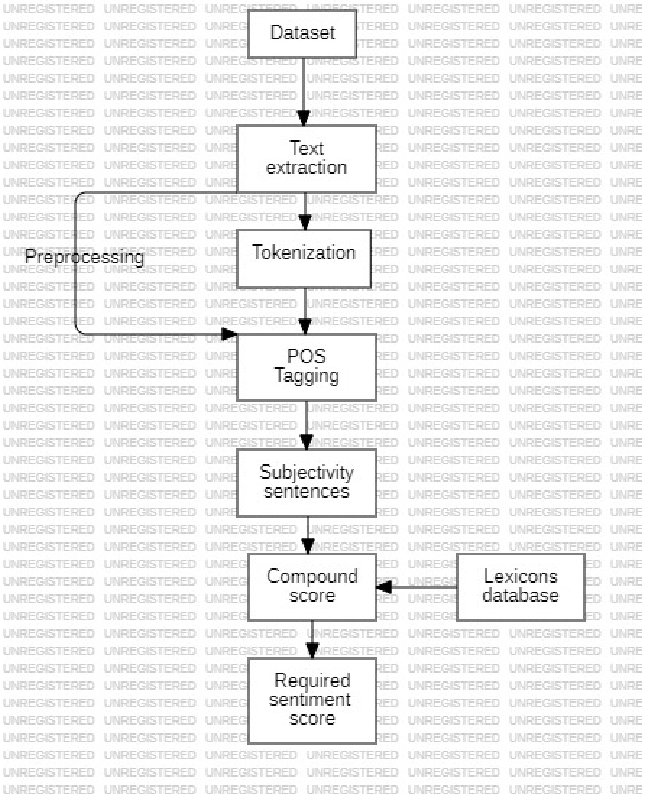


**Figure 3.1 VADER Algorithm Workflow**

## 3.5 K-means clustering

### 3.5.1 Overview

The K-means clustering algorithm is a popular unsupervised machine learning technique used for partitioning the data into K distinct, non-overlapping clusters based on similarity of data points.The algorithm aims to minimize the within-cluster sum of squared distances from each data point to its assigned cluster centroid. It is widely applied in various domains for data segmentation, pattern recognition, and data mining tasks.

### 3.5.2 Working of Kmeans algorithm

**Initialization:**

Choose the number of clusters (K) that the dataset should be divided into. Initialize K cluster centroids either randomly or based on predefined criteria (e.g., randomly selecting K data points as initial centroids).

**Assignment of Data Points to Clusters:**

Iterate through each data point in the dataset. For each data point, calculate the distance to each of the K cluster centroids using a distance metric such as Euclidean distance. Assign the data point to the cluster whose centroid is closest (i.e., has the minimum distance).

**Update Cluster Centroids:**

After assigning all data points to clusters, recalculate the centroids of the clusters based on the mean (average) of the data points assigned to each cluster. Move each cluster centroid to the new mean location calculated from the data points assigned to its cluster.

**Iterative Refinement:**

Repeat the assignment of data points to clusters and centroid updates iteratively until convergence criteria are met. Convergence is typically achieved when the cluster assignments and centroids stabilize (i.e., minimal change in cluster assignments between iterations).

### 3.5.3 Kmeans Architecture

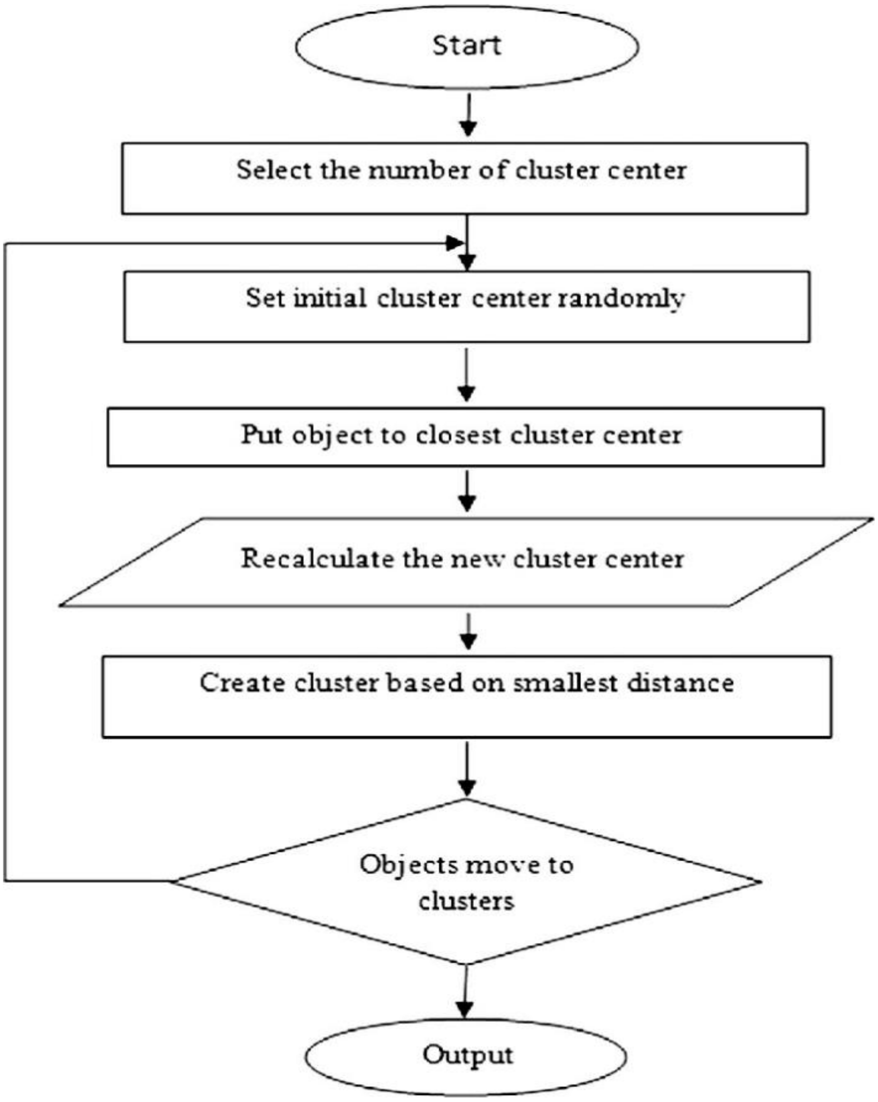

Start
Select the number of cluster center
Set initial cluster center randomly
Put object to closest cluster center
Recalculate the new cluster center
Create cluster based on smallest distance
Objects move to clusters
Output

**Figure 3.2 KMeans Architecture**

## 3.6 Proposed architecture

Customizing the above vader and kmeans the proposed architecture looks as follows:

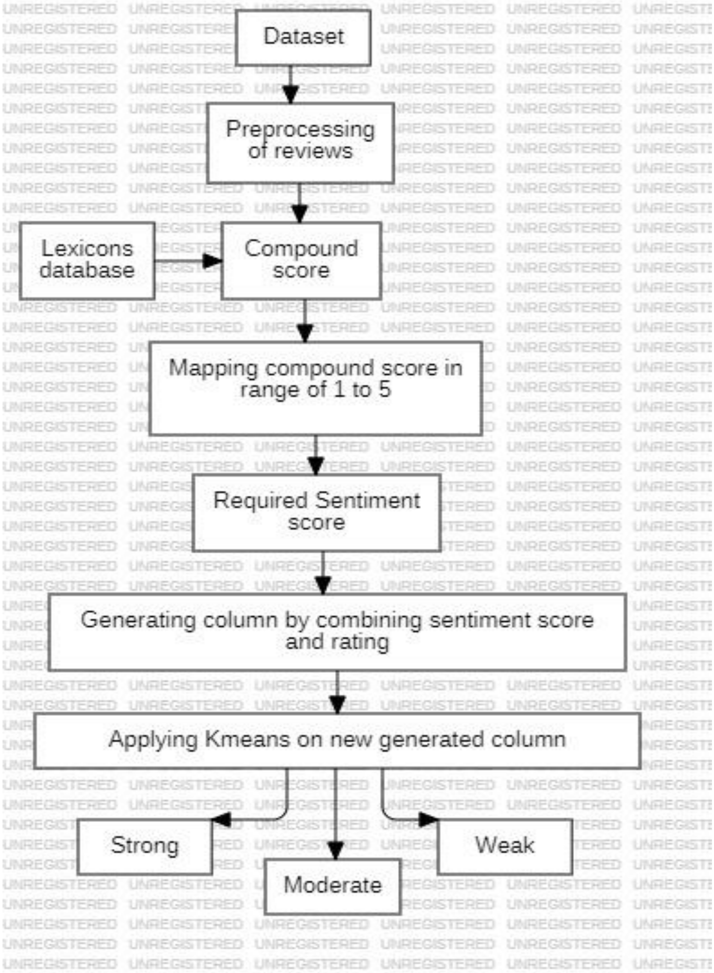

Dataset
Preprocessing of reviews
Lexicons database
Compound score
Mapping compound score in range of 1 to 5
Required Sentiment score
Generating column by combining sentiment score and rating
Applying Kmeans on new generated column
Strong
Moderate
Weak

**Figure 3.3 Proposed Architecture**

## 3.7 Advantages of Proposed system

**Personalized Recommendations:** The project aims to deliver personalized product recommendations based on customer preferences and sentiment analysis. This personalized approach can enhance user satisfaction and increase engagement by offering tailored suggestions aligned with individual preferences.

**Improved User Experience:** By incorporating customer feedback and sentiment analysis, the project can enhance the overall user experience. Customers will receive product recommendations that reflect their sentiments and preferences, leading to more meaningful interactions with the platform.

**Effective Handling of Heterogeneous Data:** The project implements robust data preprocessing techniques to handle heterogeneous data types such as reviews and ratings. This ensures data quality and reliability in the sentiment analysis process, leading to more accurate insights and recommendations.

**Real-time Recommendation Updates:** Leveraging efficient algorithms like VADER for sentiment analysis and K-means for clustering allows the project to perform real-time recommendation updates. This agility enables the system to adapt quickly to changing customer sentiments and preferences.

**Data-driven Decision Making:** By categorizing product features based on sentiment and feedback, the project facilitates data-driven decision making in recommendation outcomes. This approach ensures that recommendations are not only personalized but also aligned with the collective sentiment of customers.

**Scalability and Adaptability:** The project's architecture, leveraging machine learning techniques and efficient algorithms, supports scalability and adaptability. As the user base grows or evolves, the recommendation system can efficiently handle larger datasets and adapt to emerging trends and patterns.

**Enhanced Business Insights:** The project generates valuable insights into customer sentiments, product preferences, and market trends through sentiment analysis. These insights can inform business strategies, marketing campaigns, and product development efforts, ultimately leading to better business outcomes.

# 4 DESIGN

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.

## 4.1 Usecase diagram

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. Actors are the external entities that interact with the system. The use cases are represented by either circles or ellipses. The Figure 4.1 shows the use case representation of the system.

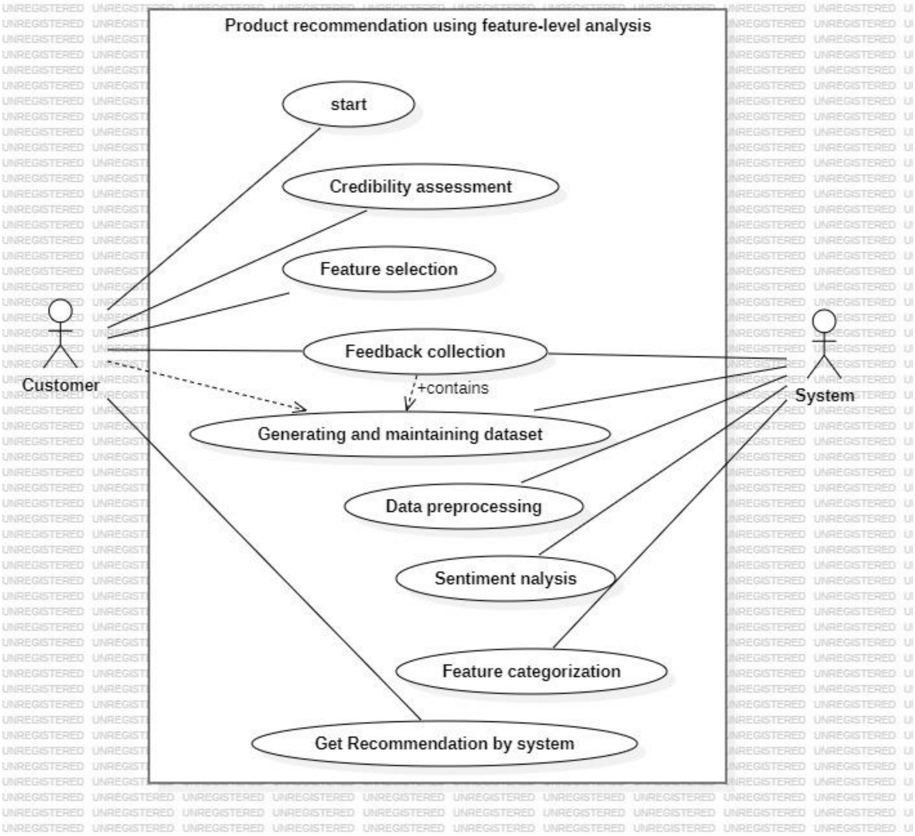


**Figure 4.1 Usecase Diagram**

## 4.2 Class diagram

Class diagrams give an overview of a system by showing its classes and the relationships among them. Class diagrams are static – they display what interacts but not what happens when they do interact. In general a class diagram consists of some set of attributes and operations. Operations will be performed on the data values of attributes. The Figure 4.2  shows the class diagram representation of the system.

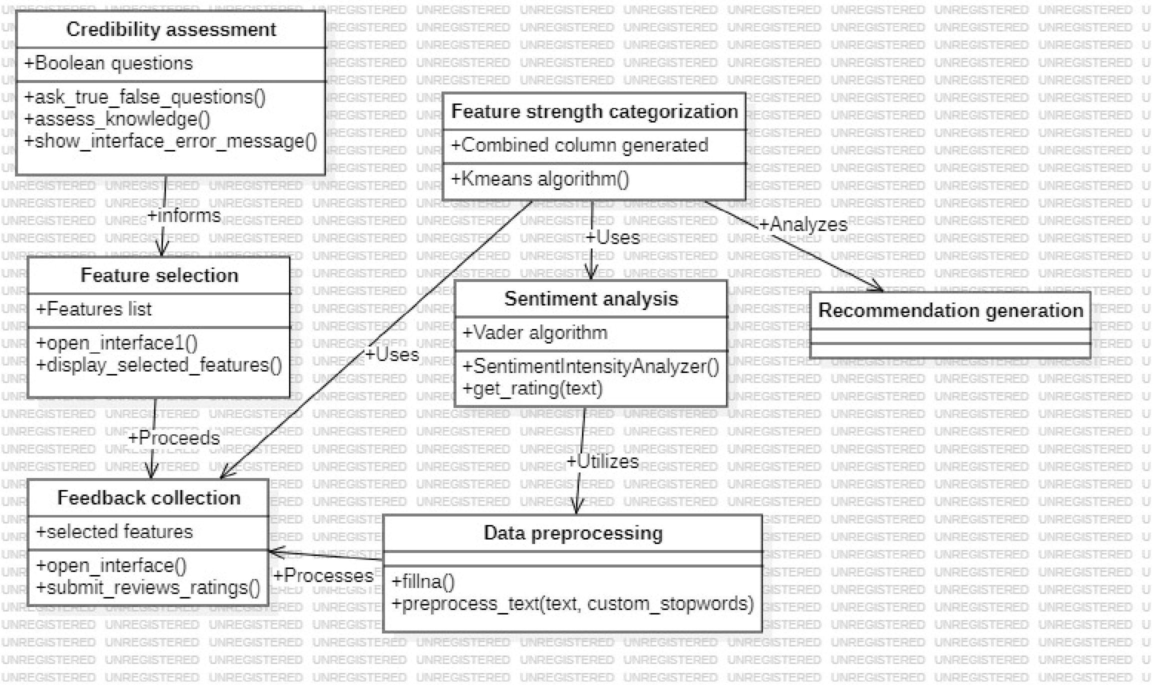


**Figure 4.2 Class Diagram**

## 4.3 Activity diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. In UML, an activity diagram provides a view of the behavior of a system by describing the sequence of actions in a process. The Figure 4.3 shows the activity diagram representation of the system.

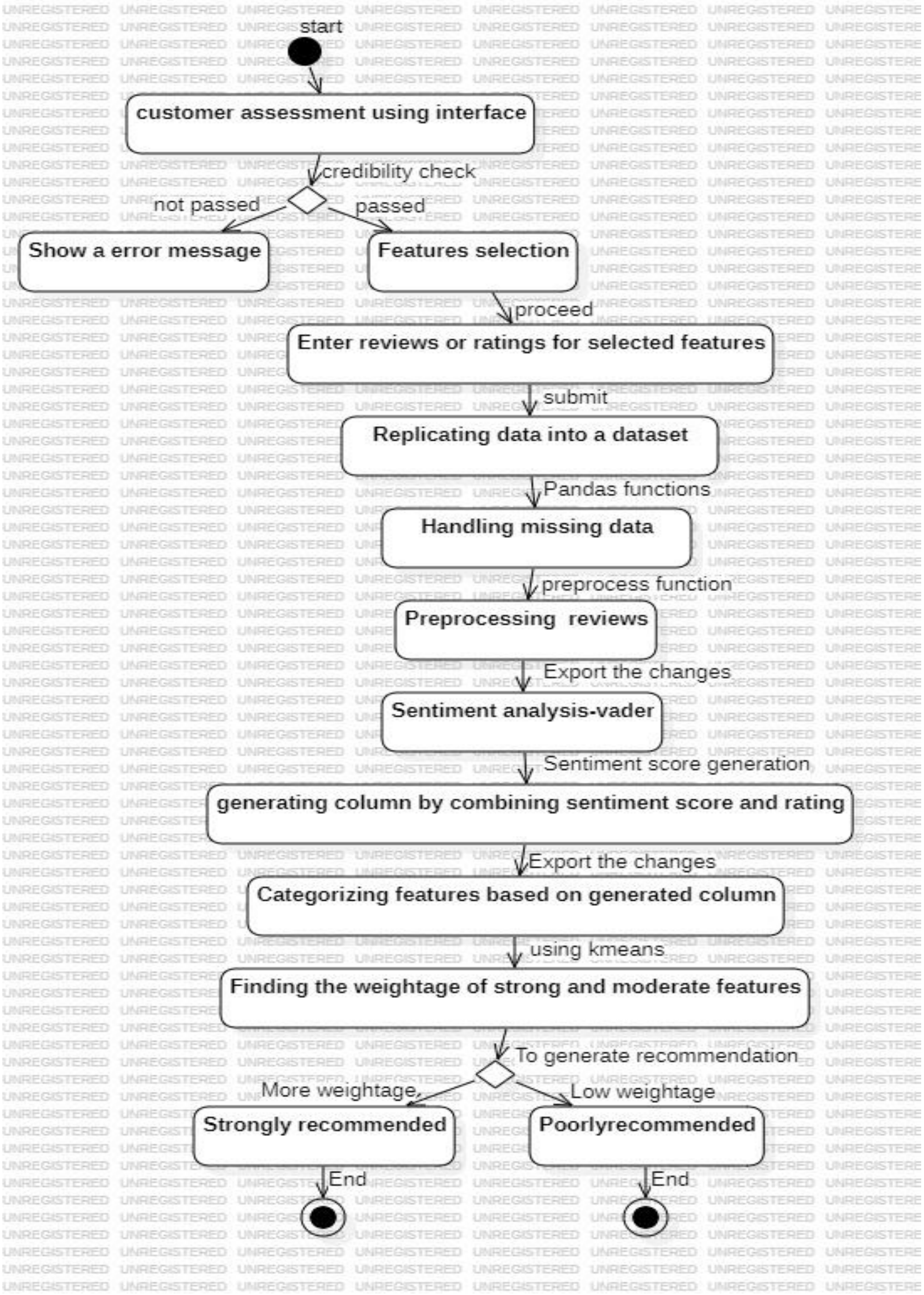

start
customer assessment using interface
credibility check
not passed
passed
Show a error message
Features selection
proceed
Enter reviews or ratings for selected features
submit
Replicating data into a dataset
Pandas functions
Handling missing data
preprocess function
Preprocessing reviews
Export the changes
Sentiment analysis-vader
Sentiment score generation
generating column by combining sentiment score and rating
Export the changes
Categorizing features based on generated column
using kmeans
Finding the weightage of strong and moderate features
To generate recommendation
More weightage
Low weightage
Strongly recommended
Poorlyrecommended
End
End

**Figure 4.3 Activity Diagram**

## 4.4 Statechart diagram

A state diagram, also known as a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML). The Figure 4.4 shows the state chart diagram representation of the system.

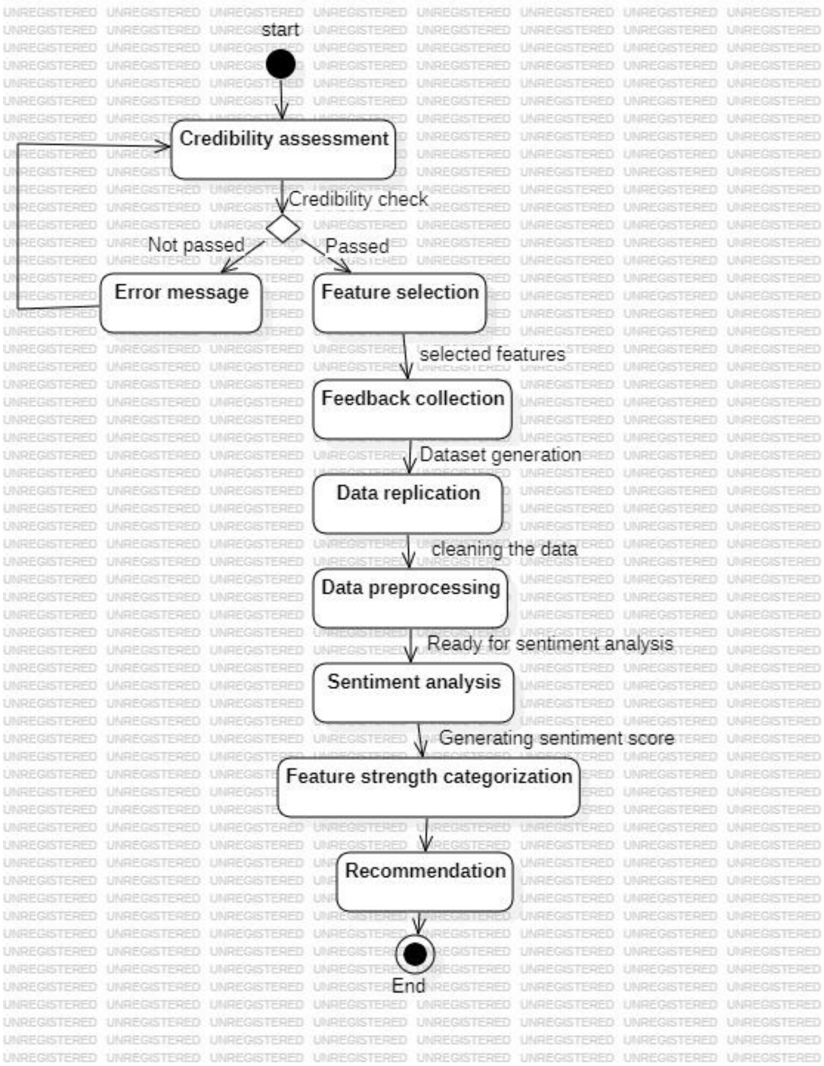


**Figure 4.4 Statechart Diagram**

## 4.5 Sequence diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. The Figure 4.5 shows the sequence diagram representation of the system.

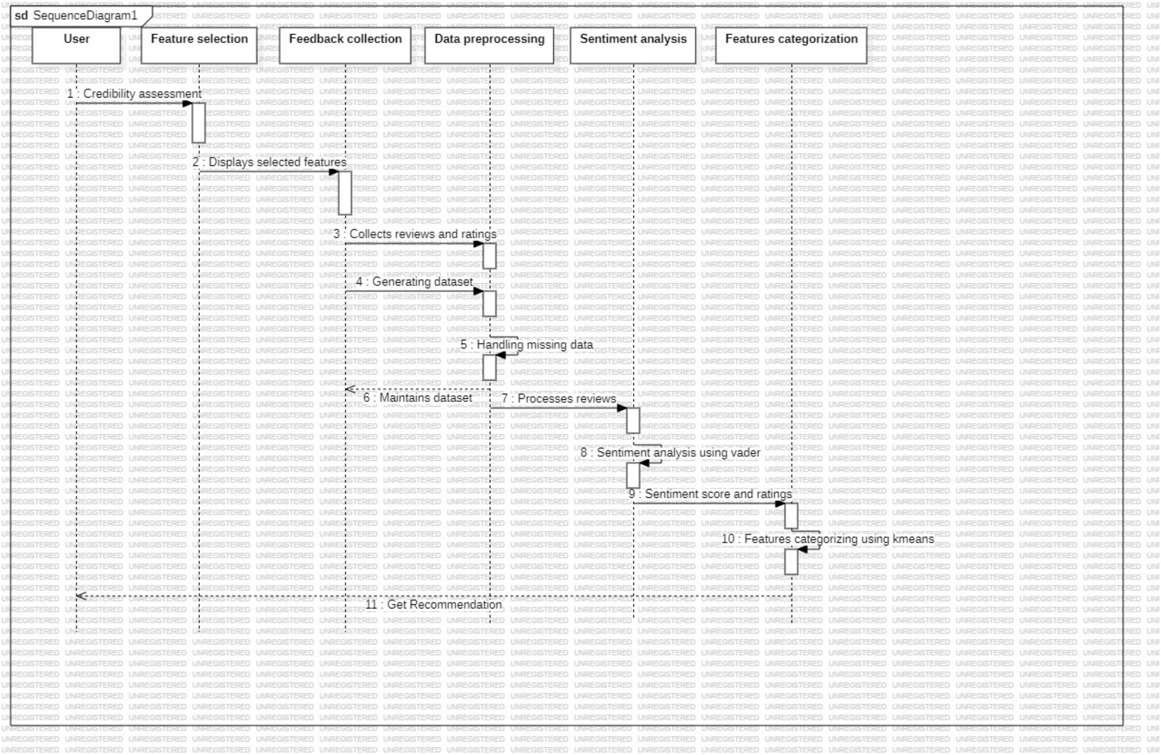

sd SequenceDiagram1
User
Feature selection
Feedback collection
Data preprocessing
Sentiment analysis
Features categorization
1 : Credibility assessment
2 : Displays selected features
3 : Collects reviews and ratings
4 : Generating dataset
5 : Handling missing data
6 : Maintains dataset
7 : Processes reviews
8 : Sentiment analysis using vader
9 : Sentiment score and ratings
10 : Features categorizing using kmeans
11 : Get Recommendation

**Figure 4.5 Sequence Diagram**

# 5 IMPLEMENTATION

The proposed system can be implemented in a modular approach as follows:

## 5.1 Credibility assessment

In this module, primarily we aim to check the credibility of a user through a simple interface which collects Boolean-type responses from users.

To implement this module, we have used tkinter library which is the standard GUI (Graphical User Interface) toolkit for Python. It provides a set of built-in widgets and functions for creating desktop applications with graphical interfaces.

The interface contains few Boolean-type questions to assess credibility of a customer. Among these, few need to be answered mandatorily such that the customer can meet the threshold and can redirect to the next interface.

If customer does not pass the assessment, an error message is displayed.

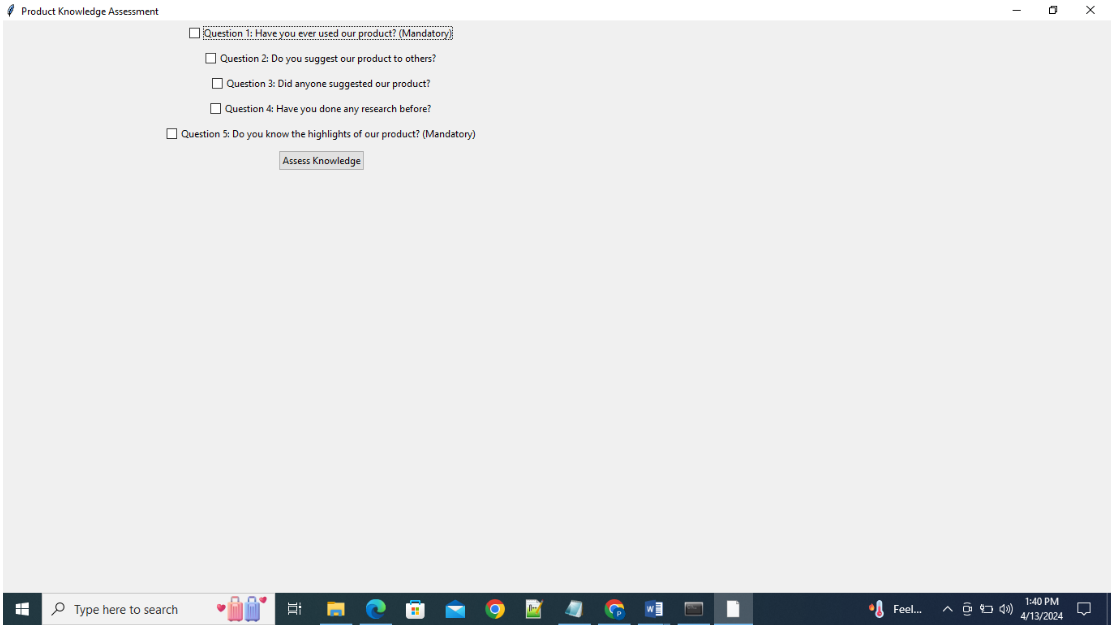
In our code, the implementation of this module can be observed through these methods:

ask_true_false_questions()

assess_knowledge()

show_interface_error_message(message)

The following images shows the results of this module:



☐ Question 1: Have you ever used our product? (Mandatory)

☐ Question 2: Do you suggest our product to others?

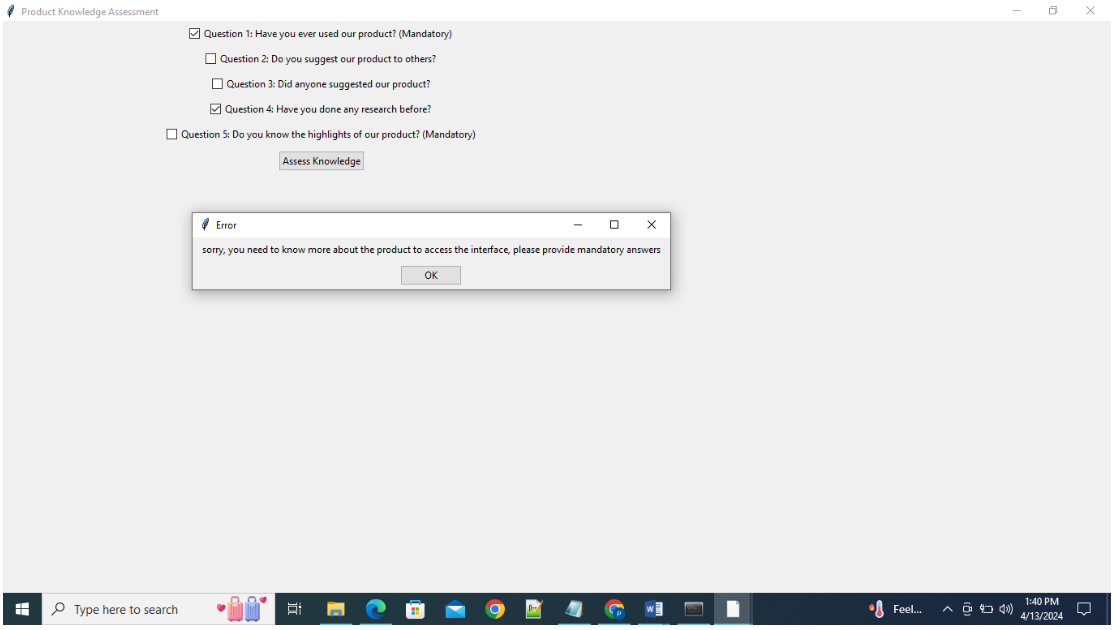
☐ Question 3: Did anyone suggested our product?

☐ Question 4: Have you done any research before?

☐ Question 5: Do you know the highlights of our product? (Mandatory)

**Figure 5.1 Credibility assessment**

If customer doesnot meet the threshold, output is as follows:



☑ Question 1: Have you ever used our product? (Mandatory)

☐ Question 2: Do you suggest our product to others?

☐ Question 3: Did anyone suggested our product?

☑ Question 4: Have you done any research before?

☐ Question 5: Do you know the highlights of our product? (Mandatory)

sorry, you need to know more about the product to access the interface, please provide mandatory answers

**Figure 5.2 Error**

## 5.2 Feature selection

After credibilty assessment, the customer is redirected to the next screen which contins a curated list of features extracted from a sample dataset. Here, we have considered around 40 features of a mobile.

Through this interface, the customer can select the features for which he want to give feedback in form of either reviews or ratings. To implement this module, we have used tkinter library. The input for feature selection is list of features and the output is the screen to provide feedback.

In our code, the implementation of feature selection can be observed through these methods:

open_interface()

display_selected_features()

The following image shows the results of this module:

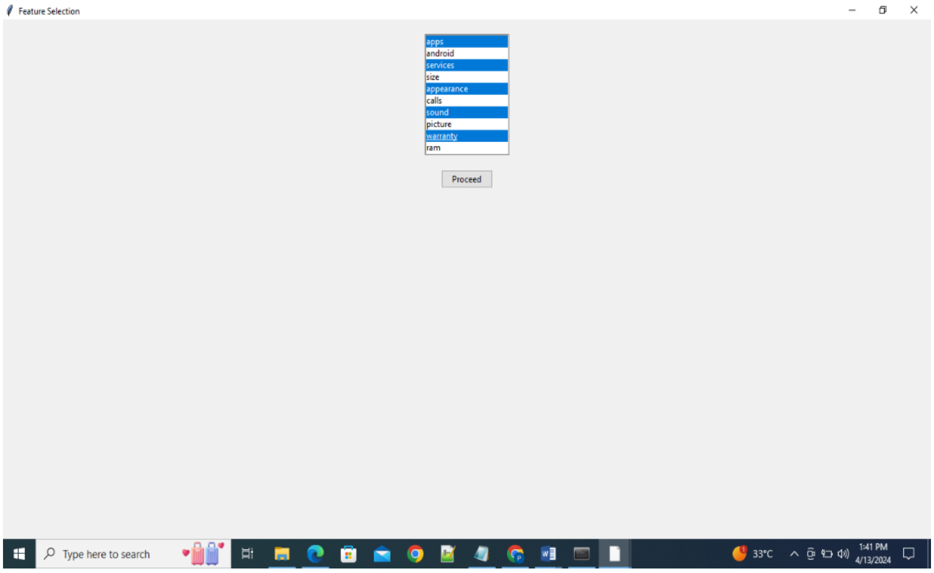


**Figure 5.3 Feature Selection**

## 5.3 Feedback collection and Dataset generation

After feature selection, the user can provide his feedback through either reviews or ratings.

The interface where customer provides their feedback, consists of selected features. After giving the feedback, the customer can click on submit, after which a dataset is generated through the collected data which helps in further analysis.

The dataset generated thus consists of selected features along with their corresponding reviews and ratings. To implement this module, we have used tkinter library.

In our code, the implementation of feedback collection can be observed through these methods:

open_interface()

submit_reviews_ratings()

The following images shows the results of this module:

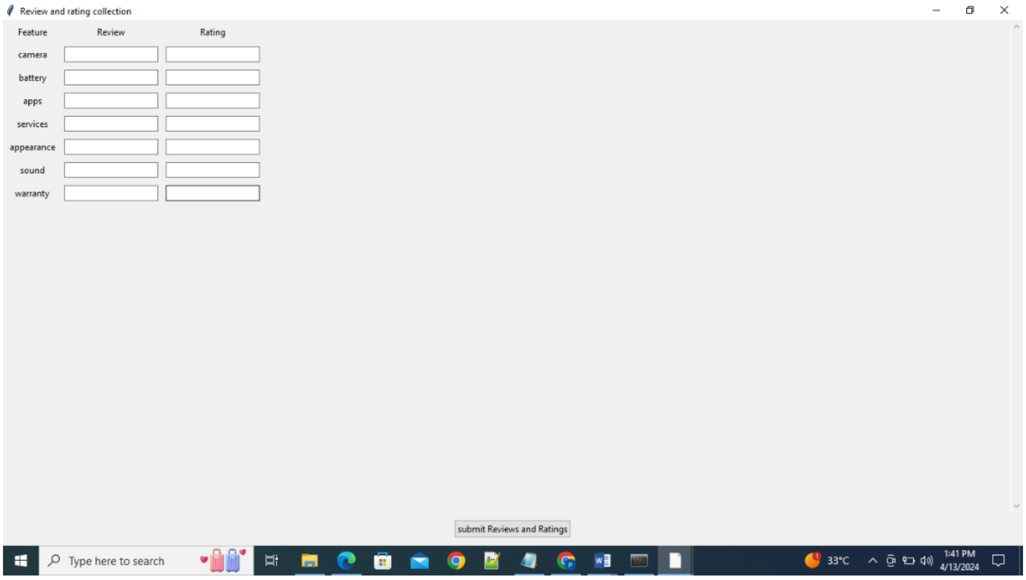
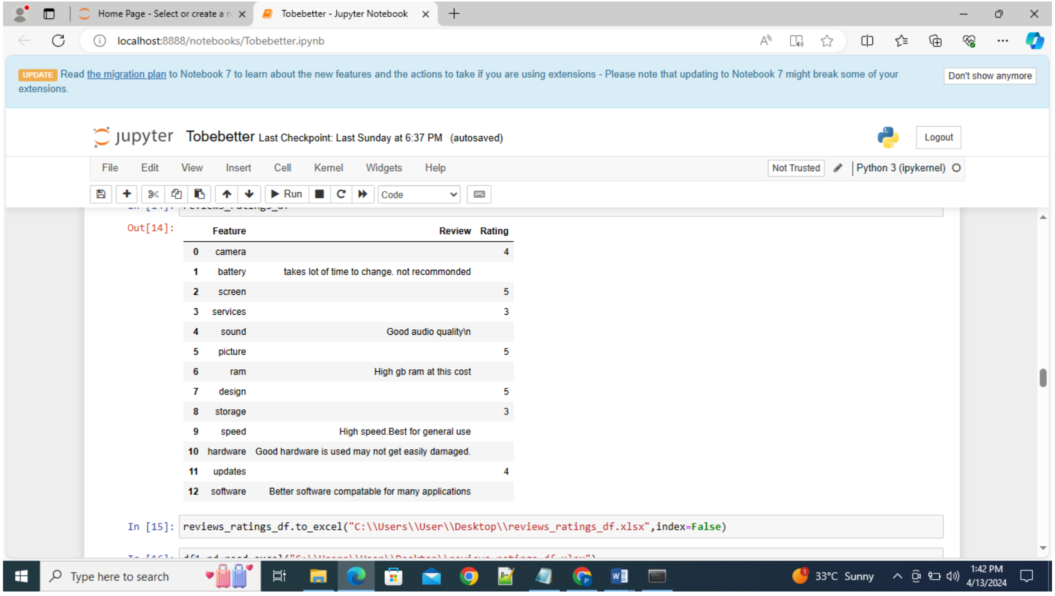


**Figure 5.4 Feedback Collection**



Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

Tobebetter Last Checkpoint: Last Sunday at 6:37 PM (autosaved)

Out[14]:

| | Feature | Review | Rating |
|---|---|---|---|
| 0 | camera | | 4 |
| 1 | battery | takes lot of time to change. not recommonded | |
| 2 | screen | | 5 |
| 3 | services | | 3 |
| 4 | sound | Good audio quality\n | |
| 5 | picture | | 5 |
| 6 | ram | High gb ram at this cost | |
| 7 | design | | 5 |
| 8 | storage | | 3 |
| 9 | speed | High speed.Best for general use | |
| 10 | hardware | Good hardware is used may not get easily damaged. | |
| 11 | updates | | 4 |
| 12 | software | Better software compatable for many applications | |

```
In [15]: reviews_ratings_df.to_excel("C:\\Users\\User\\Desktop\\reviews_ratings_df.xlsx",index=False)
```

**Figure 5.5 Generated Dataset**

## 5.4 Data preprocessing

As the dataset generated contains missing data it need to be handled, and also the textual reviews need to undergo preprocessing before we continue further with sentiment analysis.

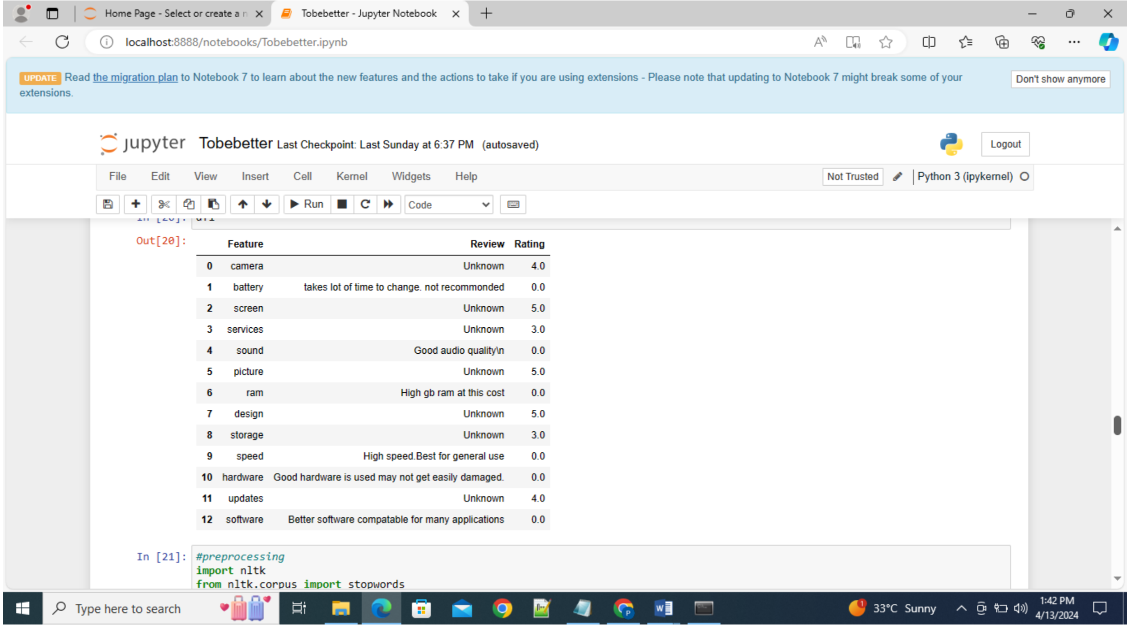
Hence, in this module, the dataset generated through feedback collection is considered as input and on which pandas library functions are applied to handle missing data.

The reviews column is considered as input for processing textual reviews which uses nltk library.

In our code, the implementation of processing textual reviews, can be observed through this method:

preprocess_text(text,custom_stopwords)

The result after handling missing data is as follows:



Out[20]:

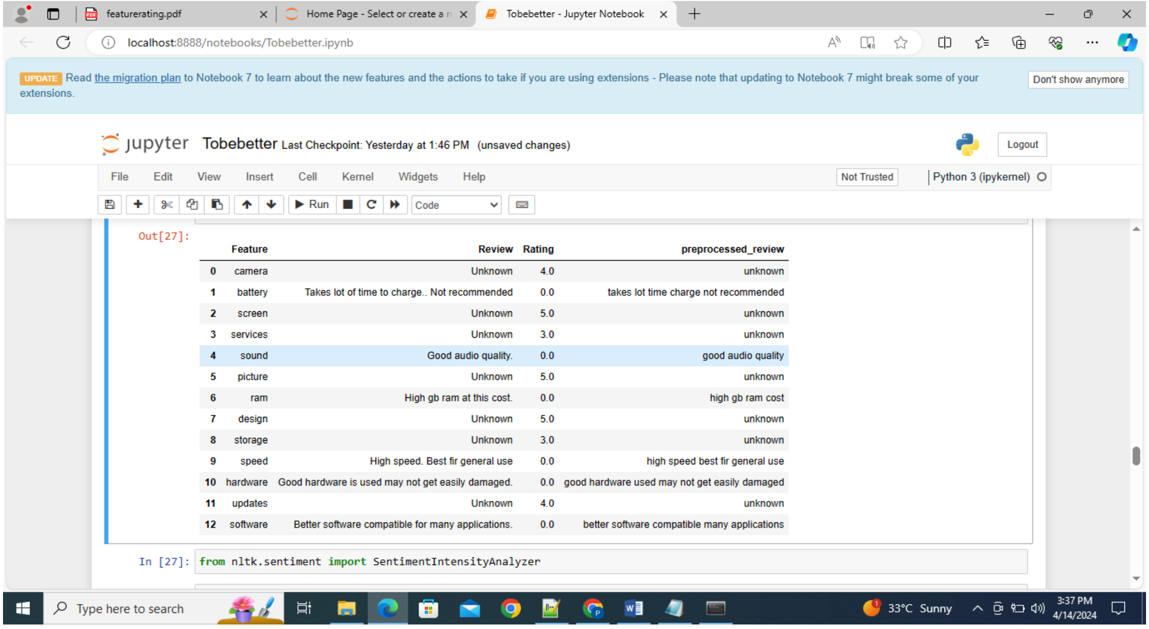
|  | Feature | Review | Rating |
|---|---|---|---|
| 0 | camera | Unknown | 4.0 |
| 1 | battery | takes lot of time to change. not recommonded | 0.0 |
| 2 | screen | Unknown | 5.0 |
| 3 | services | Unknown | 3.0 |
| 4 | sound | Good audio quality\n | 0.0 |
| 5 | picture | Unknown | 5.0 |
| 6 | ram | High gb ram at this cost | 0.0 |
| 7 | design | Unknown | 5.0 |
| 8 | storage | Unknown | 3.0 |
| 9 | speed | High speed.Best for general use | 0.0 |
| 10 | hardware | Good hardware is used may not get easily damaged. | 0.0 |
| 11 | updates | Unknown | 4.0 |
| 12 | software | Better software compatable for many applications | 0.0 |

In [21]:

```
#preprocessing
import nltk
from nltk.corpus import stopwords
```

**Figure 5.6 Handling Missing Data**

The result after processing reviews is as follows:



Out[27]:

|  | Feature | Review | Rating | preprocessed_review |
|---|---|---|---|---|
| 0 | camera | Unknown | 4.0 | unknown |
| 1 | battery | Takes lot of time to charge.. Not recommended | 0.0 | takes lot time charge not recommended |
| 2 | screen | Unknown | 5.0 | unknown |
| 3 | services | Unknown | 3.0 | unknown |
| 4 | sound | Good audio quality. | 0.0 | good audio quality |
| 5 | picture | Unknown | 5.0 | unknown |
| 6 | ram | High gb ram at this cost. | 0.0 | high gb ram cost |
| 7 | design | Unknown | 5.0 | unknown |
| 8 | storage | Unknown | 3.0 | unknown |
| 9 | speed | High speed. Best fir general use | 0.0 | high speed best fir general use |
| 10 | hardware | Good hardware is used may not get easily damaged. | 0.0 | good hardware used may not get easily damaged |
| 11 | updates | Unknown | 4.0 | unknown |
| 12 | software | Better software compatible for many applications. | 0.0 | better software compatible many applications |

In [27]:

```
from nltk.sentiment import SentimentIntensityAnalyzer
```

**Figure 5.7 Preprocessed Dataset**

## 5.5 Sentiment analysis

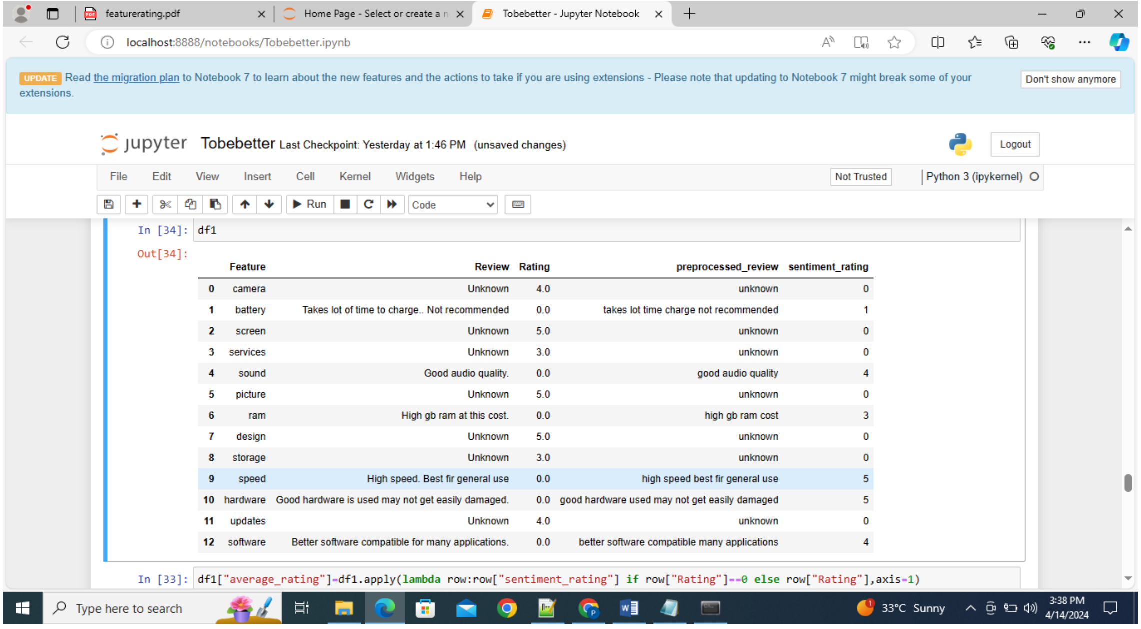
The key module of the proposed model is sentiment analysis. It helps in better feature-level analysis. Here, we are using VADER algorithm to generate sentiment scores for reviews in the range of 1 to 5.

The preprocessed reviews are considered as input and the output is the generated sentiment score.

In our code, the implementation of sentiment analysis to generate sentiment score can be observed through the method:

generate_rating()

The result after sentiment analysis is as follows:


featurerating.pdf
Home Page - Select or create a n
Tobebetter - Jupyter Notebook
localhost:8888/notebooks/Tobebetter.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter Tobebetter Last Checkpoint: Yesterday at 1:46 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
33°C Sunny
3:38 PM
4/14/2024

```
In [34]: df1
Out[34]:
```

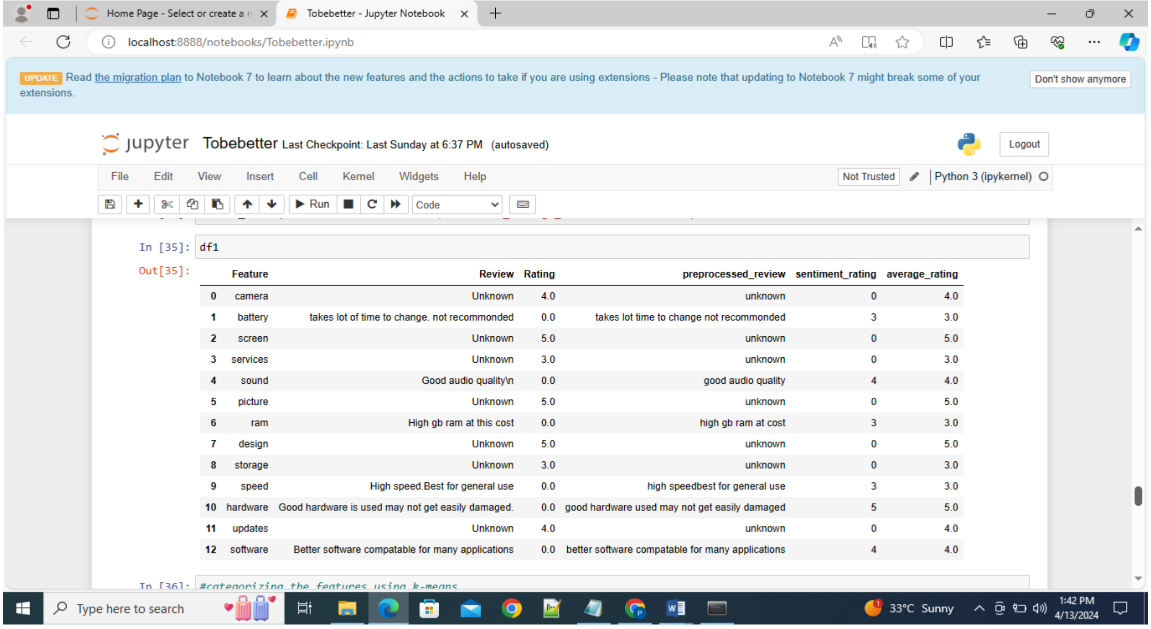
| | Feature | Review | Rating | preprocessed_review | sentiment_rating |
|---|---|---|---|---|---|
| 0 | camera | Unknown | 4.0 | unknown | 0 |
| 1 | battery | Takes lot of time to charge.. Not recommended | 0.0 | takes lot time charge not recommended | 1 |
| 2 | screen | Unknown | 5.0 | unknown | 0 |
| 3 | services | Unknown | 3.0 | unknown | 0 |
| 4 | sound | Good audio quality. | 0.0 | good audio quality | 4 |
| 5 | picture | Unknown | 5.0 | unknown | 0 |
| 6 | ram | High gb ram at this cost. | 0.0 | high gb ram cost | 3 |
| 7 | design | Unknown | 5.0 | unknown | 0 |
| 8 | storage | Unknown | 3.0 | unknown | 0 |
| 9 | speed | High speed. Best fir general use | 0.0 | high speed best fir general use | 5 |
| 10 | hardware | Good hardware is used may not get easily damaged. | 0.0 | good hardware used may not get easily damaged | 5 |
| 11 | updates | Unknown | 4.0 | unknown | 0 |
| 12 | software | Better software compatible for many applications. | 0.0 | better software compatible many applications | 4 |

```
In [33]: df1["average_rating"]=df1.apply(lambda row:row["sentiment_rating"] if row["Rating"]==0 else row["Rating"],axis=1)
```

**Figure 5.8 Generating Sentiment Score**

After generating sentiment score, these scores from reviews will be integrated with feature ratings to derive a combined metric, which is considered as final rating of the corresponding feature.

The dataset after generating final ratings for features looks as follows:


Home Page - Select or create a n
Tobebetter - Jupyter Notebook
localhost:8888/notebooks/Tobebetter.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter
Tobebetter Last Checkpoint: Last Sunday at 6:37 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
33°C Sunny
1:42 PM
4/13/2024

```
In [35]: df1
Out[35]:
```

| | Feature | Review | Rating | preprocessed_review | sentiment_rating | average_rating |
|---|---|---|---|---|---|---|
| 0 | camera | Unknown | 4.0 | unknown | 0 | 4.0 |
| 1 | battery | takes lot of time to change. not recommonded | 0.0 | takes lot time to change not recommonded | 3 | 3.0 |
| 2 | screen | Unknown | 5.0 | unknown | 0 | 5.0 |
| 3 | services | Unknown | 3.0 | unknown | 0 | 3.0 |
| 4 | sound | Good audio quality\n | 0.0 | good audio quality | 4 | 4.0 |
| 5 | picture | Unknown | 5.0 | unknown | 0 | 5.0 |
| 6 | ram | High gb ram at this cost | 0.0 | high gb ram at cost | 3 | 3.0 |
| 7 | design | Unknown | 5.0 | unknown | 0 | 5.0 |
| 8 | storage | Unknown | 3.0 | unknown | 0 | 3.0 |
| 9 | speed | High speed.Best for general use | 0.0 | high speedbest for general use | 3 | 3.0 |
| 10 | hardware | Good hardware is used may not get easily damaged. | 0.0 | good hardware used may not get easily damaged | 5 | 5.0 |
| 11 | updates | Unknown | 4.0 | unknown | 0 | 4.0 |
| 12 | software | Better software compatable for many applications | 0.0 | better software compatable for many applications | 4 | 4.0 |

```
In [36]: #categorizing the features using k-means
```

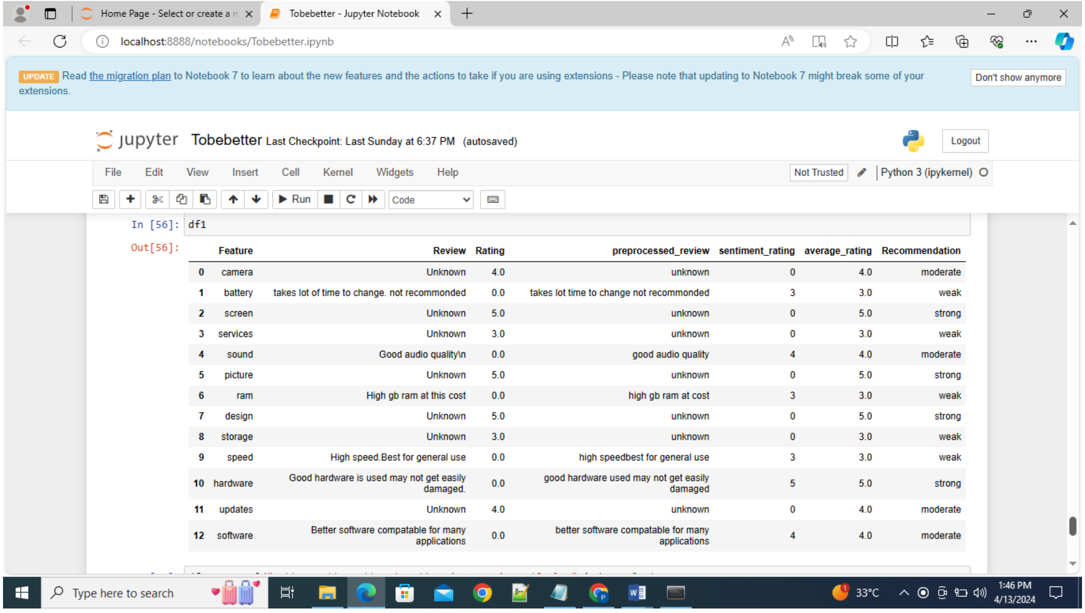
**Figure 5.9 Generating Final Feature Rating**

## 5.6 Feature strength categorization

In this module, with the help of generated final rating column, features are catehgorized as strong,moderate and weak through kmeans clustering algorithm.

This categorization further helps in recommendation generation and provides a more meaningful feature-level analysis.

The result of this module is as follows:



Home Page - Select or create a n
Tobebetter - Jupyter Notebook
localhost:8888/notebooks/Tobebetter.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter Tobebetter Last Checkpoint: Last Sunday at 6:37 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
33°C
1:46 PM
4/13/2024

```
In [56]: df1
```

Out[56]:

|  | Feature | Review | Rating | preprocessed_review | sentiment_rating | average_rating | Recommendation |
|---|---|---|---|---|---|---|---|
| 0 | camera | Unknown | 4.0 | unknown | 0 | 4.0 | moderate |
| 1 | battery | takes lot of time to change. not recommonded | 0.0 | takes lot time to change not recommonded | 3 | 3.0 | weak |
| 2 | screen | Unknown | 5.0 | unknown | 0 | 5.0 | strong |
| 3 | services | Unknown | 3.0 | unknown | 0 | 3.0 | weak |
| 4 | sound | Good audio quality\n | 0.0 | good audio quality | 4 | 4.0 | moderate |
| 5 | picture | Unknown | 5.0 | unknown | 0 | 5.0 | strong |
| 6 | ram | High gb ram at this cost | 0.0 | high gb ram at cost | 3 | 3.0 | weak |
| 7 | design | Unknown | 5.0 | unknown | 0 | 5.0 | strong |
| 8 | storage | Unknown | 3.0 | unknown | 0 | 3.0 | weak |
| 9 | speed | High speed.Best for general use | 0.0 | high speedbest for general use | 3 | 3.0 | weak |
| 10 | hardware | Good hardware is used may not get easily damaged. | 0.0 | good hardware used may not get easily damaged | 5 | 5.0 | strong |
| 11 | updates | Unknown | 4.0 | unknown | 0 | 4.0 | moderate |
| 12 | software | Better software compatable for many applications | 0.0 | better software compatable for many applications | 4 | 4.0 | moderate |

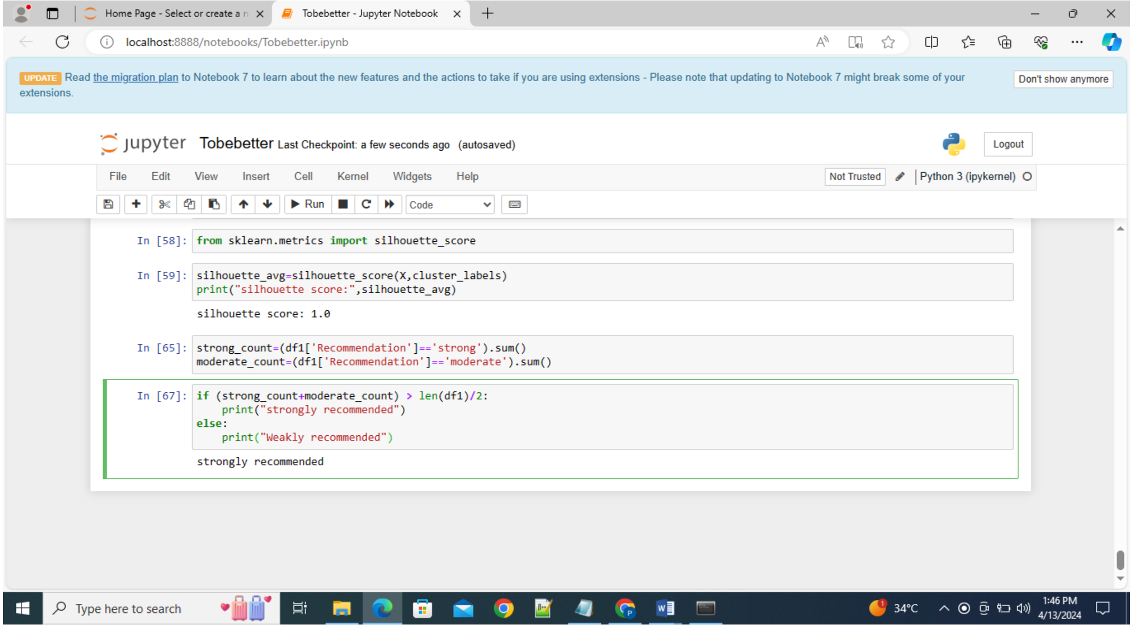
Figure 5.10 Feature Strength Categorization

## 5.7 Recommendation generation

In this module, with the help of feature categories, the recommendation is generated as strong or weak.

Decision rules will be implemented to determine recommendation outcomes (e.g., strong recommendation vs. weak recommendation) based on the balance and strength of categorized features selected by the customer.

The result of this module is as follows:



Home Page - Select or create a n
Tobebetter - Jupyter Notebook
localhost:8888/notebooks/Tobebetter.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter Tobebetter Last Checkpoint: a few seconds ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
34°C
1:46 PM
4/13/2024

```
In [58]: from sklearn.metrics import silhouette_score

In [59]: silhouette_avg=silhouette_score(X,cluster_labels)
         print("silhouette score:",silhouette_avg)

         silhouette score: 1.0

In [65]: strong_count=(df1['Recommendation']=='strong').sum()
         moderate_count=(df1['Recommendation']=='moderate').sum()

In [67]: if (strong_count+moderate_count) > len(df1)/2:
             print("strongly recommended")
         else:
             print("Weakly recommended")

         strongly recommended
```

**Figure 5.11 Recommendation Generation**

Finally, Through the implementation of model and by understanding the results, we can say that the proposed system serves as a part of recommendation systems, though it does not completely implement a recommendation system, it aids in the processes of generating accurate recommendations through feature-level analysis. It can be merged with the actual existing recommendation systems to generate more personalized product suggestions which in turn helps businesses by providing actionable insights such as identifying popular product features, understanding customer sentiment trends, and optimizing product recommendations.

# 6 CONCLUSION AND FUTURE WORK

In today’s world, we can observe most of the businesses are running through their own websites. So, it is necessary that consumer opinion need to be considered and their requirements are met. This system provides a deep feature-level analysis by incorporating customer feedback and sentiment analysis. It combines credibility assessment, feature selection, feedback collection, sentiment analysis and feature strength categorization which solely relies on features of the product. The system results in personalized product suggestions that resonate with individual users. This personalized approach enhances user engagement and satisfaction with the platform. It keeps both the manufacturer and the customer well-informed in the decisions to make in improving the product and buying, respectively. Different customers are interested in different features. Thus, feature-level ratings can make buying decisions personalized. The system can find its applications in recommendation systems, customer research etc..

**Future enhancements**

Looking ahead, there are several avenues for further development and enhancement of this system:

**Strict credibility assessment**

Rather than simply evaluating credibility using a method, the system can incorporate any cryptographic algorithm through which false reviews can be prevented.

### Advanced Machine Learning Models

For sentiment analysis, one can opt different kind of algorithms like LSTM etc.. and one can explore different probabilities of results.They may provide more accurate results too.

### User Interface and Interaction Design

The user interface can be designed more engaging and interactive.Incorporate interactive features such as personalized dashboards, recommendation explanations, and proactive feedback mechanisms to encourage user interaction and feedback.

### Dynamic Clustering Techniques

Implement dynamic clustering techniques that adapt to evolving customer sentiments and product trends in real-time. Explore algorithms that automatically adjust cluster boundaries based on changing data distributions, ensuring the recommendation system remains agile and responsive.

### Incorporating into dynamic recommendation systems

Incorporate the system into current recommendation systems to give tremendous results depicting personalized product suggestions and accurate recommendations through feature-level analysis.

# 7 REFERENCES

[1] J. Mehta, J. Patil, R. Patil, M. Somani, and S. Varma, “Sentiment analysis on product reviews using hadoop,” International Journal of Computer Applications, vol. 9, no. 11, pp. 0975–8887, 2016.

[2] X. Fang and J. Zhan, “Sentiment analysis using product review data,”Journal of Big Data, vol. 2, no. 1, p. 5, Jun 2015. [Online].Available:https://doi.org/10.1186/s40537-015-0015-2

[3] D. K. Raja and S. Pushpa, “Feature level review table generation for e commerce websites to produce qualitative rating of the products,” Future Computing and Informatics Journal, vol. 2, no. 2, pp. 118–124, 2017.

[4] N. Nandal, R. Tanwar, and J. Pruthi, “Machine learning based aspect level sentiment analysis for amazon products,” Spatial Information Research, pp. 1–7, 2020.

[5]R.-C. Chen et al., “User rating classification via deep belief network learning and sentiment analysis,” IEEE Transactions on Computational Social Systems, 2019.

[6] S. Kumar, K. De, and P. P. Roy, “Movie recommendation system using sentiment analysis from microblogging data,” IEEE Transactions on Computational Social Systems, 2020.

[7] M. Ling, Q. Chen, Q. Sun, and Y. Jia, “Hybrid neural network for sina weibo sentiment analysis,” IEEE Transactions on Computational Social Systems, 2020.

[8] K. Chakraborty, S. Bhattacharyya, and R. Bag, “A survey of sentiment analysis from social media data,” IEEE Transactions on Computational Social Systems, vol. 7, no. 2, pp. 450–464, 2020.