**AUTOMATED REVIEW RATING SYSTEM**

1. **Objective**

The goal of this project is to automatically predict or assign ratings (like 1–5 stars) to user reviews (text) using Natural Language Processing (NLP) techniques.  
This saves time compared to manual rating and can be used for e-commerce, app reviews, or any platform collecting feedback.

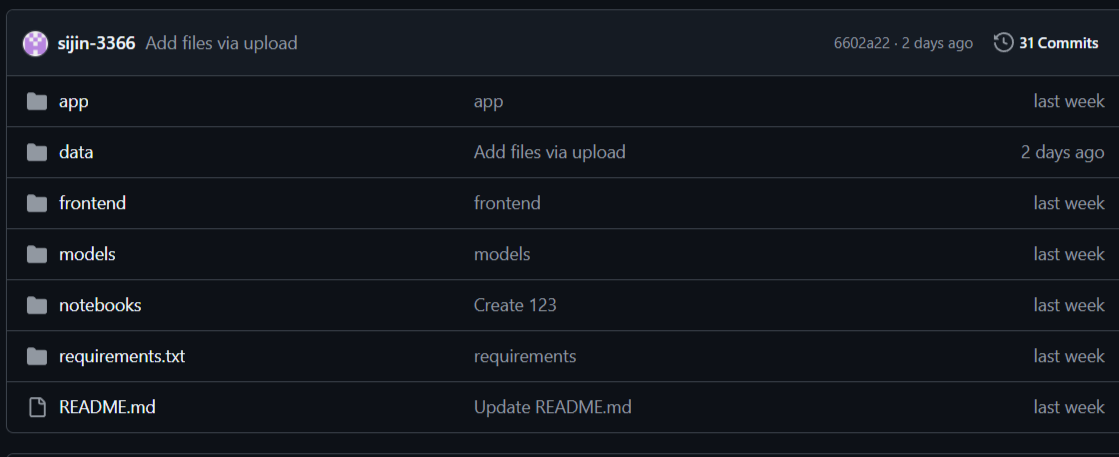
1. **Environment Setup**

* Installed python 3.13.5 with libraries: numpy,pandas ,matplotlib ,seaborn ,scikit-learn,nltk.
* IDE used:VS code

1. **GitHub Setup**

Created a repository : [Automated\_review\_rating\_system](https://github.com/sijin-3366/Automated_review_rating_system)

Structure of directory



1. **Data Collection**

* Datas are collected from Kaggle and Data.world
* Data collects from Kaggle about 500000 rows
* Data collects from Data.world about 390000
* Kaggle Dataset Link: <https://github.com/sijin-3366/Automated_review_rating_system/tree/main/data/Kaggle>
* Data.world Dataset Link:<https://github.com/sijin-3366/Automated_review_rating_system/tree/main/data/Google>

1. **Import packages + Data loading + Concatenate**

**Code:**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**import nltk**

**import warnings**

**warnings.filterwarnings('ignore')**

**df1=pd.read\_csv('product\_data.csv')**

**df2=pd.read\_csv('product\_data1.csv')**

**df=pd.concat([df1,df2],ignore\_index=True)**

**Data Preprocessing**

* **Data preprocessing** is the step of **cleaning and transforming raw data** into a usable format for analysis or model training.
* It improves **data quality**, reduces **noise**, and helps models learn more effectively.

**5.1 Basic Steps**

* Shape: (number\_of\_rows, number\_of\_columns)

df.shape

* Head: Collect first 5 rows

df.head()

* Tail: Collect Last 5 rows

df.tail()

* Columns: Find the column Labels

df.columns

* Drop Unnecessary columns

**5.2 Handling Missing Values**

* Handling missing values is one of the most important parts of data preprocessing.
* Missing values can distort your analysis or model training
* Find Missing values

df.isna().sum()

rating 421

review 41

* Drop Missing values

df=df.dropna()

df.isna().sum()

rating 0

review 0

**5.3 Unique Rating**

* **Making row unique**

**Code:**

**df[‘rating’].unique**

**array(['5', '3', '1', '4', '2',**

**'Pigeon Favourite Electric Kettle?ÿ?ÿ(1.5 L, Silver, Black)',**

**'Bajaj DX 2 L/W Dry Iron',**

**'Nova Plus Amaze NI 10 1100 W Dry Iron?ÿ?ÿ(Grey & Turquoise)', 's',**

**'4.2', '4.0', '3.9', '4.1', '4.3', '4.4', '4.5', '3.7', '3.3',**

**'3.6', '3.4', '3.8', '3.5', '4.6', '3.2', '5.0', '4.7', '3.0',**

**'2.8', '3.1', '4.8', '2.3', '|', '2.6', '2.9', 2, 3, 4, 1, 5,**

**'1.0', '2.0'], dtype=object)**

**df=df[pd.to\_numeric(df['rating'],errors='coerce').between(1.0,5.0)]**

**df['rating']=np.round(df['rating']).astype(int)**

**df['rating'].unique()**

**array([5, 3, 1, 4, 2])**

* 1. **Remove**
  2. **Remove duplicates and their corresponding row**

**Code:**

**duplicate=df\_clean[df\_clean.duplicated(subset=['rating','review'], keep=False)]**

**new\_data=df\_clean.drop(duplicate.index)**

* 1. **Remove Conflicting reviews**

**Code:**

**variable=df.groupby('review')['rating'].nunique().loc[lambda x: x > 1].index**

**df\_clean=df[~df['review'].isin(variable)]**

**df\_clean=df\_clean.reset\_index(drop=True)**

**print(df\_clean)**

* 1. **Visualization**

**Bar chart**

**Code:**

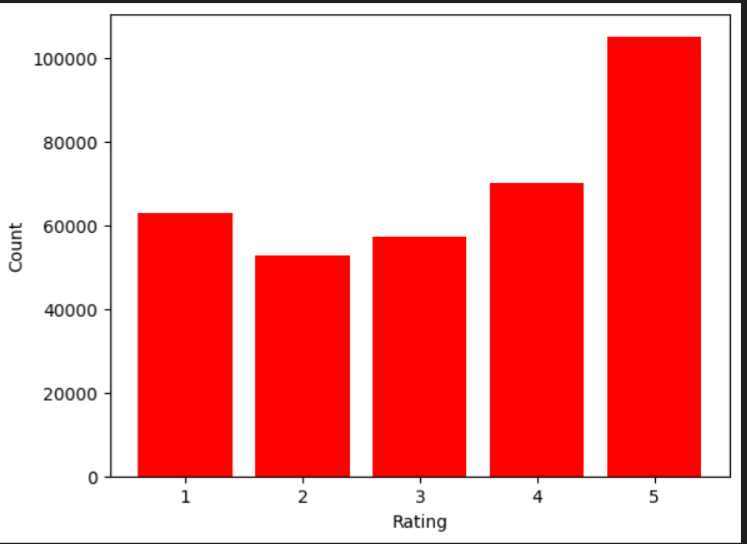
**x=new\_data['rating'].value\_counts().sort\_index()**

**plt.bar(x.index,x.values,color='red')**

**plt.xlabel("Rating")**

**plt.ylabel("Count")**

**plt.show()**

****

**Taking unique rows**

**Code:**

**dat=pd.read\_csv('bal\_clean.csv')**

**drop\_df=new\_data.merge(dat,how='outer',indicator=True).query('\_merge=="left\_only"').drop('\_merge',axis=1)**

**drop\_df['rating'].value\_counts()**

* 1. **Imbalancing**
* **5→20%=30000  ,  4→30%=45000   ,   3→25%=37500   ,   2→15%=22500   ,   1→10%=15,000**

**Code:**

imb\_1=drop\_df[drop\_df['rating']==1].sample(8000,random\_state=42)

imb\_2=drop\_df[drop\_df['rating']==2].sample(12000,random\_state=42)

imb\_3=drop\_df[drop\_df['rating']==3].sample(20000,random\_state=42)

imb\_4=drop\_df[drop\_df['rating']==4].sample(24000,random\_state=42)

imb\_5=drop\_df[drop\_df['rating']==5].sample(16000,random\_state=42)

imbalance=pd.concat([imb\_1,imb\_2,imb\_3,imb\_4,imb\_5],ignore\_index=True)

imbalance

**Imbalanced dataset link:** [**https://github.com/sijin-3366/Automated\_review\_rating\_system/blob/main/data/imb\_clean.csv**](https://github.com/sijin-3366/Automated_review_rating_system/blob/main/data/imb_clean.csv)

**Bar chart**

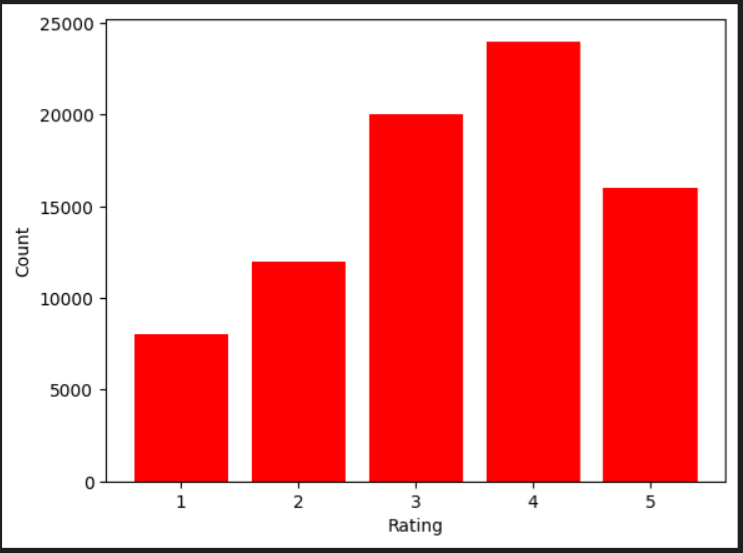
x=imbalance['rating'].value\_counts().sort\_index()

plt.bar(x.index,x.values)

plt.xlabel('Rating')

plt.ylabel('Count')

plt.show

****

**Pie chart**

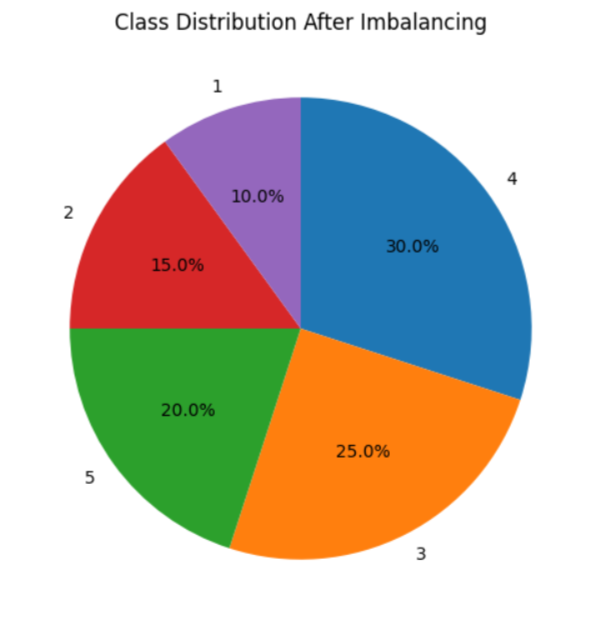
count= balance['rating'].value\_counts()

plt.figure(figsize=(6,6))

plt.pie(count, labels=count.index,autopct='%1.1f%%',startangle=90,counterclock=False)

plt.title('Class Distribution After Balancing')

plt.show()

**e**

1. **NATURAL LANGUAGE PROCESSING**

Natural Language Processing (NLP) is a field of Artificial Intelligence that focuses on enabling computers to understand, interpret and generate human language. In an automatic review rating system, NLP techniques are used to read customer reviews and predict their sentiment or rating automatically

**6.1 Tokenization**

Tokenization is the process of breaking a large text into smaller units called tokens. These tokens can be words, subwords, characters, or even sentences, depending on the level of analysis required.

**Code:**

**from nltk.tokenize import TweetTokenizer**

**tk=TweetTokenizer()**

**review=review.apply(lambda x:tk.tokenize(x)).apply(lambda x:' '.join(x))**

**6.2 Regular Expression**

A regular expression (often abbreviated as regex) is a special sequence of characters that defines a search pattern. It is used to match, find, or manipulate specific strings of text according to defined rules.

**Code:**

**import re**

**review=review.str.replace('[^a-zA-Z0-9]',' ',regex=True)**

**review**

What this code does:

It cleans the text by removing all special characters (punctuation, symbols) and replacing them with spaces, leaving only letters and numbers. This is a common preprocessing step in NLP to normalize text before tokenization or modeling.

**6.3 Filter text**

**Code:**

**from nltk.tokenize import TweetTokenizer**

**review=review.apply(lambda x:' '.join([w for w in tk.tokenize(x) if len(w)>=3]))**

**review**

**What this code does:**

* Tokenizes each text (especially social media-style text) into words and symbols using TweetTokenizer.
* Removes short tokens (less than 3 characters).
* Reconstructs the cleaned tokens into a single string per review.

**6.4 Lemmitization**

Lemmatization is a text-processing technique in Natural Language Processing (NLP).  
It reduces words to their lemma (dictionary form or base form) by considering the word’s meaning and part of speech (POS).The resulting form may not always be a real dictionary word but serves as a common representation for related words.

**Code:**

**from nltk.stem import WordNetLemmatizer**

**nltk.download('wordnet')**

**nltk.download('omw-1.4')**

**lemmatizer = WordNetLemmatizer()**

**review=review.apply(lambda x:[lemmatizer.lemmatize(i.lower(),pos='v') for i in tk.tokenize(x)]).apply(lambda x:' '.join(x))**

**review**

**Why use Lemmatization instead of stemming?**

**Stemming just chops off prefixes/suffixes using simple rules but Lemmatization uses a dictionary + part-of-speech to return an actual word .**

**6.5 Remove Stopwords**

* Stop words are common words (such as *“the”*, *“is”*, *“and”*, *“in”*) that usually carry little meaning by themselves.
* Removing them is a typical preprocessing step to reduce noise and focus on meaningful terms

**Purpose:**

* Reduce dimensionality of text data.
* Speed up processing and training of models.
* Improve accuracy by keeping only informative words.

**Stopwords =[**'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves',

'you', "you're", "you've", "you'll", "you'd", 'your', 'yours',

'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she',

"she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself',

'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which',

'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am',

'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has',

'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the',

'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of',

'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into',

'through', 'during', 'before', 'after', 'above', 'below', 'to',

'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under',

'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where',

'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most',

'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same',

'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just',

'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o',

're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",

'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn',

"hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn',

"mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan',

"shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren',

"weren't", 'won', "won't", 'wouldn', "wouldn't"]

* **English stop word list contains 179 stop words.**

**Code:**

**from nltk.corpus import stopwords**

**stop=stopwords.words('english')**

**review=review.apply(lambda x:[i for i in tk.tokenize(x) if i not in stop]).apply(lambda x:' '.join(x))**

**review**

1. **Train-Test Split**

Train–test split is the process of dividing a dataset into two parts:

* a training set used to fit (train) the model, and
* a test set used to evaluate the model’s performance on unseen data.

**Purpose:**

* Prevent overfitting: ensures the model is evaluated on data it hasn’t seen before.
* Estimate generalization: gives a realistic measure of how well the model will perform on new data.
* Provide a fair evaluation: separates data used for learning from data used for testing.

**Code:**

**from sklearn.model\_selection import train\_test\_split**

**x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.30,random\_state=42,stratify=y)**

**Parameters:**

* x,y - Features(x) and Target labels(y)
* test\_size - Fraction or number of samples to use for the test set (e.g., 0.3 = 30%)
* random\_state- Seed for reproducibility; using the same seed gives the same split.
* Stratify=y - Preserves class proportions of y in train/test sets (important for classification).

1. **TF–IDF (Term Frequency – Inverse Document Frequency)**

TF–IDF is a numerical statistic that reflects how important a word is to a document in a collection (corpus).  
It combines two measures:

* Term Frequency (TF): how frequently a word appears in a single document.
* Inverse Document Frequency (IDF): how unique or rare the word is across all documents.

**Formula:**

**TF-IDF(t,d)=TF(t,d)×IDF(t)**

* **TF(t,d) = number of times term *t* appears in document *d* ÷ total number of terms in *d*.**
* **IDF(t) =log(Total Documents/no:of Documents containing t)**
* **t=term(word)**
* **d=document**

**Samples of 1 star**

* **Product quality is quite nice but earphone is quite big then my ears that's why I am returning this product**
* **You should decide.., buy or not... It is just after one week I have found the damage... Actually, I have started to use it...**
* **Didn't like the product earbuds, they are not at all comfortable. Asked my few other friends to try it on, even they said not comfortable. Starts paining in the ear after 20 mins of usage, which results into massive headache.Also there's no return policy from Amazon nor JBL as well.Feels like both companies doesn't care about customer complaint.My suggestion would be to buy another model except this.**
* **Waste of money, not an Original product..**
* **This is the first time Very bad experiance from flip cart. I didnt get the installation on proper time. My own risk find some one**
* **Very bad  quality product  i want to refund**

**Samples of 2 star**

* **Need more value For the Money**
* **Almost perfect but not quite**
* **Terrible. Not worth the price. Awful durability. Had a return option been there I would have definitely used it.**
* **No bass until you use equlilizer\*Battery is only good in this product\*Bluetooth stopped working not expected from jbl because my second jbl worked for approx 2 year as it was third.**
* **This product one side not working one side good condition but another site not working how to change my Bluetooth ear please inform to me**
* **Battery about to die I am very sad because I have lost my money battery backap zero so don't purchase.**

**Samples of 3 star**

* **Noise level need to reduce and mini jaar should require 4 blades instead of 2 no proper mixing.**
* **Sometimes fails to stay connected**
* **Quality is not good according to the price price is high than quality**
* **Product is good. I got it for 42k. Programming related software works perfectly. Have issue with battery it last only for max 3-4hrs without power saving on.Sound is also clear but not that loud.**
* **Quality is OK but not value for money**
* **Color is nice...size is also good... fabric quality is good as per the price..but i got the piece which was little dirty.**

**Samples of 4 star**

* **A very powerful, distortion-free audio with great bass.  Inline remote control and microphone for mobile devices. Comfortable, secure, lightweight.**
* **The product is very good in all aspects. Only thing is bad which is not shine on blazer.**
* **Very good value for money and with almost all the useful features. This washing machin is the only one at this price point that comes with an in built heater to handel stains better. Design, Features and Price are perfect. Overall a good bar get at around 12k."**
* **Definitely vote for this product. Load for cleaning vessels is cut off. I'm happy with the purchase. Hard stains will not be removed until hand scrubbed. I guess this is common with every dishwasher. I've been using this past 5 months. I use fortune detergent which is much economical than finish products. Only fault I find is when there is power cut during the wash time it starts the cycle but during dry time it doesn't start from drying time and finish the cycle. That time it doesn't dry the...**
* **Good and easy to use..**
* **Products are good. Go for it.**

**Samples of 5 star**

* **good  product i am proud of this long lasting super bajaj is good product seel thanks**
* **I really enjoy it love the sound and it's really easy.**
* **Amazing results....truly a very convienent and best for small families use....keep it up marq...,best quality ð"**
* **Very good product at this price range âºï¸âºï¸..**
* **These earphones may be best in their price segment.These earphones may not be appealing to you into the first time....as it happened with me...but after some after a day or two they feel real awesome...Great BassGreat Sound ClarityYou may fail the authenticity test....as I did..though....my were original...so don't panic... Send them a mail with the required info they ask you...and you will get it authenticated.**
* **Good product and price is low.**