AUTOMATED REVIEW SYSTEM

1.Project Overview

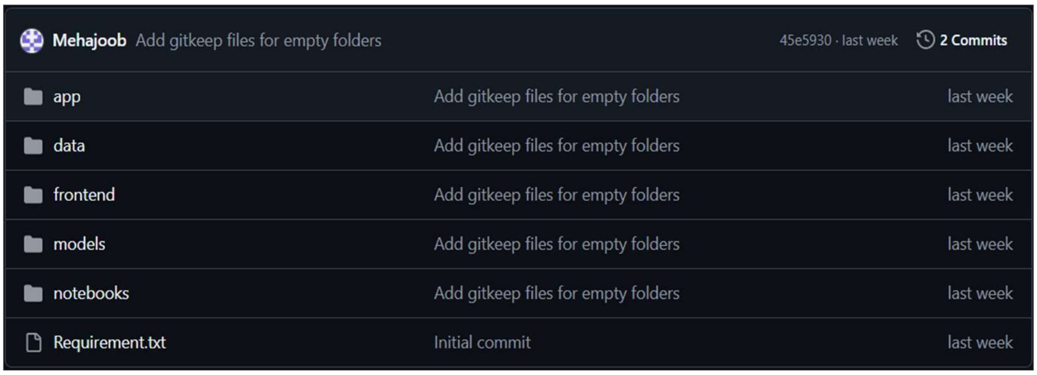
The Automated Review Classification System is a Natural Language Processing (NLP) project developed using the Amazon Fine Food Reviews dataset. The primary objective is to interpret customer opinions and automatically categorize each review into a rating scale from 1 to 5. The workflow includes comprehensive text preprocessing, TF-IDF feature extraction, and sentimentoriented NLP techniques to capture underlying opinion patterns. By transforming raw textual feedback into structured insights, the model helps assess customer satisfaction trends and supports decision-making for business improvement and product enhancement.

2. Environment Setup

The project was executed in Python 3.10 using Jupyter Notebook. Core libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn supported data handling, visualization, and model development. NLTK was used for text preprocessing, and Imbalanced-Learn helped address rating imbalance. All dependencies were managed using pip within an Anaconda virtual environment.

3.GitHub Project Setup

3.1 Created GitHub Repository : Automated-Review-Rating-System Structure of directory



4.Data Collection

The dataset was sourced from Kaggle, containing Amazon customer feedback and associated star ratings. After importing the dataset, unnecessary attributes such as product IDs, profile details, and timestamps were removed, retaining only the review text and rating score for model training.

The distribution of ratings in the collected data is as follows:

Rating Count

5 ★ 363,122

4 ★ 80,655

3 ★ 42,640

2 ★ 29,769

Rating Count

1 ★ 52,268

Dataset link:https://github.com/Mehajoob/Automated_review_rating_system/tree/main/data/balanced

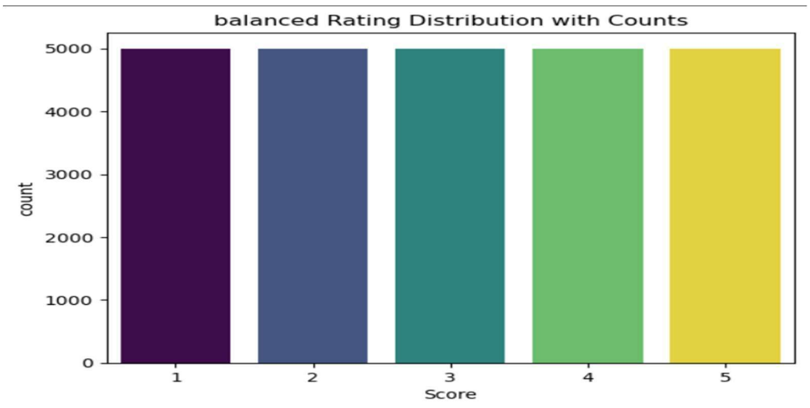
5. Data Preprocessing

The preprocessing stage covered both dataset cleaning and text refinement. Duplicate entries, missing values, and non-review text were removed to maintain reliability. All reviews were transformed to lowercase, and noise elements such as HTML tags, URLs, emojis, punctuation, and special symbols were stripped out. Stopwords were eliminated, and lemmatization was used to convert words to their base form. Additionally, extremely short and unusually long reviews were filtered out to ensure that only meaningful and readable text was retained for model training.

```
url_pattern = r"https?://\S+|www\.\S+" html_pattern = r"<.*?>"

emoji_pattern = r"[\U00010000-\U0010ffff]|[\u263a-\U0001f645]"

special_pattern = r"[^a-zA-Z0-9\s]" def clean_basic(text):

  text = text.lower()       #(converting to lowercase)     text =

re.sub(url_pattern, " ", text)    #(removing url)     text =

re.sub(html_pattern, " ", text)  #(removing html)     text =

re.sub(emoji_pattern, " ", text)  #(removing emoji)     text =

re.sub(special_pattern, " ", text)  #(removing special patterns)     doc =

nlp(text)

   tokens = [token.lemma_ for token in doc if token.text not in STOP_WORDS and len(token.text) >
1]     #(lemmetization)     cleaned = " ".join(tokens)     return cleaned

df["cleaned_text"] = df["Text"].apply(clean_basic)    #(applied to our text)
```

6.Balance Dataset

To address the skewed distribution of ratings, an equal sample was drawn from each class. A total of 5,000 reviews per rating (1–5) were selected, resulting in a balanced dataset of 25,000 reviews. This uniform distribution prevents the model from favoring majority ratings and supports more stable learning across all rating categories.



7. Natural Language Processing (NLP)

NLP techniques were implemented to convert raw review text into a structured format suitable for analysis. The process involved tokenization and stopword filtering to eliminate low-value and repetitive words that do not contribute significantly to sentiment understanding. After cleaning, the text was transformed into numerical features using TF-IDF vectorization, allowing the model to assess how important specific terms are across different reviews.

7.1 Stopword Removal

Stopword removal was carried out using spaCy's English stopword list, which filters common words such as the, is, at, and, etc. These words typically do not carry meaningful sentiment and can introduce noise into the dataset. By removing them, the model focuses on terms that better reflect opinion strength and emotional tone, leading to clearer sentiment signals in classification.

```
{'off', 'six', 'everyone', 'about', 'side', 'he', 'third', 'down', 'ten', 'me', 'i', 'up', 'under', 'we', 'more', 'those', 't
he', 'due', 'everywhere', 'just', 'top', 'an', 'been', 'out', 'mine', 'sixty', 'all', 'be', 'here', 'nothing', 'back', 'wel
l', 'amount', 'anyway', 'empty', 'hundred', 'twenty', 'five', 'even', 'but', 'forty', 'else', 'keep', 'am', 'together', 'some
one', 'few', 'last', 'four', 'please', 'anyone', 'still', 'except', 'put', 'via', 'less', 'show', 'name', 'own', 'get', 'on
e', 'front', 'give', 'say', 'bottom', 'part', 'something', 'go', 'do', 'will', 'have', 'over', 'may', 'on', 'us', 'take', 't
o', 'in', 'full', 'see', 'make', 'these', 'can', 'from', 'not', 'move', 'call'}
```

7.2 Lemmatization

Lemmatization is the process of converting words to their base or dictionary form (lemma) while preserving proper meaning and grammatical validity. For example:

- running → run
- better → good
- cars → car

This step ensures that different word variations are treated as a single term, improving vocabulary consistency and reducing dimensionality in TF-IDF representation.

Why Lemmatization Instead of Stemming?

- Lemmatization understands context, while stemming simply cuts word endings.
- Stemming can generate non-words (e.g., "studies" → "studi"), causing loss of meaning.
- Lemmatization uses linguistic rules and part-of-speech tagging to generate meaningful base forms.

Feature	Stemming	Lemmatization
Output type	Often truncated, non-dictionary forms	Proper dictionary words
Meaning preservation	Low	High

Technique	Removes suffixes mechanically	Uses vocabulary + grammar rules
Example	“studies → studi”	“studies → study”

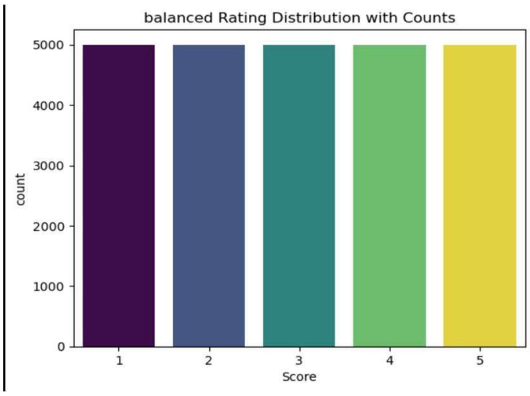
Because lemmatization maintains semantic accuracy, it was selected over stemming to support clearer sentiment interpretation and more reliable TF-IDF feature representation.

8. Data Visualization

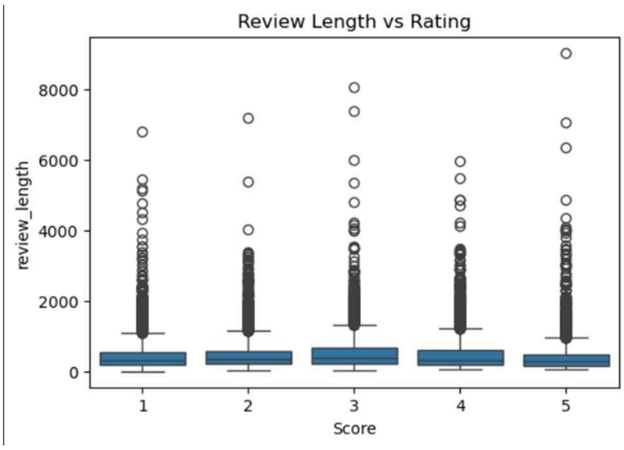
Data visualization was used to explore review patterns and rating distribution within the dataset. Bar charts clearly illustrated the imbalance across rating classes, with 5-star reviews dominating the majority. Word count histograms and box plots were utilized to examine text length variation and detect unusually short or lengthy entries. Additionally, a few sample reviews from each rating category were inspected to better understand sentiment tone and content differences between highly positive and negative feedback.

8.1 Balanced Data

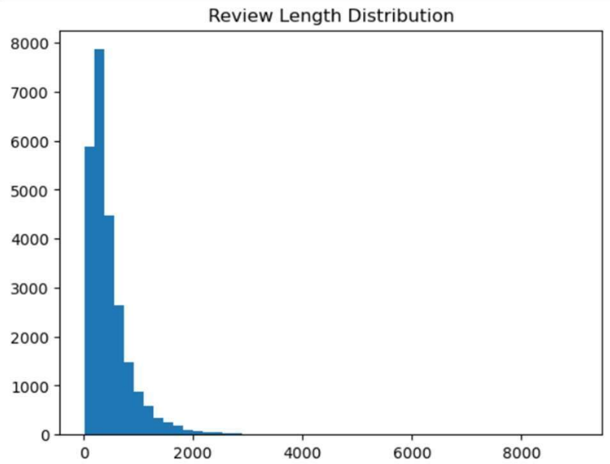
8.1.1 Rating Distribution



8.1.2 Review Length vs Rating Distribution (boxplot)



8.1.3 Review Length Distribution (Histogram)



8.1.4 Sample Reviews

==

Rating: 1 | Showing 10 sample reviews

==

Review 2326:

kikkoman mirin probably call mirin fu chomiryo literally mirin like season certainly hon mirin true mirin look ingredient list ll big away rice water corn syrup alcohol salt sure rice wine get rice little alcohol sweeten corn syrup mirin real mirin sweeten rice subject aspergillus oryzae fungus break rice starch sugar production sake mirin use fungus produce unfermented sugar sweeten essential japanese cooking wine look real thing asian market check label sugar corn syrup isn authentic eden food make widely available good mirin organic brown rice rice koji aspergillus oryzae pure water sea salt salt mirin sell authentic add calssify wine cook wine avoid alcohol taxis favor read label opt authenticity win regret

Review 152:

buy couple food couldn finish throw away second resemble food raman kick butt absolutely disgusting ew gross don waste money way

Review 4221:

sickenly sweet artificial tasting getting rid box mix regular cup buy

Review 1437:

product worth throw past flavor oil product terrible

Review 4939:

product disappointing order tropical trio regularly decide try new open bag see greasy pretty slimy excuse fruit bag repulsive look taste way look greasy gross sure exactly dry fruit get greasy cup tea product plan order

Review 241:

purchase large can chili price ridiculously cheaply time plan split can small portion freeze small portion large number meal open look little different hormel chili past bean mash large chunk cub meat chili normally meat resemble little piece ground meat weird unappetizing chunk chili chili taste taste like slightly burn fry bean night eat day numerous trip bathroom win detail wasn pleasant day later open second hop time thing second mashed bean meat chunk toss large can chili use maybe bad batch scrap old chili residue pot throw leftover meat chunk can send needless purchase product

Review 4719:

horrified find product china realize month old baby like watch closely piece easily break dissolve quickly choke luckily able cough piece cracker

Review 229:

item arrive completely enclose package inside package damage original voss box tear drop couple bottle leak bottle place box kind way pay good money receive order problem correct

Review 1131:

love terra yukon gold potato chip think order small packaging carton arrive 24 small bag product stale expiration date package month ahead date receive chip small bag contain chip looked burn disappointed

Review 3200:

stuff bad crap packet word describe walden farm dressing sauce print amazon clorox mouth rinse need taste chemical bombshell

==

Rating: 2 | Showing 10 sample reviews

==

Review 9737: agree previous reviewer listen boston terrier chew eat entire bone minute good

treat chew toy

Review 7322:

white rock star drinkable drink chalky aftertaste fairly wretched flintsone grape vitamin taste like chewable flintstone grape vitamin chalky aftertaste ll love rockstar zero carb energy drink taste

Review 8677:

weak replace time brand trash bag reason didn star cause try produce biodegradable product

Review 6215:

love peanut butter chocolate plain peanut butter good taste artificial disconcert pepto bismol pink color suggest kind

Review 8039:

trap job nearly trap difficult set require use special tool come trap tool trap pain set compare set place active run job issue trap design color difficult disance impossible tell spring remeber check periodically recommend poke tool ground trap know

Review 8255:

disappoint quality order package burn dark bitter sesame seed actually black order variety serve guest

Review 9253:

show minus expect picture cocoa matte finish greasy compare similar chocolate cover almond ve sample lot brand look valor want good almond chocolate

Review 6772:

buy pear think good snack unfortunately tasty try away didn eat toss remainder nice packaging quick service

Review 7601:

disappointed lack concern customer product clear warning affect people gi upset gas etc taste isn bad think product sugar alcohol reduce effect considerably order product state sugar free low calorie serve size 17 piece 41 serve calorie 160 sugar version effect sorry disappointed product

Review 6560:

half star allow doesn deserve star star color texture wrong doesn look like picture look cook taste average nice garlic taste nice france appreciate give little flavor home regularly

==

Rating: 3 | Showing 10 sample reviews

==

Review 11494:

tasty drink natural small ounce contain relatively high 140 calorie roughly number calorie 12 ounce coke coke 50 large volume like sweet bit sweet nice mixer adult beverage haven try mix probably nice fruity alcoholic drink price isn bargain nearly writing wouldn compelling value like doubt purchase leave bit linger metallic aftertaste wasn thrill don mind calorie enjoy beverage sweet look natural drink worth consider

Review 12761:

buy bar tale oatmeal lower cholesterol good buy information bar contain corn syrup label bar ingredient clear day corn syrup ingredient want stay away unhealthy ingredient shock list bar package

Review 10465:

10 year old love soup darned time find locally anymore local 99 ranch market stock find local 99 packet cheap buy pack 12 amazon quick easy boil cup water mix remove heat stir beat egg serve star cheap

Review 12494:

like have snack bar quick easy especially excited try underwhelmed taste packaging say sweet salty live harmoniously sweet salty kind bland fair friend try bar half like care half like like fact bar organic peanut gluten free personally wish little flavor

Review 10676:

make good bread ve disappointed single box ve open date box far have plastic packaging inside seal vertical seam apparently adhesive seal plastic bag contain mix didn hold heat seal seam technique didn work

Review 11102:

absolute love jelly get way price start buy 95 oz jar dismay open recent shipment cost 10 discover jar contain ounce barely large sample wonderful flavor rationalize spending money jelly disappointed

Review 10065:

teenager diagnose celiac year love stuff shell lasagna ravioli don great treat expensive eat bag serve approximately 12 bag treat get awhile thank conte make special work price star

Review 10970:

big fan frosted mini wheat cinnamon excited cereal review unfortunately think kellogg good thing well new mini wheat consist bite sized frosted bit shape like pillow bite reveal wheaty layer brown sugar flavor inside significant fiber make good choice want stay little bite version change formula way make bite small add cinnamon like cinnamon nice background flavor real punch like afraid change flavor small size misstep opinion mean surface area wheat layer immediately turn mush great thing normal mini wheat outside soft inside stay crunchy long time little bite instead turn quickly porridge hate cereal general fan change old formula flavor limited time stuck traditional flavor instead think bad experiment

Review 13810:

don generally buy fat free product usually taste like crap brownie surprising good fat free brownie little sweet come major sweet tooth eat taste great maybe ll time start ll notice slight aftertaste kind like miss important fat aren bad ride sweet craving definitely delicious real homemade brownie

Review 11366:

title say aftertaste taste stale vanilla big sugar clump jar think moisture get point

==

Rating: 4 | Showing 10 sample reviews

==

Review 16494:

base review try jerky buffalo bill think order moist tender good flavor

Review 17760:

personal preference course like flavor tea variety flavor tea pre sweeten slightly definitely sweet like product like fact sweeten drink 50 calorie serve serving relatively small ounce tea plus additional volume melt ice make 12 ounce serve

Review 15468:

cinnamon crunch yummy albeit tad sweet like unsweetened almond milk sliced banana awhile sale store buy good product cascade farm

Review 17494:

detract star box pay   yummy

Review 15680:

good stuff   nice spice   update delicious corn add

Review 16105:

ve new clear product day week   product similar week   honestly tell difference   guess formulate fight dandruff issue time   clear product use head   shoulder   pro   nice design bottle dispense cleanly leave mess bottle clump time cap easy use clear product   smell clean strong ok smell opinion   clean oily hair clear clean quickly   lather nicely   lather well   con   issue product small leak shipping   product fault aware buy liquid have ship leak   fairness don blame amazon product packaging leak small mess clean thing happen ship risk face purchase way interesting   shampoo feel look bit like elmer glue little hand sort fun

Review 15065:

order box lyon tea 160 bag   great everyday tea expiration date close   month away happy need month drink tea

Review 15973:

dog love   dog 17 pound 22 pound   hard chew easily break tooth   wife bring workbench break hammer dog break sharp actually   smell bit like urine human dog love   favorite treat right dry liver   careful

Review 18369:

stuff perfect camping trip lazy cut tofu green onion mix proper miso paste   yo son love nice salty willing open tub tofu cut half ounce stuff   perfect make convenience

Review 16367:

haven try lot water additive try different flavor mio didn care taste mio costly size bottle oz real nutritional value try fruit punch flavor pretty good jump joy taste favorite far try taste remind tahitian treat soda sweetener like come large 12 oz bottle value important price fairly reasonable add vitamin b6 b5 b12 b3 zinc chromium magnesium unknown caffeine small dose serve hardly noticeable overall like product look forward try flavor

==

Rating: 5 | Showing 10 sample reviews

==

Review 24009: recieve 12 jar find jar seal properly replacement jar fine product look taste like real honey

Review 20145:

recently discover tea develop leaky gut syndrome see shelf food believe fast work 15 minute feel wonderful morning awake bad cramping immediately cup half hour fine stuff miracle tell

Review 23075:

review tofu shirataki noodle angel hair shape repetitive read review read review difference noodle wide hard texture actuality appear way noodle wide time hear shirataki noodle year ago decide try go fiesta get tofu shirataki noodle pasta fettuccine shape shirataki konjac yam add tofu give hard texture noodle tofu add calorie time shirataki noodle cook like regular pasta spaghetti shape turn gel year ago learn bad mistake change noodle time correctly try eat pasta sauce didn work flavor merry accustom hard texture pasta try fettuccine shape shirataki thing come mind lo mein fettucini shaped noodle soy sauce ginger stir fry veggie realize noodle well asian style recipe remember rinse product noodle flavor take flavor cook sure overcook fine personally cook long minute update research shirataki noodle order expand idea recipe find reason shirataki noodle turn mush expire shirataki noodle good year edible turn mush cook long noodle apparently shrink thin people like go store tofu shirataki noodle look expiration date lo behold expire year see store expire year non expire shirataki noodle alot chewy cook long mom stir fry try bake great idea lazy use boullion cube noodle work fine let sit seasoning hour flavor soup additional update find government allow company list product calorie calorie bag calorie half bag entire bag calorie research see pickle calorie pickle see entire pickle calorie

Review 24844:

dog love chew like buy case

Review 22585:

find treat cat travel rv cat come rv anytime want noise like open bag treat come run turn nose live canada buy order amazon

Review 20737:

flour expensive flour see good quality flour purchase bread pizza come well bread flour good additive worth cost

Review 23449:

go work start pump baby week old pump time day work nurse night didn problem get oz time milk supply begin decrease baby month old search internet find fenugreek take week decide buy try eat drink didn help pill time day see slight change milk supply second day take week different oz pump work great wish buy week soon baby didn formula

Review 21605:

dingo bone traditionally overpriced order amazon free shipping cheap way course dog love treat long time plenty meat inside easy open long house

Review 23762:

recently buy power buttermilk shipping fast product look find powered buttermilk store close feed baby puppy Review 21727:

drink good drink hot eay cup total enjoymet make glad get

9. Train–Test Split

The train–test split is an essential step for assessing how well the model generalizes to new, unseen data. In this stage, the dataset is separated into two parts:

1. Training Set: Used for model learning and parameter optimization, generally covering 70% of the data.
2. Testing Set: Used strictly for performance evaluation on unseen reviews, making up the remaining 30%.

Configuration Details

- test_size: Indicates the portion of data allocated for testing (0.30 was used in this project).
- random_state: Sets a fixed seed value to ensure the split remains consistent across multiple runs.
- shuffle: Randomizes the dataset prior to splitting to prevent any ordering bias; enabled by default.
- stratify: Ensures equal representation of all rating categories in both sets, which is especially critical when working with previously imbalanced ratings.

This approach guarantees that the model is trained effectively while still being tested fairly on data it has never encountered.

```
# splitting to target and feature
x = df_filtered['Text']   #(f)
y = df_filtered['Score']  #(t)
x.head()
# Splitting into Train and Test 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)
```

10. Vectorization

Once the reviews were cleaned and lemmatized, they were transformed into numerical form using TF-IDF vectorization. This method assigns weights to words based on how frequently they appear within a review while considering how rare they are across the entire dataset. As a result, the feature matrix focuses on words that carry meaningful sentiment information rather than simply frequent terms.

10.1 TF-IDF (Term Frequency–Inverse Document Frequency)

TF-IDF is a numerical statistic used to reflect how important a word is to a document in a collection (corpus).
It reduces the weight of commonly occurring words and increases the weight of rare but informative terms.

Equation

$$\text{TF-IDF}(t,d)=\text{TF}(t,d)\times\text{IDF}(t)$$

10.2 Why TF-IDF Instead of Bag of Words

While both TF-IDF and Bag of Words (BoW) convert text into numerical vectors, TF-IDF was chosen due to its ability to differentiate between common and informative words.

Key Reasons for Choosing TF-IDF

- It down-weights common words that may not hold specific sentiment value.
- It highlights unique, impactful terms that distinguish one rating level from another.
- TF-IDF produces a more informative and discriminative feature set, which improves model accuracy.
- Compared to BoW, TF-IDF reduces noise, especially in long review texts where filler words repeat often.

10.2 Summary

- Bag of Words: counts words but treats all with equal importance.
- TF-IDF: assigns higher value to words that matter sentiment-wise, improving classification capability.

```
from sklearn.feature_extraction.text import TfidfVectorizer vectorizer =
TfidfVectorizer(max_features=10000, ngram_range=(1,2)) x_train_vect
= vectorizer.fit_transform(x_train) x_test_vect =
vectorizer.transform(x_test)


print("Training data shape:", x_train_vect.shape) print("Test
data shape:", x_test_vect.shape)
```

11. Model Building (Balanced Dataset)

After balancing the dataset to ensure equal representation of all rating classes (1–5), multiple machine learning models were trained and evaluated. The objective was to compare linear and non-linear classifiers on TF-IDF features and analyze their generalization performance.

The following models were implemented:

- Logistic Regression
- Random Forest Classifier
- Support Vector Machine (Linear SVM)

All models were trained on the **balanced training set** and evaluated on the **balanced test set**.

11.1 Feature Representation

Text reviews were converted into numerical form using **TF-IDF Vectorization**.

Why TF-IDF?

Reduces the influence of frequently occurring but less informative words

Highlights words that are important to specific reviews

Works well with linear models for text classification

The same vectorizer was used consistently for both training and testing to avoid data leakage.

11.2 Logistic Regression (Balanced)

Model Description

Logistic Regression is a linear classification algorithm that estimates class probabilities using a softmax function for multi-class problems.

Configuration Used:

Solver: newton-cg

Max Iterations: 1000

Random State: 42

Performance

Training Accuracy: ~0.80
Testing Accuracy: ~0.52

Observations:

The model performs well on training data, indicating it learns meaningful patterns.

Test performance drops, showing moderate generalization.

Middle ratings (2, 3, 4) are harder to predict compared to extreme ratings (1 and 5).

Macro and weighted F1-scores indicate balanced performance across classes but limited expressive power due to linearity.

Code:

```
from sklearn.linear_model import LogisticRegression
logit = LogisticRegression(solver='newton-cg')
logit.fit(x_train_vect,y_train_balanced)

y_pred_train = logit.predict(x_train_vect)
y_pred_test = logit.predict(x_test_vect)

from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
print("train:",accuracy_score(y_train_balanced,y_pred_train))
print("test:",accuracy_score(y_test_balanced,y_pred_test))

print("train:",classification_report(y_train_balanced,y_pred_train))
print("test:",classification_report(y_test_balanced,y_pred_test))
```

11.3 Random Forest Classifier (Balanced)

Model Description

Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting.

Configuration Used:

Number of Trees (n_estimators): 300

Class Weight: balanced

Max Features: sqrt

Bootstrap Sampling: Enabled

Random State: 42

Parallel Processing: Enabled

Performance

Training Accuracy: 1.00
Testing Accuracy: ~0.48

Observations:

Perfect training accuracy indicates **severe overfitting**.

The model memorizes training samples but fails to generalize.

TF-IDF features are high-dimensional and sparse, which tree-based models struggle with.

Performance on test data is comparable or worse than linear models.

Code:

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(class_weight='balanced', n_estimators=300,
    max_depth=None,
    min_samples_split=2,
    min_samples_leaf=1,

    max_features='sqrt',
    bootstrap=True,
    random_state=42,
    n_jobs=-1)
rf.fit(x_train_vect,y_train_balanced)

y_pred_train_1 = rf.predict(x_train_vect)
y_pred_test_1 = rf.predict(x_test_vect)

from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
print("train:",accuracy_score(y_train_balanced,y_pred_train_1))
print("test:",accuracy_score(y_test_balanced,y_pred_test_1))

print("train:",classification_report(y_train_balanced,y_pred_train_1))
print("test:",classification_report(y_test_balanced,y_pred_test_1))
```

11.4 Support Vector Machine (Linear SVM)

Model Description

Linear SVM attempts to find an optimal hyperplane that maximizes the margin between classes. It is particularly effective for text classification tasks.

Configuration Used:

Kernel: Linear (LinearSVC)

Default regularization parameters

Performance

Testing Accuracy: ~0.48

Observations:

Performance is similar to Random Forest on test data.

Performs better on extreme classes (1 and 5) than neutral ratings.

Requires careful hyperparameter tuning (C) to improve results.

Sensitive to class overlap common in review rating data.

Code:

```
from sklearn.svm import LinearSVC


svm_model = LinearSVC()

svm_model.fit(x_train_vect, y_train_balanced)


y_pred_svm = svm_model.predict(x_test_vect)


print("Accuracy:", accuracy_score(y_test_balanced, y_pred_svm))

print("\nClassification Report:\n", classification_report(y_test_balanced, y_pred_svm))
```

11.5 Model Comparison Summary

Model	Train Accuracy	Test Accuracy	Overfitting	Suitability for Text
Logistic Regression	~0.80	~0.52	Low	High
Random Forest	1.00	~0.48	Very High	Low
Linear SVM	—	~0.48	Moderate	High

11.6 Key Insights

Balancing the dataset improved fairness across classes but did not guarantee high accuracy.

Linear models (Logistic Regression, SVM) are more suitable for TF-IDF features than tree-based models.

Random Forest severely overfits due to high-dimensional sparse input.

Predicting middle ratings (2, 3, 4) remains challenging due to semantic overlap in reviews.

11.7 Final Conclusion

Among the tested models on the balanced dataset, **Logistic Regression demonstrated the best trade-off between performance and generalization**. While Random Forest achieved perfect training accuracy, it failed to generalize, and Linear SVM showed comparable but slightly lower performance.