Final Project Report Template

Analysis Of Amazon Cell Phone Reviews

1.INTRODUCTION

1.1 PROJECT OVERVIEWS

The project focuses on the comprehensive analysis of customer reviews for cell phones available on Amazon. With the rise of e-commerce, online reviews have become a critical source of information for consumers, influencing their purchasing decisions. Amazon, as one of the largest online marketplaces, offers a vast collection of user-generated reviews for various products, including cell phones. This project aims to extract, analyze, and derive meaningful insights from these reviews to understand customer sentiments, key factors influencing satisfaction or dissatisfaction, and emerging trends in consumer preferences. The analysis involves leveraging natural language processing (NLP) techniques to process the text of reviews, identifying sentiment (positive, neutral, or negative), and extracting common themes or product features that frequently appear in reviews, such as battery life, camera quality, or performance.

Additionally, the project investigates the impact of different variables such as product pricing, brand popularity, and review ratings on customer satisfaction. This analysis can provide valuable insights for both consumers and businesses. For consumers, it offers guidance on product selection based on aggregated feedback from verified users. For businesses and manufacturers, it highlights areas for product improvement, customer service, and marketing strategies by identifying strengths and weaknesses in their offerings. By utilizing machine learning algorithms and data visualization tools, the project presents data-driven insights in a user-friendly format, making it easier to track trends over time and assess the overall perception of different cell phone models on Amazon. In conclusion, this project provides a detailed understanding of customer sentiment and preferences, contributing to enhanced decision-making for both buyers and sellers in the competitive cell phone market.

This project not only aims to provide a granular understanding of customer opinions but also seeks to predict future trends in cell phone reviews. By identifying emerging patterns in customer preferences, we can offer valuable insights for manufacturers and retailers to better tailor their products to customer needs. Moreover, this analysis can serve as a benchmark for comparing various phone models and brands, helping consumers make more informed purchasing decisions based on data-driven insights.

1.2. OBJECTIVES

The project focuses on the comprehensive analysis of customer reviews for cell phones available on Amazon. With the rise of e-commerce, online reviews have become a critical source of information for consumers, influencing their purchasing decisions. Amazon, as one of the largest online marketplaces, offers a vast collection of user-generated reviews for various products, including cell phones. This project aims to extract, analyze, and derive meaningful insights from these reviews to understand customer sentiments, key factors influencing satisfaction or dissatisfaction, and emerging trends in consumer preferences. The analysis involves leveraging natural language processing (NLP) techniques to process the text of reviews, identifying sentiment (positive, neutral, or negative), and extracting common themes or product features that frequently appear in reviews, such as battery life, camera quality, or performance.

- **Sentiment Analysis**: The project aims to determine the overall sentiment of customer reviews—whether they are positive, negative, or neutral. This will help identify the general consumer perception of various cell phone models and brands.

- **Identification of Key Features**: By analyzing the content of reviews, the project seeks to uncover the most commonly mentioned product features, such as battery life, camera quality, display, or software performance. This will allow us to understand which aspects of a phone are most important to customers and which contribute most to satisfaction or dissatisfaction.

- **Detection of Trends and Patterns**: The project aims to track shifts in consumer preferences over time by identifying trends and recurring patterns in the reviews. This could involve changes in popular brands, evolving customer expectations, or the impact of new product releases.

- **Impact of Review Ratings**: Another objective is to assess the correlation between review ratings (e.g., star ratings) and the actual sentiment expressed in the review text. This will help evaluate whether a high or low star rating always corresponds to positive or negative feedback, or if there are discrepancies.

- **Comparison of Brands and Models**: The project will compare different cell phone brands and models based on aggregated review data. The goal is to highlight which brands or models are consistently rated higher by customers, and to pinpoint the features that contribute to these ratings.

- **Actionable Insights for Manufacturers and Retailers**: Finally, the project aims to provide actionable recommendations for manufacturers and retailers. By identifying common complaints or praises in reviews, companies can make data-driven decisions to improve their products, customer service, and marketing strategies.

Through these objectives, the project seeks to offer a deep understanding of customer sentiment, product performance, and market trends in the Amazon cell phone category, ultimately facilitating better decision-making for both businesses and consumers.

2.Project Initialization and Planning Phase

2.1. Define Problem Statements (Customer Problem Statement Template):

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for your customers' challenges. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Example:

I AM	I WANT TO	BUT	BECAUSE	WHICH MAKES ME FEEL
A data analyst reviewing Amazon cell phone reviews	Analyze the sentiment of the reviews (positive, negative, neutral)	The dataset contains a large volume of text and unstructured data	Reviews are often lengthy, unstructured and it's hard to extract useful insights from	Motivated to extract meaningful insights from the data

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A customer seeking to purchase a cell phone on Amazon.	Decide which phone to buy based on customer reviews.	The reviews are overwhelming, and it's difficult to discern which are helpful or relevant.	There are too many reviews, with varied sentiments, making it hard to determine the phone's actual quality.	Confused and uncertain about the best purchasing decision.

2.2. Project Proposal (Proposed Solution)

The proposed solution involves analyzing Amazon cell phone reviews using natural language processing techniques to identify key sentiment trends, product features, and customer preferences. This analysis will provide valuable insights into customer satisfaction and help improve product offerings.

Project Overview	
Objective	The primary objective of this project is to perform sentiment analysis and extract key insights from Amazon customer reviews on cell phones, identifying trends, customer satisfaction levels, and product features that are frequently discussed.
Scope	This project will analyze a dataset of Amazon cell phone reviews to provide valuable feedback to retailers and manufacturers. It will focus on understanding customer sentiment (positive, negative, neutral) and analyzing the frequency of specific terms (e.g., battery life, camera, performance). The scope will be limited to reviews in English for products in the cell phone category.
Problem Statement	
Description	This analysis involves examining Amazon customer reviews of cell phones to assess sentiment, highlight common complaints or praises, and identify trends in product performance, helping guide improvements and inform consumer decisions.
Impact	Solving this problem will provide manufacturers with actionable insights into consumer preferences and recurring issues. It will also allow customers to make informed purchasing decisions based on aggregated reviews and sentiments.
Proposed Solution	
Approach	The project will use natural language processing (NLP) techniques for sentiment analysis. Text pre-processing

	methods will clean the review text, and a machine learning model (such as Naive Bayes or a neural network) will classify reviews into different sentiment categories. Additionally, the project will utilize topic modeling to identify common themes.
Key Features	- Sentiment analysis (positive, negative, neutral classification) - Topic modelling for key aspects (e.g., camera quality, battery life) - Word cloud visualizations to highlight frequent terms - Insights into the most discussed cell phone brands/models

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	2 x NVIDIA V100 GPUs for processing
Memory	RAM specifications	16 GB RAM for data processing tasks
Storage	Disk space for data, models, and logs	1 TB SSD for storing reviews, models, and logs
Software		
Frameworks	Python frameworks	Flask for web interface

Libraries	Additional libraries	scikit-learn, pandas, NumPy, NLTK for sentiment analysis
Development Environment	IDE, version control	Jupyter Notebook for development, Git for version control
Data		
Data	Source, size, format	Amazon product reviews dataset, estimated size: 100,000 reviews, format: CSV

2.3. Initial Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Team Members	Task Start Date	Sprint End Date (Planned)
Sprint-1	Prerequisites	USN-1	Install Anaconda	High	Vaibhava Lakshmi	29-09-2024	29-09-2024
Sprint-1	Prerequisites	USN-2	Python Packages Installation	High	Vaibhava Lakshmi	30-09-2024	30-09-2024
Sprint-2	Data Collection	USN-3	Download The Dataset	High	Tirumala	1-10-2024	1-10-2024
Sprint-3	Feature Selection	USN-4	Import Libraries	Medium	Sai Sushmitha	2-10-2024	2-10-2024
Sprint-3	Feature Selection	USN-5	Reading The Dataset	Medium	Adithya	3-10-2024	3-10-2024
Sprint-3	Feature Selection	USN-6	Checking The Null Values	Medium	Tirumala	4-10-2024	4-10-2024

Sprint-3	Feature Selection	USN-7	Drop The Columns	Medium	Vaibhava Lakshmi	5-10-2024	5-10-2024
Sprint-3	Feature Selection	USN-8	Drop The columns -2	Medium	Vaibhava Lakshmi	5-10-2024	5-10-2024
Sprint-3	Feature Selection	USN-9	Split The Data into Input and Output	High	Tirumala	6-10-2024	6-10-2024
Sprint-4	Text Preprocessing	USN-10	Import Required Libraries	Medium	Sai Sushmitha	7-10-2024	8-10-2024
Sprint-4	Text Preprocessing	USN-11	Text Cleaning	High	Adithya	8-10-2024	9-10-2024
Sprint-4	Text Preprocessing	USN-12	Text Cleaning-2	High	Tirumala	9-10-2024	10-10-2024
Sprint-4	Text Preprocessing	USN-13	Text Cleaning -3	High	Vaibhava Lakshmi	10-10-2024	10-10-2024
Sprint-4	Text Preprocessing	USN-14	Text Cleaning-4	High	Adithya	10-10-2024	10-10-2024
Sprint-4	Text Preprocessing	USN-15	Text Cleaning-5	High	Adithya	10-10-2024	11-10-2024
Sprint-5	Model Building	USN-16	Train The Model	Medium	Vaibhava Lakshmi	11-10-2024	11-10-2024
Sprint-5	Model Building	USN-17	Building An Index.HTML File	High	Sai Sushmitha	11-10-2024	12-10-2024
Sprint-5	Model Building	USN-18	HTML Page	Medium	Sai Sushmitha	14-10-2024	14-10-2024
Sprint-5	Model Building	USN-19	Build Python Code	High	Tirumala	15-10-2024	16-10-2024
Sprint-5	Model Building	USN-20	Run The App	High	Tirumala	17-10-2024	17-10-2024
Sprint-6	Train The Model on IBM	USN-21	Register For IBM Cloud	High	Sai Sushmitha	18-10-2024	18-10-2024
Sprint-6	Final Project Document	USN-22	Creation Of Final Documentation	High	Vaibhava Lakshmi	19-10-2024	19-10-2024

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan & Raw Data Sources Identification

For the analysis of Amazon cell phone reviews, the data will be collected primarily from Amazon’s customer review sections using web scraping techniques or publicly available datasets. This includes gathering textual review data, star ratings, timestamps, customer metadata and product-specific information like brand, model, and features. The scraping process will ensure a diverse and comprehensive dataset by covering reviews across multiple brands and phone models.

The data for the project will be collected from publicly available customer reviews on Amazon. This includes user ratings, comments, and metadata for cell phone products. The data will be gathered via web scraping or from pre-existing datasets on platforms like Kaggle or public APIs. The identified raw data source is an Amazon cell phone review dataset in CSV format, which is publicly accessible. It contains all necessary features such as review content, ratings, and product metadata for analysis.

Data Collection Plan

Section	**Description**
Project Overview	This project aims to analyze customer reviews of cell phones sold on Amazon, applying machine learning techniques to extract insights regarding user satisfaction, common issues, and product performance trends.
Data Collection Plan	The data will be collected from Amazon's public product review dataset, focusing specifically on cell phone reviews. Methods include web scraping or accessing public datasets via APIs.
Raw Data Sources Identified	Amazon review dataset containing cell phone review texts, ratings, and metadata such as review date and product information.

Raw Data Sources Report:

Source Name	Description	Location/URL	Format	Size	Access Permissions
Kaggle Dataset	This dataset contains reviews, ratings, and metadata of various cell phones sold on Amazon, including customer feedback, review dates, star ratings, and product information.	https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews	CSV	5 MB	Public

3.2 Data Quality Report Template

The Data Quality Report outlines data quality issues from the selected Amazon cell phone reviews dataset, including their severity levels and the proposed resolution plans. This report helps systematically identify and address any discrepancies or issues with the dataset to ensure high-quality input for the machine learning model.

Data Source	Data Quality Issue	Severity	Resolution Plan
Amazon Cell Phone Reviews Dataset	Duplicate reviews based on "Review ID" and "Review Text"	Moderate	Identify and remove duplicate entries to ensure uniqueness in reviews.
Amazon Cell Phone Reviews Dataset	Star rating skew towards 5 stars (rating imbalance)	Moderate	Apply weighting or filtering to account for bias and provide a balanced analysis
Amazon Cell Phone Reviews Dataset	Outliers in "Helpful Votes" field (unusually high counts)	High	Investigate and flag for potential manipulation or spam, potentially removing those reviews.

3.3 Data Preprocessing

The data preprocessing phase is crucial for ensuring the quality and consistency of the Amazon cell phone reviews dataset. This process involves cleaning the raw data by removing duplicates, handling missing values, and filtering out irrelevant information like promotional content or non-review text. Text data will undergo normalization techniques, such as tokenization, lowercasing, and removing stopwords, punctuation, and special characters, to prepare it for natural language processing (NLP). Sentiment labels and product features will be extracted, and reviews will be categorized based on factors like rating, date, and product model, ensuring the dataset is structured and ready for analysis.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>

<u>Dimension:</u> 67986 rows × 17 columns

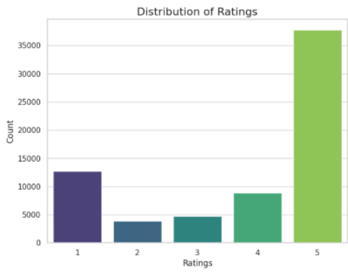
<u>Descriptive statistics:</u>

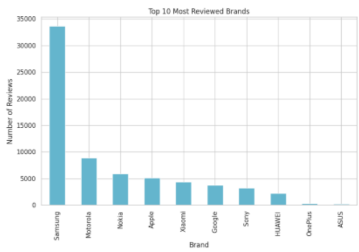
<pre>
[ ] print(merged_df.describe())

          rating_x helpfulVotes   rating_y totalReviews   price \
count 67986.000000 27215.000000 67986.000000 67986.000000 67986.000000
mean   3.807916   8.229690   3.766826   373.742800   222.050506
std   1.582906   31.954877   0.429197   262.560876   188.863986
min   1.000000   1.000000   1.000000   1.000000   0.000000
25%   3.000000   1.000000   3.500000   153.000000   103.980000
50%   5.000000   2.000000   3.800000   336.000000   179.990000
75%   5.000000   5.000000   4.100000   558.000000   300.550000
max   5.000000   990.000000   5.000000   983.000000   999.990000

       originalPrice
count   67986.000000
mean   84.057634
std   201.923373
min   0.000000
25%   0.000000
50%   0.000000
75%   0.000000
max   999.990000
</pre>

</td></tr>
<tr><td>Univariate Analysis</td><td>





</td></tr>
<tr><td>Bivariate Analysis</td><td>—</td></tr>
</table>

<table>
<tr><td>Multivariate Analysis</td><td>
</td></tr>
<tr><td colspan="2">Data Preprocessing Code Screenshots</td></tr>
<tr><td>Loading Data</td><td>LOADING THE DATASET
<pre><code># Load the datasets (replace the file paths with your actual file paths)
items_df = pd.read_csv('/content/drive/MyDrive/20191226-items.csv')
reviews_df = pd.read_csv('/content/drive/MyDrive/20191226-reviews.csv')

# Merge datasets based on 'asin' to combine product info with reviews
merged_df = pd.merge(reviews_df, items_df, on='asin')

View the first few rows to check the loaded data
print(merged_df.head())</code></pre></td></tr>
<tr><td>Handling Missing Data</td><td><pre><code>print(merged_df.isnull().sum())

asin   0
name   3
rating_x   0
date   0
verified   0
title_x   29
body   26
helpfulVotes   40771
brand   200
title_y   0
url   0
image   0
rating_y   0
reviewUrl   0
totalReviews   0
price   0
originalPrice   0
dtype: int64</code></pre></td></tr>
</table>

<table>
<tr>
<td>Data Preprocessing</td>
<td>

TEXT PREPROCESSING

<pre>
# Define a function to preprocess the review text
def preprocess_text(text):
    # Check if the text is a string
    if isinstance(text, str):
        # Convert to lowercase
        text = text.lower()
        # Remove punctuation using NLTK's tokenizer
        text = nltk.RegexpTokenizer(r'\w+').tokenize(text)
        # Re-join words into a single string
        return ' '.join(text)
    else:
        # Handle non-string values (e.g., return an empty string or a placeholder)
        return ''

Apply preprocessing to the review body column
merged_df['cleaned_review'] = merged_df['body'].apply(preprocess_text)

# View the preprocessed reviews
print(merged_df[['body', 'cleaned_review']].head())
</pre>

<pre>
                                                body \
0 I had the Samsung A600 for awhile which is abs...
1 Due to a software issue between Nokia and Spri...
2 This is a great, reliable phone. I also purcha...
3 I love the phone and all, because I really did...
4 The phone has been great for every purpose it ...

                                      cleaned_review
0 i had the samsung a600 for awhile which is abs...
1 due to a software issue between nokia and spri...
</pre>
</td>
</tr>
<tr>
<td>Visualizations</td>
<td>
<pre>
import matplotlib.pyplot as plt
import seaborn as sns

# Set plot style
sns.set(style="whitegrid")
</pre>

<pre>
# Plot the distribution of ratings
plt.figure(figsize=(8, 6))
sns.countplot(x='rating_x', data=merged_df, palette='viridis')
plt.title('Distribution of Ratings', fontsize=16)
plt.xlabel('Ratings', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()
</pre>

<pre>
<ipython-input-11-32ceca067aaa>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x='rating_x', data=merged_df, palette='viridis')
</pre>

</td>
</tr>
<tr>
<td>Feature Engineering</td>
<td>Attached the code in the final submission.</td>
</tr>
<tr>
<td>Save Processed Data</td>
<td>—</td>
</tr>
</table>

4. Model Development Phase

4.1. Model Selection Report

For analyzing Amazon cell phone reviews, model selection involves choosing algorithms that accurately capture sentiment, aspects, and topics. Popular choices include Naive Bayes, Support Vector Machines and Random Forest. Considering factors like dataset size, complexity, and interpretability ensures optimal model selection, leveraging tools like NLTK, scikit-learn, and TensorFlow to achieve accurate and informative insights. In Model Selection multiple machine learning models are evaluated to identify the best approach for classifying amazon cell phone reviews. The models are compared based on performance, accuracy, complexity, and computational requirements. The aim is to select optimized model for Amazon cell phone reviews.

Model	Description
Random Forest	Random Forest is an ensemble learning algorithm that builds multiple decision trees and merges them to get more accurate and stable predictions. It is well-suited for handling overfitting and works well with complex data structures.
Logistic Regression	Logistic Regression is a linear classification model used to predict the probability of a binary outcome (positive or negative sentiment) by applying a logistic function to a linear combination of the input features. It works well for simple text classification tasks.
Support Vector Machine (SVM)	SVM is a robust classification algorithm that works by finding the optimal hyperplane that best separates the data points into different classes (positive/negative sentiment). It is especially effective in Working with high-dimensional text data.

Conclusion

After analyzing all these models, we should use **Random Forest** over other models because of its robustness and ability to handle complex datasets without overfitting. Unlike linear models like Logistic Regression and SVM, Random Forest can model nonlinear relationships and interactions between features effectively. It is less sensitive to noisy data and automatically handles missing values, making it more adaptable for diverse text data.

4.2. Initial Model Training Code, Model Validation andEvaluation Report

The initial model training code will be included in the final submission through screenshots. The model validation and evaluation report will summarize the performance of the Analysis of amazon cell phone reviews using metrics such as accuracy, precision, recall, F1-score, and confusion matrices. Models like SVM, Decision Tree, Naive Bayes, Gradient Boosting and Logistic Regression will be evaluated to select the best-performing machine learning models that can be used for sentiment analysis of Amazon cell phone reviews.

Initial Model Training Code:

```
[ ] # Import required libraries for Random Forest
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, ConfusionMatrixDisplay
    from sklearn.model_selection import train_test_split
    from sklearn.feature_extraction.text import TfidfVectorizer

    # Step 1: Convert text data into numerical features using TF-IDF
    # You can adjust ngram_range and max_features as per your requirement
    vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=3000)
    X = vectorizer.fit_transform(merged_df['cleaned_review'])

    # Convert ratings to sentiment categories for Random Forest model
    def convert_rating_to_sentiment(rating):
        if rating >= 4:
            return 'positive'
        elif rating == 3:
            return 'neutral'
        else:
            return 'negative'

    y = merged_df['rating_x'].apply(convert_rating_to_sentiment)
```

SPLITTING THE DATA INTO TRAINING AND TESTING SETS

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

MODEL BUILDING (RANDOM FOREST)

```
[ ]  rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

     # Step 4: Train the Random Forest Classifier on the training data
     rf_classifier.fit(X_train, y_train)

     # Step 5: Predict sentiment on the test set
     y_pred = rf_classifier.predict(X_test)

     # Step 6: Evaluate the model
     accuracy = accuracy_score(y_test, y_pred)
     classification_report_output = classification_report(y_test, y_pred)

     print("Random Forest Accuracy: ", (accuracy * 100))
     print(f"Classification Report:\n{classification_report_output}")
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest	Classification Report: precision recall f1-score support negative 0.80 0.77 0.79 3350 neutral 0.91 0.07 0.14 926 positive 0.87 0.97 0.92 9322 accuracy 0.86 13598 macro avg 0.86 0.60 0.61 13598 weighted avg 0.86 0.86 0.83 13598	85.68%	Random Forest Confusion Matrix [[2580 2 768] [327 69 530] [315 5 9002]]

5.Model Optimization and Tuning Phase

5.1. Tuning Phase

In the tuning phase of the analysis, machine learning models and natural language processing (NLP) algorithms will be optimized to accurately extract insights from the Amazon cell phone reviews. This includes fine-tuning sentiment analysis models, adjusting hyperparameters, and refining topic modelling techniques to ensure high precision in detecting customer sentiments and key topics. The tuning process will also involve evaluating model performance using metrics like accuracy, precision, and recall, and iterating on feature selection to improve the overall quality of insights derived from the reviews**.**

Hyperparameter Tuning Documentation:

<table>
<tr><th>Model</th><th>Tuned Hyperparameters</th><th>Optimal Values</th></tr>
<tr><td>Random Forest</td><td><pre>
# Step 1: Convert text data into numerical features using TF-IDF
# You can adjust ngram_range and max_features as per your requiremen
vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=3000)
X = vectorizer.fit_transform(merged_df['cleaned_review'])

# Convert ratings to sentiment categories for Random Forest model
def convert_rating_to_sentiment(rating):
    if rating >= 4:
        return 'positive'
    elif rating == 3:
        return 'neutral'
    else:
        return 'negative'

y = merged_df['rating_x'].apply(convert_rating_to_sentiment)
</pre></td><td><pre>
Classification Report:
              precision   recall f1-score   support

    negative   0.80   0.77   0.79   3350
     neutral   0.91   0.07   0.14   926
    positive   0.87   0.97   0.92   9322

    accuracy   0.86   13598
   macro avg   0.86   0.60   0.61   13598
weighted avg   0.86   0.86   0.83   13598
</pre></td></tr>
<tr><td></td><td><pre>
[ ] rf_classifier = RandomForestClassifier(n_estimators=100, random_state=

    # Step 4: Train the Random Forest Classifier on the training data
    rf_classifier.fit(X_train, y_train)

    # Step 5: Predict sentiment on the test set
    y_pred = rf_classifier.predict(X_test)

    # Step 6: Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    classification_report_output = classification_report(y_test, y_pred)

    print("Random Forest Accuracy: ", (accuracy * 100))
    print(f"Classification Report:\n{classification_report_output}")
</pre></td><td></td></tr>
</table>

Performance Metrics Comparison Report:

Model	Optimized Metric
Model-1 Random Forest	Random Forest Confusion Matrix [[2580 2 768] [327 69 530] [315 5 9002]] Classification Report: precision recall f1-score support negative 0.80 0.77 0.79 3350 neutral 0.91 0.07 0.14 926 positive 0.87 0.97 0.92 9322 accuracy 0.86 13598 macro avg 0.86 0.60 0.61 13598 weighted avg 0.86 0.86 0.83 13598

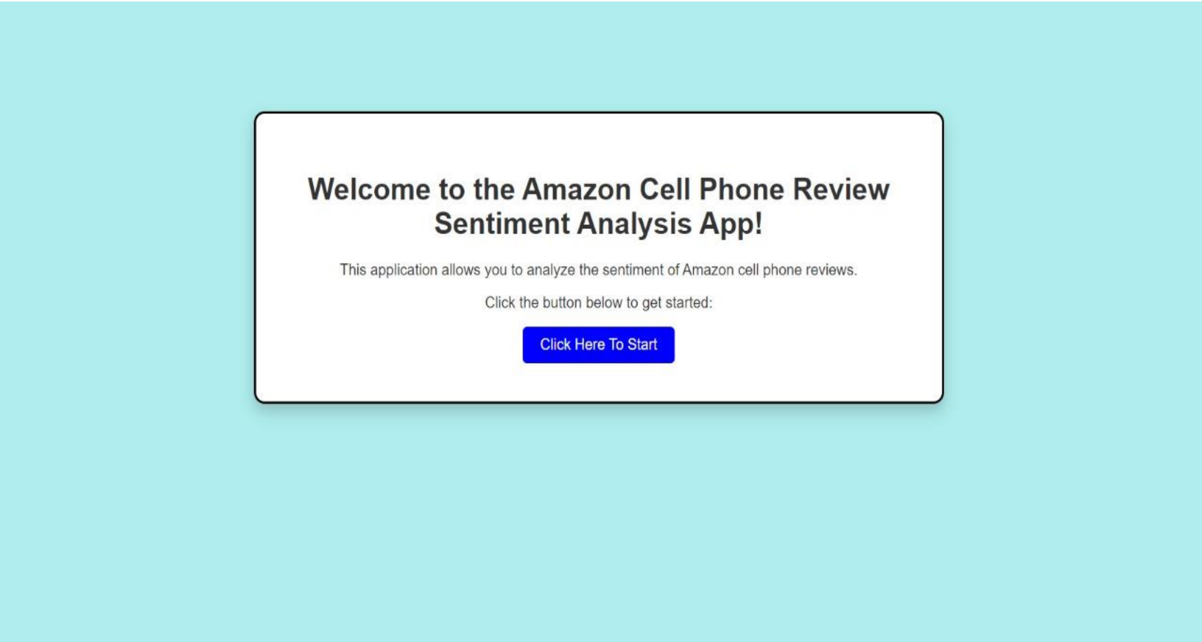
NOTE: Although various models were considered, the primary focus was on Random Forest, as it is well-suited for text classification tasks due to its simplicity and effectiveness and ability to handle complex datasets without overfitting.

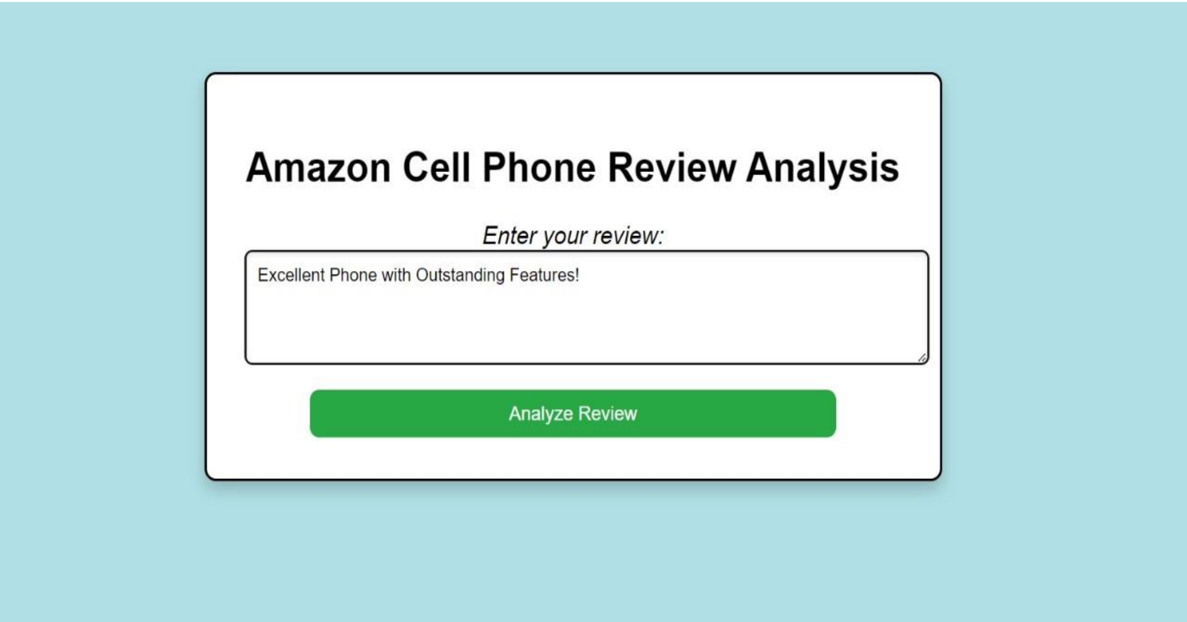
5.2. Final Model Selection Justification:

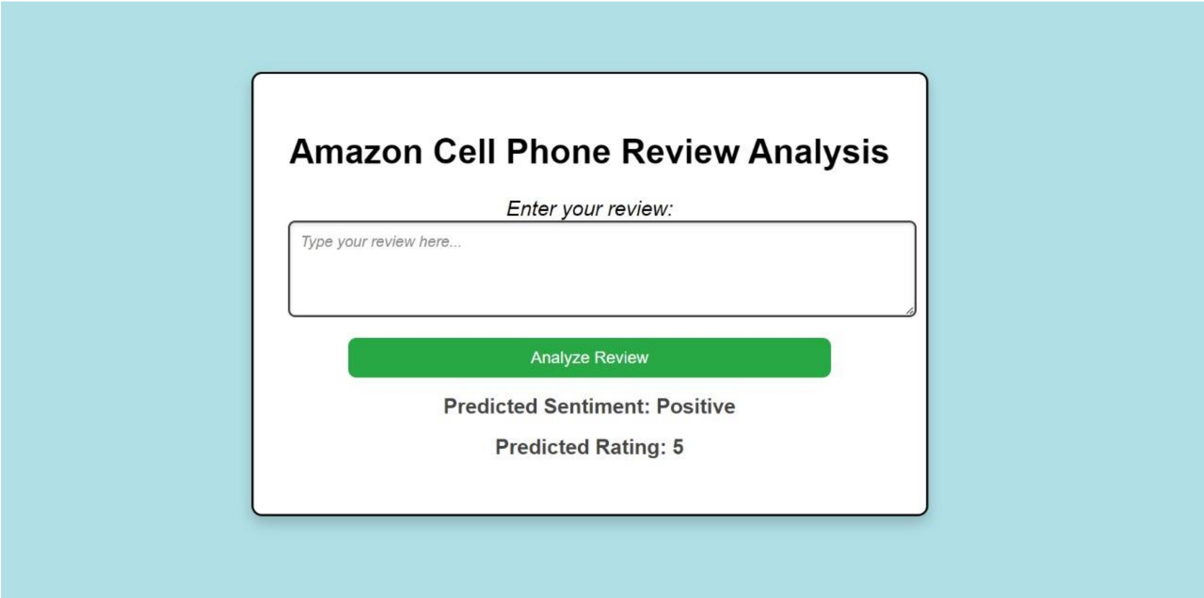
Final Model	Reasoning
Random Forest	Random Forest was chosen over other models because of its ability to handle complex, nonlinear relationships in the text data, which can improve sentiment prediction accuracy. It aggregates multiple decision trees, reducing the risk of overfitting and providing more robust, generalizable results compared to models like Logistic Regression or SVM.

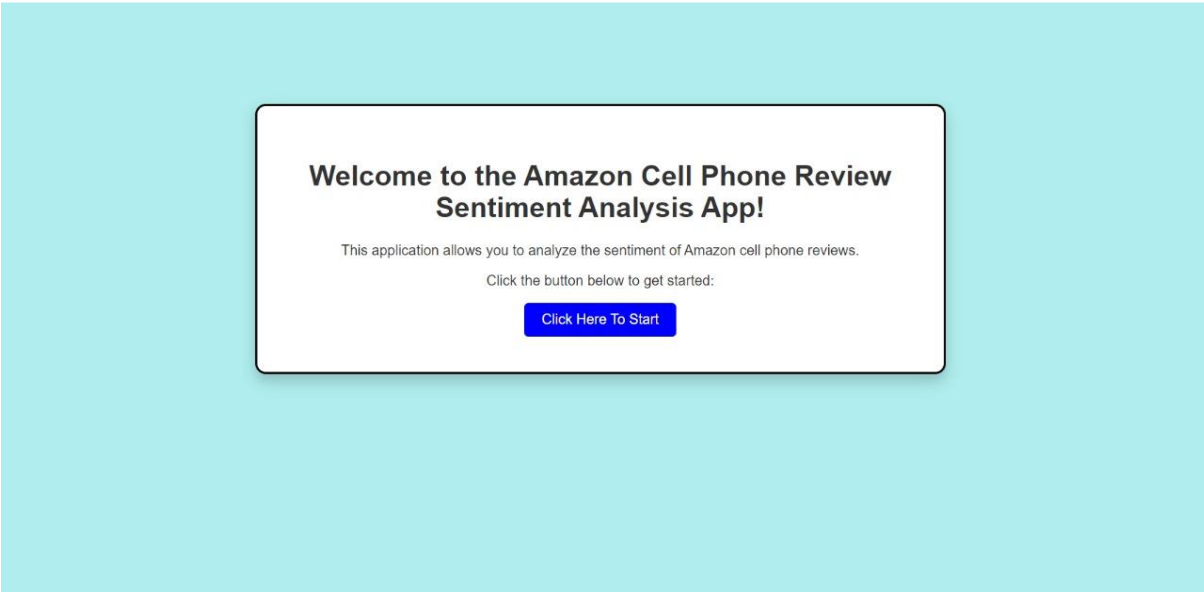
6.Results

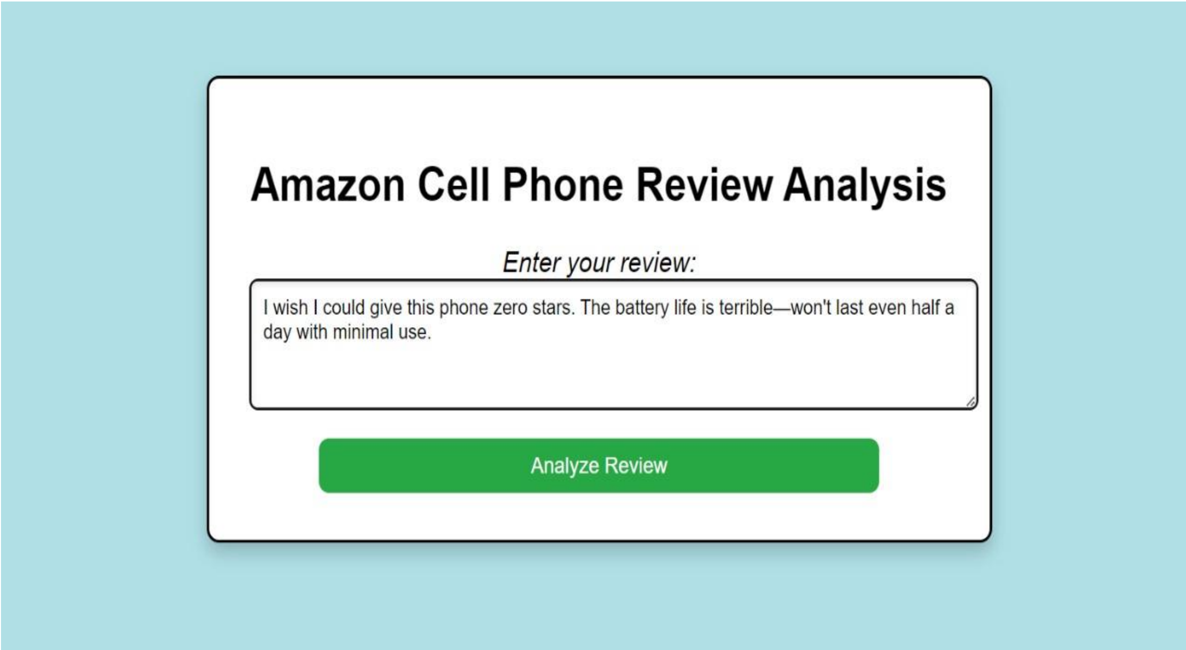
6.1. Output Screen shots

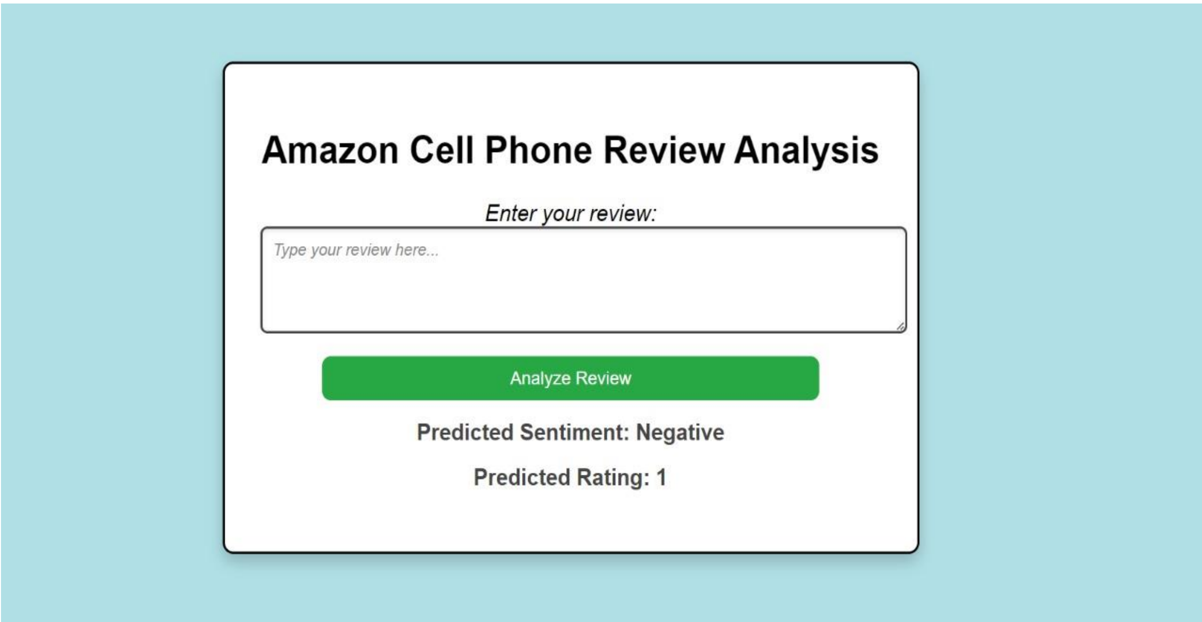





Amazon Cell Phone Review Analysis
Enter your review:
Type your review here...
Analyze Review
Predicted Sentiment: Positive
Predicted Rating: 5


Welcome to the Amazon Cell Phone Review Sentiment Analysis App!
This application allows you to analyze the sentiment of Amazon cell phone reviews.
Click the button below to get started:
Click Here To Start


Amazon Cell Phone Review Analysis
Enter your review:
I wish I could give this phone zero stars. The battery life is terrible—won't last even half a day with minimal use.
Analyze Review


Amazon Cell Phone Review Analysis
Enter your review:
Type your review here...
Analyze Review
Predicted Sentiment: Negative
Predicted Rating: 1

7.Advantages & Disadvantages

Advantages:

Analyzing Amazon cell phone reviews offers several significant benefits:

- **Informed Decision-Making:** By analyzing customer reviews, potential buyers gain valuable insights into the strengths and weaknesses of various cell phone models, enabling them to make more informed purchasing decisions.
- **Understanding Consumer Sentiment:** Sentiment analysis helps identify overall customer satisfaction levels, revealing trends in how users feel about specific features like battery life, camera quality, and user experience
- **Product Improvement Feedback:** Manufacturers can utilize insights from reviews to understand common pain points and areas for enhancement, leading to better product development and increased customer satisfaction.
- **Market Trends Identification:** The analysis can uncover emerging trends in consumer preferences, such as the rising demand for specific features or design elements, allowing brands to stay ahead of the competition.
- **Competitive Analysis:** By comparing reviews across different brands and models, companies can identify their competitive advantages and weaknesses, guiding marketing strategies and product positioning.
- **Fraud Detection:** Analyzing review patterns can help detect fraudulent or biased reviews, ensuring that consumers have access to genuine feedback and enhancing the credibility of the review system.
- **Tailored Marketing Strategies:** Insights gained from reviews can inform targeted marketing campaigns, helping brands to address specific customer needs and improve engagement with their audience.
- **Data-Driven Recommendations:** The findings from the analysis can be leveraged to provide personalized recommendations for consumers, enhancing their shopping experience and increasing conversion rates.

Overall, the analysis of Amazon cell phone reviews not only benefits consumers in making better choices but also provides manufacturers with actionable insights to drive product innovation and enhance customer satisfaction.

Disadvantages:

While analyzing Amazon cell phone reviews has several advantages, it also presents certain disadvantages and challenges:

- **Bias and Subjectivity**: Customer reviews can be influenced by individual biases, emotions, or specific experiences that may not reflect the overall quality of a product. This subjectivity can skew analysis results and lead to misleading conclusions.
- **Volume of Data**: The sheer volume of reviews can make it difficult to extract meaningful insights efficiently. Processing large datasets requires significant computational resources and time, which may pose challenges for smaller organizations or individual researchers.
- **Data Quality Issues**: Reviews may contain noise, such as spam, irrelevant comments, or overly technical jargon, complicating the data cleaning process. Poor-quality data can lead to inaccurate analysis and unreliable results.
- **Temporal Changes**: Customer sentiments can change over time, especially following updates or changes in product features. Analyzing static reviews may not accurately reflect the current state of customer satisfaction, leading to outdated insights.
- **Limited Context**: Reviews often lack context regarding user needs or specific use cases. A review that is negative for one user may not be relevant for another with different requirements. This variability can make it challenging to generalize findings across diverse consumer segments.
- **Ethical Concerns**: The use of web scraping or data collection methods raises ethical issues regarding user privacy and compliance with platform policies. Failing to adhere to ethical standards can result in legal ramifications or damage to reputation.
- **Overemphasis on Ratings**: Focusing too heavily on star ratings can overlook the nuances in written reviews. A high rating might accompany specific complaints that are crucial for understanding customer sentiment, while low ratings may miss positive aspects mentioned in the text.

Overall, while the analysis of Amazon cell phone reviews can yield valuable insights, it is essential to approach it with caution and be aware of these potential disadvantages to ensure accurate and ethical outcomes.

6. Conclusion

The analysis of Amazon cell phone reviews serves as a vital resource for both consumers and manufacturers, bridging the gap between product offerings and consumer expectations. By leveraging advanced techniques in natural language processing and sentiment analysis, we can extract meaningful insights from the extensive and diverse pool of user-generated feedback. This analysis illuminates key aspects of consumer sentiment, enabling potential buyers to make informed decisions based on real-world experiences rather than solely relying on marketing claims.

One of the primary advantages of this analysis is its ability to highlight common trends and recurring themes within reviews, such as performance, battery life, camera quality, and user interface. These insights empower consumers to understand which features resonate most with users, guiding them toward products that best meet their specific needs. Furthermore, the ability to compare feedback across different brands and models allows consumers to identify the best options available in the market, fostering a sense of confidence in their purchasing decisions.

For manufacturers, the analysis of customer reviews provides invaluable feedback that can drive product improvement and innovation. Understanding the strengths and weaknesses highlighted by users can inform design choices, feature enhancements, and customer service strategies. This data-driven approach not only enhances product quality but also aligns offerings more closely with consumer desires, ultimately leading to increased customer satisfaction and loyalty.

However, it is crucial to recognize the limitations associated with analyzing Amazon cell phone reviews. Biases in user feedback, the prevalence of fake or misleading reviews, and the challenges of managing vast datasets can compromise the quality of insights derived from this analysis. Ethical considerations, such as user privacy and compliance with platform policies, must also be addressed to ensure responsible data usage.

In conclusion, the analysis of Amazon cell phone reviews represents a powerful mechanism for harnessing consumer feedback to improve product offerings and enhance the overall shopping experience. By providing actionable insights for both buyers and manufacturers, this analysis contributes to a more transparent and informed marketplace, ultimately benefiting all stakeholders involved. As technology continues to evolve and consumer expectations shift, ongoing analysis of product reviews will remain a critical component in navigating the complexities of the digital retail landscape.

9.Future Scope

The feature scope for analyzing Amazon cell phone reviews encompasses a comprehensive set of elements aimed at extracting meaningful insights from user-generated feedback. This analysis seeks to leverage various aspects of the reviews, including textual content, metadata, and user interactions, to provide a holistic understanding of consumer sentiments and preferences. The following features outline the key components of the analysis:

1. **Sentiment Analysis**: Implementing natural language processing (NLP) techniques to determine the overall sentiment of each review—whether it is positive, negative, or neutral. This feature will help quantify consumer feelings toward specific products.
2. **Topic Modeling**: Identifying prevalent themes and topics within the reviews, such as camera quality, battery life, durability, and software performance. This feature allows for a deeper understanding of what aspects of cell phones are most frequently discussed.
3. **Feature Extraction**: Highlighting specific attributes that consumers mention in their reviews. This could include aspects like screen size, design, price, and additional functionalities (e.g., 5G capability, waterproofing), which can be essential for comparative analysis.
4. **Rating Correlation**: Analyzing the relationship between star ratings and sentiment scores, enabling the identification of discrepancies between ratings and review content. This feature helps in understanding whether the star ratings accurately reflect consumer sentiment.
5. **Temporal Analysis**: Examining how sentiments and opinions evolve over time, particularly in response to software updates, new product releases, or changes in market trends. This feature provides insights into how consumer perceptions shift over the lifecycle of a product.
6. **User Demographics and Behavior**: Where possible, analyzing demographic data of reviewers (e.g., age, location) to understand how different consumer segments perceive products differently. This can help tailor marketing strategies to specific audiences.
7. **Comparative Analysis**: Conducting cross-brand comparisons to evaluate consumer preferences and satisfaction levels across different manufacturers. This feature can assist consumers in making informed choices based on comparative feedback.

8. **Spam and Fake Review Detection**: Implementing algorithms to identify and filter out fake or misleading reviews, ensuring that the analysis is based on genuine user experiences. This is crucial for maintaining the integrity of the dataset.
9. **Visualizations and Reporting**: Creating interactive dashboards and visualizations to present findings in an accessible format. This feature allows stakeholders to easily interpret data trends and insights, facilitating informed decision-making.
10. **Recommendation System**: Utilizing insights from reviews to develop a recommendation system that suggests cell phones based on user preferences and sentiment analysis. This feature can enhance the shopping experience by personalizing product suggestions.

In conclusion, the feature scope for the analysis of Amazon cell phone reviews encompasses a diverse range of elements aimed at capturing the nuances of consumer feedback. By focusing on sentiment analysis, topic modeling, and comparative evaluations, this analysis can provide valuable insights for both consumers and manufacturers. Ultimately, these features will contribute to a better understanding of market dynamics, consumer preferences, and product performance, facilitating informed decision-making in the competitive landscape of cell phones.

10. Appendix

10.1. Source Code

```
# IMPORTING THE LIBERARIES

# Import libraries

import pandas as pd

import nltk

from sklearn.model_selection import train_test_split

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

from sklearn.metrics import accuracy_score, classification_report

# LOADING THE DATASET

# Load the datasets (replace the file paths with your actual file paths)
```

```
items_df = pd.read_csv('/content/drive/MyDrive/20191226-items.csv')
reviews_df = pd.read_csv('/content/drive/MyDrive/20191226-reviews.csv')

# Merge datasets based on 'asin' to combine product info with reviews
merged_df = pd.merge(reviews_df, items_df, on='asin')

# View the first few rows to check the loaded data
print(merged_df.head())
print(merged_df.columns)
print(merged_df.shape)
print(merged_df.info())
print(merged_df.describe())
print(merged_df.isnull().sum())
# TEXT PREPROCESSING
# Define a function to preprocess the review text
def preprocess_text(text):
    # Check if the text is a string
    if isinstance(text, str):
        # Convert to lowercase
        text = text.lower()
        # Remove punctuation using NLTK's tokenizer
        text = nltk.RegexpTokenizer(r'\w+').tokenize(text)
        # Re-join words into a single string
        return ' '.join(text)
    else:
        # Handle non-string values (e.g., return an empty string or a placeholder)
```

```
    return ''

# Apply preprocessing to the review body column
merged_df['cleaned_review'] = merged_df['body'].apply(preprocess_text)

# View the preprocessed reviews
print(merged_df[['body', 'cleaned_review']].head())
```

VISULIZATIONS

```
import matplotlib.pyplot as plt
import seaborn as sns

# Set plot style
sns.set(style="whitegrid")

# Plot the distribution of ratings
plt.figure(figsize=(8, 6))
sns.countplot(x='rating_x', data=merged_df, palette='viridis')
plt.title('Distribution of Ratings', fontsize=16)
plt.xlabel('Ratings', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.show()
# Calculate average rating by brand
avg_rating_by_brand =
merged_df.groupby('brand')['rating_x'].mean().sort_values()

# Plot average rating by brand as a pie chart
plt.figure(figsize=(8, 8))
```

```
plt.pie(avg_rating_by_brand.values, labels=avg_rating_by_brand.index,
autopct='%1.1f%%', startangle=90, colors=sns.color_palette('magma',
len(avg_rating_by_brand)))

plt.title('Average Rating by Brand', fontsize=16)

plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a circle.

plt.show()

# Distribution of brands in the dataset

plt.figure(figsize=(10,6))

merged_df['brand'].value_counts().nlargest(10).plot(kind='bar', color='c')

plt.title('Top 10 Most Reviewed Brands')

plt.xlabel('Brand')

plt.ylabel('Number of Reviews')

plt.show()

from wordcloud import WordCloud

import matplotlib.pyplot as plt


# Assuming 'review_text' is the column with the review content

# Replace 'review_text' with the actual column name if different

# Join review texts into a single string

all_reviews = ' '.join(str(v) for v in merged_df['brand']) # Cast each value to a
string before joining


wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(all_reviews)


# Display the word cloud

plt.figure(figsize=(10, 8))
```

```
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Most Common Words in Reviews')
plt.show()
```

SENTIMENT ANALYSIS

```
# Features (input) and labels (output)
X = merged_df['cleaned_review']
y = merged_df['rating_x']  # Adjust as needed for sentiment or classification

# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Check the shape of the data
print(f"Training data size: {X_train.shape}, Test data size: {X_test.shape}")
# Initialize the VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Define a function to get the sentiment (positive, neutral, negative)
def get_sentiment(text):
    score = analyzer.polarity_scores(text)
    if score['compound'] >= 0.05:
        return 'positive'
    elif score['compound'] <= -0.05:
        return 'negative'
    else:
        return 'neutral'
```

```
# Apply sentiment analysis to the test set
X_test_sentiment = X_test.apply(get_sentiment)

# Convert ratings to sentiment categories for evaluation
def convert_rating_to_sentiment(rating):
    if rating >= 4:
        return 'positive'
    elif rating == 3:
        return 'neutral'
    else:
        return 'negative'

y_test_sentiment = y_test.apply(convert_rating_to_sentiment)

# View a few sentiment labels
print("Predicted Sentiments:\n", X_test_sentiment.head())
print("Actual Sentiments:\n", y_test_sentiment.head())
# Evaluate the accuracy of the sentiment analysis
accuracy = accuracy_score(y_test_sentiment, X_test_sentiment)
classification_report_output = classification_report(y_test_sentiment,
X_test_sentiment)

print("Accuracy: ",(accuracy)*100)
print(f"Classification Report:\n{classification_report_output}")
# Group reviews by brand and calculate sentiment distribution
```

```
competitor_sentiment = merged_df.groupby('brand').apply(lambda x:
x['cleaned_review'].apply(get_sentiment).value_counts(normalize=True))


# Display competitor sentiment analysis
print("Competitor Sentiment Analysis:\n", (competitor_sentiment)*100)
# List of features to analyze (you can modify this list)
features = ['battery', 'camera', 'screen', 'performance', 'price']


# Performing sentiment analysis for each feature
for feature in features:
    feature_reviews =
merged_df[merged_df['cleaned_review'].str.contains(feature)]
    feature_sentiment =
feature_reviews['cleaned_review'].apply(get_sentiment).value_counts(normal
ize=True)
    print(f"Sentiment Analysis for {feature}:\n", (feature_sentiment)*100)
```

MODEL BUILDING (RANDOM FOREST)

```
from sklearn.feature_extraction.text import TfidfVectorizer


vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=3000)
# Import required libraries for Random Forest
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix, ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer


# Step 1: Convert text data into numerical features using TF-IDF
```

```
# You can adjust ngram_range and max_features as per your requirement
vectorizer = TfidfVectorizer(ngram_range=(1, 2), max_features=3000)
X = vectorizer.fit_transform(merged_df['cleaned_review'])

# Convert ratings to sentiment categories for Random Forest model
def convert_rating_to_sentiment(rating):
    if rating >= 4:
        return 'positive'
    elif rating == 3:
        return 'neutral'
    else:
        return 'negative'

y = merged_df['rating_x'].apply(convert_rating_to_sentiment)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# Step 4: Train the Random Forest Classifier on the training data
rf_classifier.fit(X_train, y_train)

# Step 5: Predict sentiment on the test set
y_pred = rf_classifier.predict(X_test)

# Step 6: Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
classification_report_output = classification_report(y_test, y_pred)
```

```
print("Random Forest Accuracy: ", (accuracy * 100))

print(f"Classification Report:\n{classification_report_output}")

import matplotlib.pyplot as plt

cm = confusion_matrix(y_test, y_pred) # Replace y_pred_best with y_pred

# Visualize the confusion matrix

plt.figure(figsize=(10, 7))

disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=['Negative', 'Neutral', 'Positive'])

disp.plot(cmap=plt.cm.Blues)

plt.title('Confusion Matrix')

plt.show()

print("Random Forest Confusion Matrix \n")

print(cm)

import pickle


# Save the model to a file

with open('random_forest_model.pkl', 'wb') as model_file:

    # Changed best_rf to rf_classifier

    pickle.dump(rf_classifier, model_file)
```

SAVING THE MODEL

```
# Save the vectorizer to a file

with open('tfidf_vectorizer.pkl', 'wb') as vectorizer_file:

    pickle.dump(vectorizer, vectorizer_file)


print("Model and vectorizer saved successfully!")
```

#Download the pickle files

For deploying the flask application, we need to create the directory withthe project name (optional)

```
Analysis Of Amazon Cell Phone Reviews        (main directory)
├── app.py              (python file consists of flask application)
└── templates/                 (directory that contains html files)
    ├── welcome.html
    ├── index.html
```

Building the flask Application(app.py)

```
from flask import Flask, request, render_template, redirect, url_for
import pickle
import random

# Initialize Flask application
app = Flask(__name__)

# Load the model and vectorizer
with open('C:\\Users\\DELL\\OneDrive\\Desktop\\final year project\\random_forest_model.pkl', 'rb') as f:
    model = pickle.load(f)

with open('C:\\Users\\DELL\\OneDrive\\Desktop\\final year project\\tfidf_vectorizer (2).pkl', 'rb') as f:
    vectorizer = pickle.load(f)

# Function to map sentiment to a numeric rating
def sentiment_to_rating(sentiment):
    if sentiment == 'negative':
        return random.randint(1, 2)  # Randomly pick 1 or 2
    elif sentiment == 'neutral':
        return 3  # Fixed rating for neutral
    else:
        return random.randint(4, 5)  # Randomly pick 4 or 5 for positive

# Route for the welcome page
@app.route('/welcome')
def welcome():
```

```
    return render_template('welcome.html')

# Redirect root URL to the welcome page
@app.route('/')
def index():
    return redirect(url_for('welcome'))

# Route for the analysis homepage
@app.route('/analyze', methods=['GET', 'POST'])
def home():
    sentiment = None
    predicted_rating = None

    if request.method == 'POST':
        # Get the input review
        review = request.form['review']

        # Preprocess the input review
        review_vector = vectorizer.transform([review])

        # Make a prediction
        predicted_sentiment = model.predict(review_vector)[0]  # This returns
'positive', 'neutral', or 'negative'

        # Convert sentiment to numeric rating
        predicted_rating = sentiment_to_rating(predicted_sentiment)

        # Capitalize the sentiment label
        sentiment = predicted_sentiment.capitalize()

    return render_template('index.html', sentiment=sentiment,
predicted_rating=predicted_rating)

if __name__ == "__main__":
    app.run(debug=True)
```

Templates

welcome.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome to Amazon Review Analysis</title>
</head>
<body>
  <div class="container">
    <h1>Welcome to the Amazon Cell Phone Review Sentiment Analysis
App!</h1>
    <p>This application allows you to analyze the sentiment of Amazon cell
phone reviews.</p>
    <p>Click the button below to get started:</p>
    <a href="/analyze"><button>Get Start</button></a>
  </div>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color:paleturquoise;
      padding: 20px;
    }

    .container {
      background-color: white;
      max-width: 700px;
      margin: 100px auto;
      padding: 40px;
      border-radius: 12px;
      border: 3px solid black;
      box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
      text-align: center;
    }

    h1{
      color: #333;
      background-color:white;
    }
    p{
      color:#333 ;
      background-color:white;
    }

    button {
      padding: 10px 20px;
```

```
            background-color:blue;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
        }

        button:hover {
            background-color: #0056b3;
        }
        *{
            background-color:paleturquoise;
        }
    </style>
</body>
</html>
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Amazon Cell Phone Review Analysis</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='index12.css')
}}">
</head>
<body>
    <div class="container">
        <h1>Amazon Cell Phone Review Analysis</h1>

        <!-- Form to submit review -->
        <form method="POST">
           <i><label for="review">Enter your review:</label></i><br>
            <textarea id="review" name="review" rows="4" cols="50"
placeholder="Type your review here..." required></textarea><br>
            <input type="submit" value="Analyze Review">
        </form>

        <!-- Display the sentiment and predicted rating -->
```

```
    {% if sentiment %}
      <h2>Predicted Sentiment: {{ sentiment }}</h2>
      <h3>Predicted Rating: {{ predicted_rating }}</h3>
    {% endif %}
  </div>

  <style>
    /* Set gradient background */
    body {
      background-color: powderblue ; /* gradient from blue to light blue */
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      text-align: center;
      animation: fadeInDown 1s ease;
    }

    /* Container for form and content */
    .container {
      background-color: white;
      max-width: 700px;
      margin: 100px auto;
      padding: 40px;
      border-radius: 12px;
      border: 3px solid black;
      box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
      text-align: center;

    }

    /* Main title styling */
    h1 {
      font-size: 2.5em;
      animation: fadeInDown 1s ease;
      color:solid black;
      margin-bottom: 30px;
      font-family: 'Helvetica', sans-serif;
    }

    /* Label styling */
    label {
      font-size: 1.5em;
      color:black;
```

```
}

/* Text area styling */
textarea {
   width: 100%;
   padding: 12px;
   font-size: 1.1em;
   border-radius: 8px;
   box-shadow: inset 0 2px 5px rgba(0, 0, 0, 0.1);
   margin-bottom: 20px;
   font-family: 'Arial', sans-serif;
   color:solid black;
   border: 3px solid #444;
}

/* Submit button styling */
input[type="submit"] {
   background-color: #28a745;
   color: white;
   padding: 12px 20px;
   font-size: 1.2em;
   border: #28a745;
   border-radius: 10px;
   cursor: pointer;
   transition: background-color 0.3s ease;
   width: 80%;
}

input[type="submit"]:hover {
   background-color: #218838;
}

/* Prediction result styling */
h2, h3 {
   margin-top: 20px;
   color: #444;
   font-size: 1.5em ;
}

/* Placeholder styling */
textarea::placeholder {
   color: #888;
   font-style: italic;
```

```
        }

        /* Mobile responsive adjustments */
        @media (max-width: 600px) {
            .container {
                margin: 20px;
                padding: 20px;
            }

            h1 {
                font-size: 2em;
            }

            textarea {
                font-size: 1em;
            }

            input[type="submit"] {
                font-size: 1em;
                padding: 10px;
            }
        }


    </style>
</body>
</html>
```

11.GitHub & Project Demo Link

https://github.com/vaibhavi1616/Analysis-Of-Amazon-Cell-Phone-Reviews

https://drive.google.com/file/d/137vcC_I4iScAr9pPiQYF4eoDLLlt9D6o/view?usp=sharing