SENTIMENT ANALYSIS ON FLIPKART REVIEWS USING PYTHON

NAME:MAHEK BEGUM

BRANCH:IT

ABSTRACT

- In today's digital age, online reviews strongly influence consumer purchasing decisions, making sentiment analysis vital for businesses. This project aims to classify Flipkart product reviews into positive, negative, or neutral sentiments using Python, Natural Language Processing (NLP), and machine learning algorithms. The workflow involves data collection from Flipkart, data preprocessing (removing noise, tokenization, lemmatization, and stop word removal), and feature extraction through techniques like word frequency, n-grams, and embeddings. Machine learning models such as Support Vector Machine, Random Forest, and Gradient Boosting are developed and evaluated using accuracy, precision, recall, and F1-score. The trained model is then deployed to perform real-time sentiment classification, helping businesses gain insights into customer satisfaction and improve their services.

INTRODUCTION

- Machine Learning (ML), a subset of Artificial Intelligence (AI), enables systems to learn from data and make predictions without explicit programming. It identifies patterns from datasets and generalizes them to unseen data.
- **2. Types of Machine Learning**
- **Supervised Learning:** Trains on labeled data to predict outcomes.
- **Unsupervised Learning:** Finds hidden patterns or groups in unlabeled data.
- **Reinforcement Learning:** Learns by interacting with the environment using rewards/punishments.

ML in Sentiment Analysis

- Flipkart reviews contain valuable insights on products and services. Manual analysis is impractical, so sentiment analysis helps classify reviews as **positive, negative, or neutral** using ML.
- Tools & Libraries
- **Python Libraries:** Pandas, NLTK, Scikit-learn, Matplotlib, Seaborn, WordCloud.
- **Methodology:**
- **Data Collection:** Import Flipkart reviews dataset.
- **Data Preprocessing:** Cleaning, tokenization, lemmatization, stop-word removal.
- **Feature Extraction:** Convert text into vectors using **TF-IDF**.
- **Model Training:** Random Forest, Support Vector Machine, Gradient Boosting.
- **Evaluation:** Accuracy, Precision, Recall, F1-score.

Sentiment Analysis Matters on Flipkart:

- **Enhancing Customer Experience**: By analyzing sentiments, Flipkart can quickly identify areas where customer satisfaction is lacking, allowing the platform to make necessary improvements to enhance the overall shopping experience. Product Development and Improvement: Sentiment analysis helps manufacturers and sellers understand how their products are perceived by customers, enabling them to address issues, improve product features, and innovate based on **customer feedback**. Marketing and Sales Strategy: Understanding the sentiment behind reviews can help Flipkart tailor marketing campaigns, highlight positive reviews, and address negative feedback more effectively, thereby driving sales. Brand Reputation Management: Continuous monitoring of customer sentiment allows Flipkart to proactively manage its brand reputation by responding to negative feedback promptly and enhancing areas that are positively received

Fragrance-1
(Lavender)

REVIEWS

1. Smells amazing! A perfect purchase :)
2. Must buy! Super amazing.
3. Quite satisfactory

Fragrance-1
(Rose)

REVIEWS

1. A decent purchase
2. Quite okayish! Smells average
3. Could have been better in lot terms

Fragrance-1
(Lemon)

REVIEWS

1. An absolute waste of money.
2. Total waste of money
3. Terrible smell, not worth buytng

SENTIMENT ANALYZER

POSITIVE (81%)

NEUTRAL (88%)

negative (91%))

SOFTWARE REQUIREMENTS

- **Python**
- Use latest version (download from **python.org** or via **Anaconda/conda**).
- **2. Python Libraries**
- **pandas** → Data manipulation (pip install pandas)
- **numpy** → Numerical operations (pip install numpy)
- **re** → Regular expressions (built-in)
- **matplotlib** → Data visualization (pip install matplotlib)
- **scikit-learn** → ML tasks (pip install scikit-learn)
  - train_test_split, TfidfVectorizer
  - Classifiers: Random Forest, Gradient Boosting, SVC
  - Evaluation: accuracy, classification report, confusion matrix
- **collections** → Frequency counts (built-in)

3. Software Setup

- IDE: **Jupyter Notebook**, PyCharm, or VSCode
- Jupyter installation: pip install notebook
- **4. Data**
- Flipkart reviews dataset (text reviews + sentiment labels).
- **5. System Requirements**
- RAM: **8GB+ recommended**
- Processor: **Multi-core CPU** for faster training.
- **6. Steps for Sentiment Analysis**
- **Data Preprocessing:** Load dataset → clean text → tokenize/vectorize (TF-IDF).
- **Model Training:** Split data → train ML models.
- **Evaluation:** Accuracy, precision, recall, F1-score, confusion matrix.
- **Visualization:** o Use matplotlib to create plots for visualizing model performance or review distributions.

Methodology

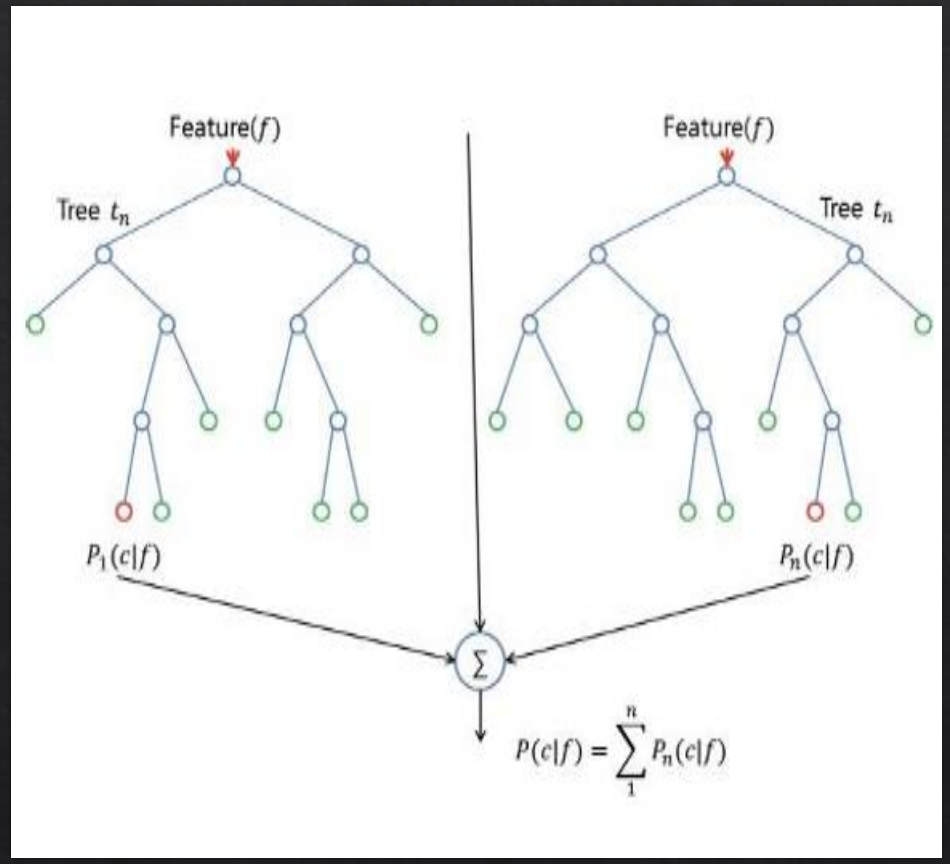
- **1. Data Collection**
- Obtain Flipkart reviews (text + sentiment labels).
- Source: Flipkart review pages, APIs, or public datasets.
- Store in CSV with review_text and sentiment columns.
- **2. Data Preprocessing**
- Load dataset using **pandas**.
- Remove missing/inconsistent values → improves accuracy.
- Remove duplicates → ensures consistent data.
- **3. Text Cleaning**
- Convert text to **lowercase**.
- **Tokenization** → split text into words.
- Remove **stop words** (“and”, “the”, etc.).
- Remove **punctuation** → focus on meaningful words.

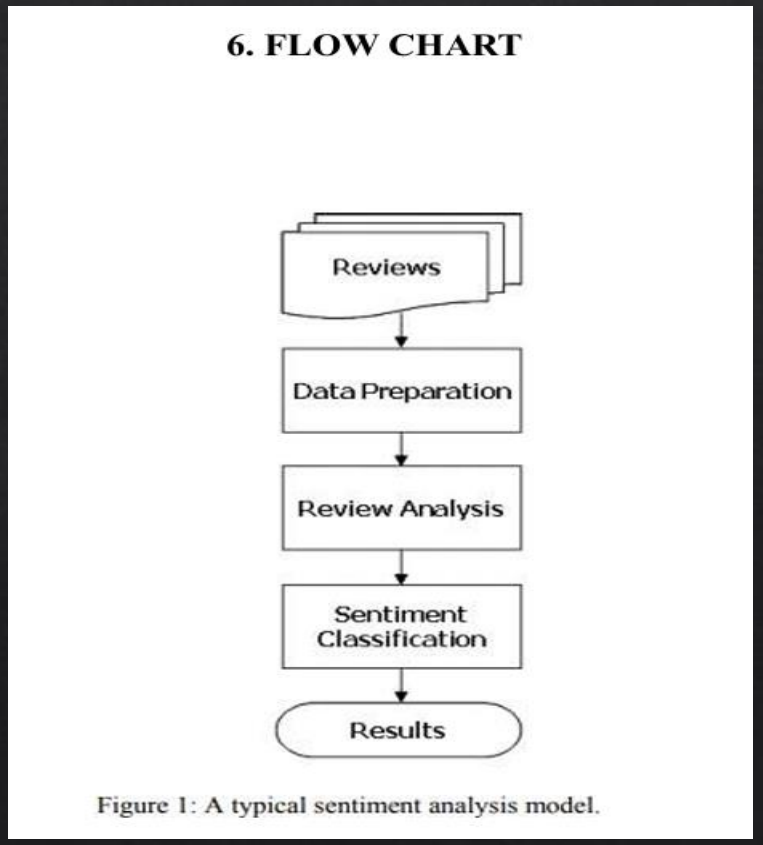
- **4. Feature Extraction**
- Use **TF-IDF vectorization** to convert text into numerical features.
- Adjust parameters (e.g., max_features) for dataset size.
- **5. Model Training**
- Split data into **train** and **test sets** (train_test_split).
- **6. Model Selection**
- Train and compare different classifiers:
  - **SVM** → works well with high-dimensional data.
  - **Random Forest** → ensemble of decision trees.
  - **Gradient Boosting** → sequential ensemble correcting errors.
- **7. Model Evaluation**
- Use metrics:
  - **Precision** → importance of avoiding false positives.
  - **Recall** → importance of avoiding false negatives.
  - **F1-score** → balance between precision and recall.
- Generate **classification report** (precision, recall, F1, support).
- Use **confusion matrix** to compare actual vs predicted labels.

SYESTEM DESIGN

- The sentiment analysis system is designed in modular form, handling each step efficiently from input to output. Raw Flipkart reviews are ingested into a pandas DataFrame in the **Input Data Module**, followed by the **Preprocessing Module**, where text is cleaned using regex, stop words removed, and words normalized through stemming/lemmatization. In the **Feature Extraction Module**, TF-IDF vectorization converts cleaned text into numerical features, with optional dimensionality reduction for efficiency. The **Model Training Module** trains SVM, Random Forest, and Gradient Boosting models on the extracted features, while the **Prediction and Voting Module** aggregates their outputs using majority voting for final sentiment classification. Finally, the **Evaluation and Visualization Module** assesses performance with confusion matrices, classification reports, accuracy scores, and generates charts to visualize sentiment distribution and model performance. This modular architecture ensures scalability, accuracy, and adaptability to different datasets.

SYESTEM DESIGN





RESULT

- The **SVM model** achieved good accuracy in classifying Flipkart reviews, especially in high-dimensional feature spaces. It showed **high precision for positive sentiments** but struggled slightly with negative ones. The **confusion matrix** revealed misclassifications mainly between neutral and negative reviews.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

- **Confusion Matrix**: The confusion matrix for the SVM model highlighted areas where the model confused neutral and negative sentiments, which is typical given the subjective nature of the data.

Accuracy: 0.9848156182212582

	precision	recall	f1-score	support
0	1.00	0.84	0.91	43
1	0.98	1.00	0.99	418
accuracy			0.98	461
macro avg	0.99	0.92	0.95	461
weighted avg	0.99	0.98	0.98	461

- .**Random Forest:**
- • **Accuracy**: The Random Forest model achieved a comparable accuracy to SVM, with the added benefit of being less prone to overfitting due to its ensemble nature. • **Precision and Recall**: Random Forest provided balanced precision and recall across all sentiment classes, though it was slightly less precise in distinguishing neutral sentiments.
- • **Confusion Matrix:** The confusion matrix for Random Forest showed a more balanced distribution of errors, with fewer instances of extreme misclassification compared to the other models.

Accuracy: 0.9544468546637744

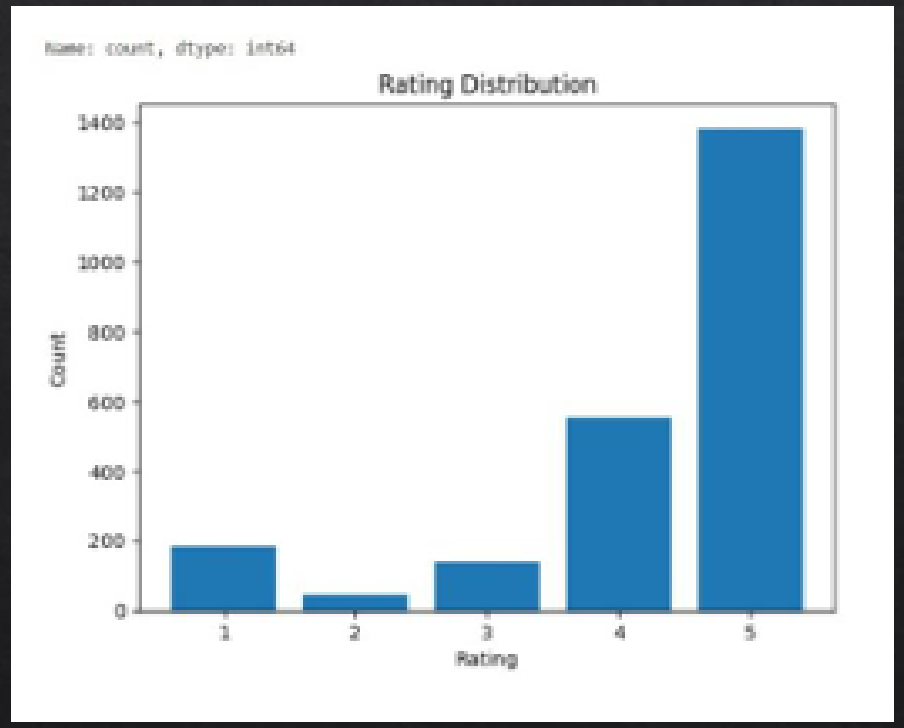
	precision	recall	f1-score	support
0	0.82	0.65	0.73	43
1	0.96	0.99	0.98	418
accuracy			0.95	461
macro avg	0.89	0.82	0.85	461
weighted avg	0.95	0.95	0.95	461

- **3.Gradient Boosting:**
- • **Accuracy:** Gradient Boosting was the most accurate model, achieving the highest overall accuracy. This is due to its iterative process of correcting errors from previous models, making it particularly effective in handling complex sentiment data.
- • Precision and Recall: This model excelled in precision across all classes, with particularly high recall for positive sentiments, indicating that it was very effective in identifying reviews with positive sentiment.
- • **Confusion Matrix:** The confusion matrix for Gradient Boosting showed the least amount of error, with very few instances of misclassification, demonstrating the model's strong predictive power.

Accuracy: 0.9848156182212582

	precision	recall	f1-score	support
0	1.00	0.84	0.91	43
1	0.98	1.00	0.99	418
accuracy			0.98	461
macro avg	0.99	0.92	0.95	461
weighted avg	0.99	0.98	0.98	461

- **Final Accuracy:** The ensemble (SVM + Random Forest + Gradient Boosting) achieved higher accuracy than any single model, reducing errors.
- **Sentiment Distribution:** Most reviews were **positive**, followed by fewer **neutral**, and the least were **negative**, highlighting improvement areas.
- **Evaluation Metrics:** The classification report showed balanced precision, recall, and F1-scores, while the confusion matrix confirmed minimal misclassifications, proving the model's robustness.




```
Confusion Matrix:
 [[394   0   0]
 [ 16   7   1]
 [  5   0  38]]
Classification Report:
              precision    recall  f1-score   support

    Positive       0.95      1.00      0.97       394
     Neutral       1.00      0.29      0.45        24
    Negative       0.97      0.88      0.93        43

    accuracy                           0.95       461
   macro avg       0.97      0.73      0.78       461
weighted avg       0.95      0.95      0.94       461

Percentage of Positive reviews: 90.02%
Percentage of Negative reviews: 8.46%
Percentage of Neutral reviews: 1.52%
Final Majority Voting Accuracy: 0.9522776572668112
```