**Name: Dhruv Patel**

**Enrollment Number: AU2444003**

**Course: M.Tech CSE**

**Course Name: CSE620 – Data Science Lab**

**Project Title: Flipkart Product Reviews - Sentiment Analysis and Prediction**

**1. Problem Definition**

The goal of this project is to analyze customer reviews for 5G mobile phones listed on Flipkart. The objectives are:

* To **scrape and collect reviews**, along with product ratings and names.
* To **clean and preprocess** the collected data.
* To **perform sentiment analysis** on customer reviews.
* To **predict the sentiment polarity** (Positive/Negative/Neutral) of new/unseen reviews.
* To **visualize insights** that can help consumers and vendors better understand product reception.

**2. Data Collection**

Data was collected using **web scraping techniques** via **Selenium** and **BeautifulSoup** in Python. The scrap.py script automates the process of:

* **Opening Flipkart**, searching for **5G mobile phones**, and navigating through product listing pages.
* **Extracting product URLs** and visiting individual product pages.
* Scraping:
  + **Product Name**
  + **Overall Product Rating**
  + **Individual Review Texts** from the "All Reviews" section (up to 8 pages/reviews per product).

**Key Features of the scrap.py Script:**

* Uses **Firefox WebDriver** in headless mode.
* Dynamic scraping with **wait conditions** for stability.
* Handles **pagination** of both products and reviews.
* Saves results in a structured format (flipkart\_reviews.csv) with the fields:
  + Product
  + Overall Rating
  + Review

This approach ensures a scalable and automated collection pipeline, capturing reviews for potentially hundreds of products with minimal manual intervention.

**3. Data Cleaning & Preprocessing**

**Data Import**:

* The cleaned and structured review data is loaded from the flipkart\_reviews.csv file generated during the scraping phase.

**Text Preprocessing**:

* **Lowercasing**: Converted all review text to lowercase for consistency.
* **Punctuation & Number Removal**: Removed all punctuation marks and numerical digits to focus only on meaningful text.
* **Stopwords Removal**: Eliminated common words (like “is”, “the”, “and”) that do not add significant value to sentiment understanding.
* **Lemmatization**: Reduced words to their base form (e.g., "running" to "run") to standardize similar terms.

**Sentiment Labeling**:

* Applied either:
  + **Rule-Based Models** (like TextBlob or VADER) to compute sentiment polarity and classify reviews as positive, negative, or neutral.
  + **Supervised Learning**: Trained a classification model on pre-labeled data to automatically predict sentiment classes for each review.

**Output**:

* The processed dataset now contains clean reviews along with associated sentiment labels, ready for analysis or model training.

**4. Exploratory Data Analysis (EDA)**

**Countplot of Number of Ratings**:

* A countplot was created to display the frequency distribution of overall product ratings (e.g., 1 to 5 stars).
* This helps in understanding the general trend of how products are rated on Flipkart.

**WordCloud of Review Texts**:

* A WordCloud visualization was generated to highlight the most frequently used words in customer reviews.
* Words appearing larger in the cloud indicate higher occurrence, offering a quick glimpse into common topics and sentiments.

**Countplot for Sentiment Distribution**:

* A countplot was plotted to show the number of reviews categorized as **positive**, **negative**, and **neutral**.
* This helps assess the sentiment balance and whether users are mostly satisfied or dissatisfied.

**Pie Chart of Sentiment Percentages**:

* A pie chart was used to display the **percentage distribution** of positive, negative, and neutral reviews.
* It provides a clearer picture of the proportion each sentiment class holds in the dataset.

**Boxplot for Outlier Detection**:

* A boxplot was generated on the sentiment scores to identify potential **outliers**.
* Outliers can indicate highly polarized or ambiguous reviews that may need special attention.

**Histogram of Sentiment Scores**:

* A histogram was plotted to visualize the distribution of sentiment polarity scores (from tools like TextBlob or VADER).
* This shows how reviews are spread across the sentiment spectrum from negative to positive.

**5. Feature Engineering**

**TF-IDF Vectorization**:

* Transformed the cleaned review texts into numerical representations using **TF-IDF (Term Frequency-Inverse Document Frequency)**.
* This technique gives importance to words that are frequent in a review but rare across other reviews, enhancing model performance.

**Sentiment Score Extraction**:

* Used tools like **TextBlob** or **VADER** to assign a **sentiment polarity score** to each review.
* These scores range from -1 (very negative) to +1 (very positive) and were added as features for training models.

These were used to train classification models or improve the sentiment analysis logic.

**6. Model Building**

**Implemented Models**:

* Trained and evaluated multiple classification models to predict review sentiments:
  + **Random Forest Classifier (RFC)**
  + **Decision Tree Classifier (DT)**
  + **Multinomial Naive Bayes (MNB)**

**Best Performing Model**:

* Among all, the **Multinomial Naive Bayes** model delivered the **highest accuracy of 91%**, making it the most effective choice for this task.
* Its performance highlights the suitability of probabilistic models for text classification, especially with features like TF-IDF.

**7. Model Evaluation Metrics**:

* The models were evaluated using:
  + **Accuracy** – to measure overall correctness.
  + **Precision, Recall, and F1-score** – to understand performance on each sentiment class.
  + **Confusion Matrix** – to visualize the classification results and detect any misclassifications.

**8. Web App Development using Streamlit**

**Purpose**:

* A simple and interactive **Streamlit web application** was developed to allow users to **analyze the sentiment** of any review they enter manually.

**Functionality**:

* Users can **input a review** in a text box.
* Upon clicking a button, the app **processes the text**, performs preprocessing, vectorization (using the trained TF-IDF), and **predicts the sentiment** using the best-performing model (Multinomial Naive Bayes).

**Sentiment Output**:

* The app classifies the entered review as:
  + **Positive**
  + **Negative**
  + **Neutral**

**User Experience**:

* The UI is intuitive and user-friendly, making it accessible for non-technical users as well.
* Results are displayed instantly with clear formatting and optional emoji-based indicators for sentiments.

**9. Conclusion**

* Successfully scraped thousands of reviews and built a predictive sentiment analysis system.
* Gained insights into how customer sentiments align with product ratings.
* Potential applications in **e-commerce analytics**, **brand reputation tracking**, and **recommendation systems**.