“Project Report”

Objective:

To analyze product review data from Amazon and classify the reviews as **positive**, **negative**, or **neutral** using Natural Language Processing (NLP) techniques and sentiment scoring tools.

📁 Dataset:

- **File Used:** `amazon.csv`
- Contains customer review data including:
  - `reviewText`, `reviewTime`, `reviewerName`
  - `overall` rating, `helpful` votes, and Wilson lower bound for ranking reviews

---

🛠️ Tools & Libraries Used:

- `pandas`, `numpy`, `matplotlib`, `seaborn`
- `TextBlob`, `VADER` from `nltk.sentiment`
- `wordcloud`, `plotly`, `cufflinks` for visualizations

✅ Steps Performed:

1. Data Loading & Cleaning:

- Loaded `amazon.csv` and removed the `Unnamed: 0` column
- Sorted by `wilson_lower_bound` to prioritize quality reviews
- Defined utility functions for:

- Missing value analysis
- Dataframe overview
- Duplicates check
- Quantile summaries
- Class (category) counts

2. Preprocessing:

- Text data cleaned using regex
- Sentiment scoring performed using:
  - **TextBlob** polarity for simple classification
  - **VADER** sentiment intensity for more nuanced scores

3. Sentiment Classification:

- Reviews labeled as:
  - **Positive** (polarity > 0)
  - **Negative** (polarity < 0)
  - **Neutral** (polarity = 0)

📊 Visualizations:

- **Word Clouds** to show most common words in reviews
- **Count & Pie Charts** for sentiment distribution
- **Interactive Bar Graphs** (using Plotly) for categorical summaries

---

🔍 Key Insights:

- Majority of reviews were **positive**, as indicated by both TextBlob and VADER analysis
- Word cloud revealed that positive reviews contain frequent praise words like *“great”, “love”, “easy”*
- Negative reviews focus on issues like *“broken”, “poor”, “waste”*

🗒️ Conclusion:

The sentiment analysis effectively categorizes customer opinions, making it useful for:

- Monitoring product feedback
- Understanding customer satisfaction
- Flagging negative reviews for customer support