README Project Chatbot Customer Support

🤖 Chatbot Customer Support on Twitter

This project aims to develop a chatbot capable of providing automated responses to customer reviews based on the sentiment of the review text.

📁Dataset

Source: Kaggle - Customer Support on Twitter
This dataset contains individual messages with a single primary input feature—the customer's text message—and a corresponding sentiment label indicating positive, neutral, or negative sentiment.

⚙ Tools

- Python (Pandas, NumPy, Scikit-Learn, Re, Joblib, Matplotlib, Seaborn, NLTK, Streamlit)
- Jupyter Notebook/VS Code

🎯 Purpose of Analysis

- Provide prompt and relevant responses to customer inquiries or complaints.
- Identify positive, neutral, and negative sentiments in customer messages to understand satisfaction levels and service issues.
- Leverage insights from sentiment analysis to improve services, products, or communication strategies.
- Reduce the workload of the customer service team by automatically handling routine queries.

🔎 Steps

1. Data Loading & Exploratory Data Analysis (EDA)

- Reading datasets using Pandas
- Checking missing values and descriptive statistics
- Examine message length, inbound distribution, number of reviews per date

2. Feature Engineering

- Text cleaning and lemmatization
- Create additional features for sentiment scores then convert scores into sentiment labels
- One-hot encoding for additional features

3. Data Splitting & Scaling

- Split the data into train (80%) and test (20%)
- The data was split using stratification to maintain balanced proportions of each sentiment label, and the text was then transformed into numerical representations using TF-IDF before being fed into the models.

4. Modeling

Model used:

- Multinomial Naive Bayes
- Logistic Regression
- Random Forest Classifier

5. Evaluation

- Classification Report (Precision, Recall, F1-support)
- Confusion Matrix
- Accuracy Score

📊 Visualization

- The visualizations indicate that the majority of customer messages are positive, while the remainder are negative or neutral.
- Temporal trends reveal fluctuations in sentiment, including spikes in negative messages during certain periods, which are likely associated with service issues.
- Overall, longer messages tend to convey more extreme sentiments, reflecting the intensity of customers' opinions regarding their experiences.

🧠Insight

1. Model Evaluation, the Logistic Regression model produced inconsistent sentiment predictions due to the limited dataset size, which caused the model to underperform when applied to real conversational inputs.
2. Model Enhancement with VADER, to overcome this limitation, the VADER Sentiment Analyzer was implemented. VADER is more effective in analyzing conversational or social media–style text and does not require a labeled dataset to deliver reliable sentiment predictions.
3. Technical Aspect, the combination of lexicon-based sentiment analysis (VADER) and rule-based intent detection enhances the chatbot's flexibility while reducing its dependency on large-scale training data.
4. Business Value, this chatbot can serve as a first-line customer support tool capable of automatically capturing user sentiment. Positive feedback can be utilized as testimonials, while negative feedback can be flagged for follow-up. This approach ultimately contributes to improved customer satisfaction and operational efficiency.

🚀 How to Run

- Ensure all required packages are installed:

  ```
  pip install pandas numpy scikit-learn re joblib matplotlib seaborn nltk streamlit
  ```

- Run the notebook in Jupyter or VS Code:

  ```
  jupyter notebook Project_ChatbotCustomerSupport.ipynb
  ```

- Run the Streamlit:

  ```
  streamlit run app.py
  ```