**Sentiment Analysis Model - Complete Workflow**

**1. Project Overview**

This project involves building a **binary sentiment analysis model** that classifies customer reviews as **positive or negative**. The workflow includes **data preprocessing, model training, evaluation, deployment, and testing** using Flask and Jupyter Notebook.

**2. Steps Followed**

**Step 1: Data Collection**

* The dataset consists of customer reviews.
* Initially, the dataset was stored in an **Excel file** (data.xlsx).
* The dataset contained **one column**, which was later split into **labels** (positive/negative) and **reviews**.

**Step 2: Data Preprocessing**

* Used **Pandas** to read and clean the dataset.
* Split the dataset into two columns:
  + **Labels** (Positive/Negative)
  + **Reviews** (Text data)
* Removed missing values and duplicates.

**Step 3: Feature Engineering (TF-IDF Vectorization)**

* Used **TF-IDF (Term Frequency-Inverse Document Frequency)** to convert text into numerical format.
* Saved the trained **TF-IDF vectorizer** as tfidf\_vectorizer.pkl for later use.

**Step 4: Model Training**

* Used **Logistic Regression** for training.
* Split the dataset into **training** and **testing** sets.
* Trained the model using TF-IDF transformed text.
* Saved the trained model as model.pkl.

**Step 5: Model Evaluation**

* Used accuracy, precision, recall, and F1-score to evaluate the model.
* Achieved a good performance in classifying reviews.

**Step 6: Model Deployment with Flask**

* Created a Flask web application.
* Loaded the model.pkl and tfidf\_vectorizer.pkl.
* Designed an **HTML interface** where users can input reviews.
* Integrated Flask to process user input and return **positive or negative** predictions.

**Step 7: Testing the Model on New Data**

* Loaded a **new Excel file (finaldata.xlsx)** containing fresh reviews.
* Applied the same **TF-IDF transformation**.
* Passed the transformed reviews into the trained **Logistic Regression model**.
* Predicted the sentiment for each review.
* Calculated the **percentage of positive and negative reviews**.
* Saved the predictions into predicted\_results\_with\_percentages.xlsx.

**Step 8: Error Handling & Fixes**

* Fixed **KeyError: 'review'** by ensuring column names match.
* Resolved **FileNotFoundError** by checking file paths.
* Fixed **string indexing errors** while reading data.
* Resolved **'DataFrame' object has no attribute 'append'** error by using .concat() instead.

**3. Final Outputs**

* **Trained Model**: model.pkl
* **TF-IDF Vectorizer**: tfidf\_vectorizer.pkl
* **Tested Data with Predictions**: predicted\_results\_with\_percentages.xlsx
* **Web App for User Testing**: Flask-based UI to classify user reviews.

**4. Future Enhancements**

* Improve the **front-end UI** for better user experience.
* Deploy the model on **AWS/GCP** for wider accessibility.
* Use **Deep Learning (LSTMs, Transformers)** for better accuracy.
* Expand the dataset for multi-class sentiment analysis (Neutral, Very Positive, etc.).