**Sentiment Analysis Project Report**

**Overview**

This Jupyter Notebook implements **sentiment analysis** using deep learning. The goal is to classify text into sentiment categories such as **positive** or **negative** by leveraging **Natural Language Processing (NLP)** techniques. The approach involves **word embeddings** and a **recurrent neural network (RNN) model** to achieve accurate classification.

**Notebook Breakdown**

**1. Data Preprocessing**

* Loads and processes the dataset.
* Uses **pre-trained GloVe embeddings** to convert words into numerical vector representations.
* Tokenizes and pads the text sequences.
* Splits the dataset into **training and testing sets**.

**2. Model Architecture**

* The sentiment classification model is based on a **Bidirectional Long Short-Term Memory (LSTM)** network.
* The architecture includes:
  + An **Embedding Layer** with pre-trained GloVe vectors.
  + A **Bidirectional LSTM Layer** to capture context from both directions.
  + Fully connected layers for **final classification**.

**3. Training and Optimization**

* The model is compiled with an appropriate **loss function** and **optimizer**.
* Training is performed over multiple epochs, and the performance is evaluated.
* Hyperparameters such as learning rate, batch size, and number of LSTM units are optimized.

**4. Evaluation and Performance**

* The trained model is tested on unseen data.
* **Final Accuracy on Training Data**: **86.24%**
* **Final Validation Accuracy**: **86.84%**
* Performance metrics such as precision, recall, and F1-score are also analyzed.
* Training loss and accuracy curves are plotted for better visualization.

**5. Results and Observations**

* The model achieves **high accuracy** in sentiment classification.
* **Pre-trained embeddings (GloVe) improve model performance**.
* The use of **Bidirectional LSTM** enhances the model’s ability to understand context better.
* Future improvements could include **fine-tuning embeddings** and **adding attention mechanisms**.

**Technical Details**

* **Programming Language**: Python 3.7.16
* **Libraries Used**: TensorFlow, Keras, NumPy, Pandas, Matplotlib
* **Data Handling**: Tokenization, Padding, Word Embeddings
* **Model Type**: Bi-directional LSTM-based classifier

**Conclusion**

This notebook serves as a **complete pipeline for sentiment analysis** using deep learning. It covers **data preprocessing, model building, training, evaluation, and visualization**. By leveraging **pre-trained embeddings and LSTM networks**, it provides an effective approach to sentiment classification.