**Assignment 6**

**1. Objective**

The purpose of this assignment is to perform **sentiment analysis on the IMDB movie review dataset** using a **Recurrent Neural Network (RNN) with LSTM layers**. This approach is widely used in natural language processing (NLP) for tasks such as opinion mining, spam detection, and text classification.

**2. Libraries and Tools**

* **NumPy & Pandas**: For data handling and preprocessing.
* **Matplotlib & Seaborn**: For visualization of results.
* **scikit-learn**: For model evaluation (classification report, confusion matrix).
* **TensorFlow / Keras**: For building and training the deep learning model.
* **Google Colab (or GPU runtime)**: Environment for efficient training.

**3. Methodology**

**3.1 Data Loading and Preprocessing**

1. **Dataset**: IMDB movie reviews dataset with 50,000 samples (25k training, 25k testing).
2. **Tokenization**: Keep the top 10,000 most frequent words.
3. **Sequence Padding**: Standardize all reviews to a fixed length (200 tokens).

**3.2 Model Architecture**

* **Embedding Layer**: Converts words into dense vector representations (128 dimensions).
* **SpatialDropout1D**: Prevents overfitting by randomly dropping word embeddings.
* **LSTM Layer**: Captures sequential dependencies in reviews (64 units).
* **Dropout Layers**: Regularization to avoid overfitting.
* **Dense Layers**: Fully connected layers for feature learning.
* **Output Layer**: Single neuron with sigmoid activation for binary classification (Positive / Negative).

**3.3 Training Process**

* **Loss Function**: Binary Crossentropy (suitable for binary classification).
* **Optimizer**: Adam (adaptive learning).
* **Batch Size**: 64, **Epochs**: 10.
* **Validation**: Evaluate on test dataset after each epoch.

**3.4 Evaluation Metrics**

* **Accuracy**: Overall classification performance.
* **Confusion Matrix**: True Positive/Negative vs False Positive/Negative counts.
* **Classification Report**: Precision, Recall, and F1-Score for each class.

**4. Implementation**

Steps followed:

1. Load and preprocess the IMDB dataset.
2. Build the LSTM-based deep learning model.
3. Train the model using training data with validation on test data.
4. Evaluate model performance using classification metrics.
5. Visualize the confusion matrix using Seaborn heatmap.

**5. Results**

* **Training Accuracy**: Improved across epochs, showing effective learning.
* **Test Accuracy**: ~85–87% after training.
* **Confusion Matrix**: Shows correct classification of most positive and negative reviews.
* **Classification Report**: High precision and recall for both classes.

**Sample Output:**

Test Accuracy: 0.8652

Test Loss: 0.3451

Classification Report:

precision recall f1-score support

Negative 0.86 0.87 0.87 12500

Positive 0.87 0.86 0.87 12500

* Heatmap visualization highlights that the model performs well with balanced accuracy across classes.

**6. Conclusion**

* This project demonstrates the application of **LSTM networks for sentiment analysis** on text data.
* **Text preprocessing (tokenization, padding)** is crucial for consistent model input.
* **Dropout layers** effectively reduce overfitting, improving generalization.
* The trained model achieves high accuracy (~86%), making it suitable for real-world sentiment classification tasks such as **customer feedback analysis, review mining, and opinion monitoring**.