Report for Intelligent Document Classifier Assignment

1. Objective

The goal of this assignment is to build an end-to-end solution for classifying documents into predefined categories using Natural Language Processing (NLP) and deep learning techniques.

---

2. Dataset Overview

The dataset contains 3000 text documents categorized into predefined classes:

- **Columns**:
    - ID: Unique identifier for each document.
    - Text: The document text.
    - Category: The label indicating the document's class (e.g., Legal, Scientific, E-commerce).
- **Data Details**:
    - Number of records: 3000
    - Number of unique categories: 3

---

3. Steps Performed

3.1 Data Exploration and Preprocessing

- **Loaded the dataset** and checked for missing values (none found).
- **Preprocessing Steps**:
    - Removed stop words using NLTK’s predefined list.
    - Converted text to lowercase.
    - Tokenized text into individual words.
    - Applied lemmatization to normalize words.
    - Vectorized text using **TF-IDF** for traditional ML models and **Tokenizer with padding** for deep learning models.

3.2 Model Development

Traditional ML Model: Logistic Regression

- Vectorized text data using TF-IDF.
- Split the data into training and testing sets (80:20).
- Trained Logistic Regression using GridSearchCV to optimize the C parameter.

Deep Learning Model: LSTM

- Tokenized and padded text data.
- Built an LSTM model with:
    - Embedding layer (10,000 words, 128 dimensions).
    - LSTM layer with 128 units.
    - Dropout layer to prevent overfitting.
    - Dense layer for final classification.
- Optimized hyperparameters (LSTM units, dropout rate, batch size) using GridSearchCV with scikeras.

3.3 Evaluation

Metrics Used:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-score**

Results:

Metric	Logistic Regression	LSTM Model
Accuracy	87.4%	90.2%
Precision	88.0%	91.0%
Recall	86.5%	89.5%
F1-score	87.2%	90.2%

---

4. Conclusion

The intelligent document classifier achieved high accuracy using both Logistic Regression and LSTM. The LSTM model outperformed Logistic Regression due to its ability to capture sequential dependencies in text data. Optimized hyperparameters further enhanced the performance.

References

1. TensorFlow Documentation
2. scikit-learn Documentation
3. NLTK Documentation

Video Explanation :

https://www.loom.com/share/447660079d0d4383b244727e14c63f27?sid=7082c730-7149-46be-9cfa-76bbacec346a