Sentiment Analysis Project

Project Description

This project aims to analyze the sentiment of product reviews using three different approaches:

- **Machine Learning (ML)** using Logistic Regression,
- **Deep Learning (DL)** using LSTM (Long Short-Term Memory),
- **Large Language Models (LLM)** using a fine-tuned BERT model from Hugging Face.

The project compares the performance of these models across multiple datasets to determine which method is the most effective for sentiment classification.

Objectives

- Build models using ML, DL, and LLM to classify product reviews as positive or negative.
- Compare the models' performance based on evaluation metrics like accuracy, precision, recall, and F1-score.
- Deploy the models via a Flask API to allow real-time predictions.

Folder Structure

├── data/ │ ├── dataset_1.csv │ ├── dataset_2.csv │ └── dataset_3.csv ├── models/ │ ├── ml_model.pkl │ ├── dl_model.h5 │ └── llm_model.pt ├── notebooks/ │ └── Sentiment_Analysis.ipynb ├── src/ │ ├── ml_model.py │ ├── dl_model.py │ └── llm_model.py ├── results/ │ └── comparison_results.csv ├── app.py ├── requirements.txt └── README.md

Datasets

- The `data/` folder contains three datasets in `.csv` format, each consisting of two columns:
  - `review` : The text of the product review.
  - `sentiment` : The label indicating positive (1) or negative (0) sentiment.

Setup & Installation

1. **Clone the repository:**

   bash

   Copy code

```
git clone
cd sentiment_analysis_project
```

2. **Install dependencies:** Install the required libraries by running the following command:

   bash
   Copy code

   ```
   pip install -r requirements.txt
   ```

3. **Prepare datasets:**

   - Place your CSV datasets inside the `data/` folder.
   - Ensure that the datasets are formatted with two columns: `review` and `sentiment`.

4. **Running Models:**

   - Navigate to the `src/` directory and run the specific model scripts:

   Machine Learning Model (Logistic Regression)

   bash
   Copy code

   ```
   python src/ml_model.py
   ```

   Deep Learning Model (LSTM)

   bash
   Copy code

   ```
   python src/dl_model.py
   ```

   Large Language Model (BERT)

   bash
   Copy code

```
python src/llm_model.py
```

5. **Running Flask API:** To serve predictions using the trained models, run the Flask API:

   bash

   Copy code

   ```
   python app.py
   ```

   This will start the API server locally at `http://localhost:5000`.

API Endpoints

- **ML Model Prediction (Logistic Regression):**
  - Endpoint: `/predict/ml`
  - Method: `POST`
  - Payload:

    json

    Copy code

    ```
    {
      "text": "This product is amazing!"
    }
    ```

- **DL Model Prediction (LSTM):**
  - Endpoint: `/predict/dl`
  - Method: `POST`
  - Payload:

    json

    Copy code

```
{
  "text": "Terrible quality, would not buy again."
}
```

- **LLM Model Prediction (BERT):**
  - Endpoint: `/predict/llm`
  - Method: `POST`
  - Payload:

    json

    Copy code

    ```
    {
      "text": "The performance is outstanding!"
    }
    ```

Each endpoint will return a JSON response with the predicted sentiment (1 for positive and 0 for negative)