**Machine Learning Project: Sentiment Analysis with DistilBERT**

This project uses the **DistilBERT** model for binary sentiment classification (positive/negative) of movie reviews. It includes training a model on the IMDB dataset and deploying it as a web application using Flask.

**Prerequisites**

Before running the project, ensure you have the following installed:

* **Python 3.7+**
* **pip** (for installing Python packages)

Additionally, you will need the following Python packages:

* transformers
* torch
* datasets
* flask
* scikit-learn
* evaluate
* numpy
* pandas

**Setup**

Follow these steps to set up the project:

**1. Clone the repository**

bash

Copy code

git clone <repository\_url>

cd <repository\_name>

**2. Install dependencies**

You can install the necessary dependencies by running the following command:

bash

Copy code

pip install -r requirements.txt

If you don't have a requirements.txt file, you can manually install the dependencies:

bash

Copy code

pip install transformers torch datasets flask scikit-learn evaluate numpy pandas

**3. Download or Prepare the Dataset**

The project uses the IMDB dataset for training the model. The dataset will be automatically loaded from a CSV file (IMDB Dataset.csv) in the script. Ensure this file exists in the same directory as the code, or adjust the code to point to the correct location.

**4. Train the Model**

* Open the Jupyter Notebook (main.ipynb).
* Run each cell sequentially to train the model on the IMDB dataset.
* The trained model will be saved in the results folder.

**5. Flask Application**

Once the model is trained, you can deploy it as a Flask web application.

* Ensure the results directory (containing the saved model and tokenizer) is in the same directory as the Flask app (app.py).
* Run the Flask application by executing:

bash

Copy code

python app.py

This will start a local development server (default on http://127.0.0.1:5000/).

**6. Use the Web Application**

* Navigate to http://127.0.0.1:5000/ in your browser.
* You’ll see a form where you can input a movie review.
* Once you submit the review, the model will predict whether the review is "positive" or "negative."

**Code Overview**

**main.ipynb:**

* **Data Preprocessing**: The notebook loads the IMDB dataset, processes the reviews, and tokenizes them using the DistilBERT tokenizer.
* **Model Training**: It trains the DistilBERT model on the IMDB dataset for binary sentiment classification and saves the trained model.
* **Evaluation**: It evaluates the model's performance using metrics like accuracy, precision, recall, F1-score, and AUC.

**app.py:**

* **Flask API**: This file contains a Flask app that serves the trained model. It includes:
  + A homepage with a form to submit reviews.
  + An API endpoint (/predict) that accepts POST requests with the review text, processes the input, and returns a prediction (positive or negative).

**Model Details**

The model used in this project is **DistilBERT** (distilbert-base-uncased), a smaller, faster version of BERT for sequence classification.

**Evaluation Metrics**

The model is evaluated using the following metrics:

* **Accuracy**: Proportion of correct predictions.
* **Precision**: Proportion of positive predictions that are actually positive.
* **Recall**: Proportion of actual positive instances that are predicted correctly.
* **F1-Score**: Harmonic mean of precision and recall.
* **AUC (Area Under the Curve)**: Measures the ability of the model to distinguish between classes.

**License**

This project is licensed under the MIT License - see the LICENSE file for details.

**Acknowledgments**

* The IMDB dataset is publicly available and was used for training and evaluation.
* Hugging Face's transformers library was used for the pre-trained DistilBERT model.
* Flask was used to build the web application.