This README provides a detailed overview of all the code files included in the project. It explains their purpose, functionality, and how they fit into the overall workflow of the Movie Recommendation System using Transformers. Follow the instructions below to navigate and use the files effectively.

In order to execute the Demo_NLP (1).ipynb, use the Stat_Software_Project_demo.zip file which consists of all the required finetuned models and files.

---

1. Baseline_Models.ipynb

Purpose:

This notebook implements baseline models for genre classification and recommendation using traditional machine learning techniques.

Key Features:

- Uses **TF-IDF vectorization** to extract features from plot summaries.
- Implements **Logistic Regression** and **Naive Bayes** models for genre classification.
- Evaluates model performance using accuracy and F1-scores.
- Demonstrates the limitations of traditional approaches in capturing semantic context.

When to Use:

Start with this file to understand the baseline performance of classical models and their limitations, which justify the use of transformers.

---

2. Distili_Bert_final.ipynb

Purpose:

This notebook contains the implementation of the fine-tuned **DistilBERT** model for genre classification and recommendation.

Key Features:

- Fine-tunes **DistilBERT** on the CMU Movie Summary Corpus for multi-class genre classification.
- Handles class imbalance using **weighted loss functions**.

- Uses enhanced inputs (plot summaries + metadata).
- Includes hyperparameter tuning (learning rate, batch size, epochs).
- Provides performance metrics like accuracy, precision, recall, and F1-scores.

When to Use:

Run this file to fine-tune DistilBERT on the dataset and achieve state-of-the-art genre classification accuracy.

---

3. SemanticMatching (1).ipynb

Purpose:

This notebook focuses on the recommendation task, matching user queries with movie plot summaries using semantic embeddings.

Key Features:

- Generates dense embeddings for plot summaries using **Sentence-BERT (SBERT)**.
- Uses **FAISS** (Facebook AI Similarity Search) for efficient similarity computation and indexing.
- Recommends top-n movies based on cosine similarity scores.
- Compares the performance of **Sentence-BERT** and **TF-IDF-based cosine similarity**.

When to Use:

Run this file to generate recommendations for user queries and evaluate the effectiveness of semantic matching approaches.

---

4. Demo_NLP (1).ipynb

Purpose:

This notebook serves as a demonstration of the entire pipeline, integrating genre classification, semantic matching, and question answering.

Key Features:

- Shows how user queries flow through genre classification and recommendation.

- Combines the outputs of DistilBERT and Sentence-BERT for end-to-end functionality.
- Demonstrates the system's interactivity with sample queries.

When to Use:

Use this file to see the complete system in action and test different user queries for recommendations.

---

5. QA (1).ipynb

Purpose:

This notebook implements the **Question Answering (QA)** module, fine-tuned on synthetic data for answering questions about recommended movies.

Key Features:

- Fine-tunes **DistilBERTForQuestionAnswering** using plot summaries and metadata as context.
- Handles extractive question answering tasks (e.g., "Who directed this movie?").
- Evaluates performance on sample QA pairs.

When to Use:

Run this file to enable question-answering functionality for user queries about recommended movies.

---

6. TFIDF_vectoriser2.ipynb

Purpose:

This notebook contains the implementation of the TF-IDF vectorizer for feature extraction.

Key Features:

- Generates sparse representations of movie plot summaries.
- Evaluates similarity using **cosine similarity**.
- Serves as a baseline for the semantic matching task.

When to Use:

Use this file as a starting point to understand how baseline feature extraction methods compare to transformer-based embeddings.
