Sentiment Analysis (Movie Reviews / Tweets)-07/10/25(Tuesday):

Sentiment Analysis Project Report (Movie Reviews):

I. Objective

The goal of this project was to build a machine learning pipeline to classify text sentiment (positive or negative) from movie reviews. This exercise provided an introduction to core Natural Language Processing (NLP) techniques, including text preprocessing, feature extraction, and model evaluation.

II. Data Preparation and Cleaning Steps

The process began with loading the synthetic IMDb dataset and preparing the text for modeling.

Step	Action Taken	Purpose
Data Inspection	Inspected structure and identified a slight class imbalance (3 Positive vs. 2 Negative reviews).	Confirmed data readiness.
Cleaning	Handled missing values and removed duplicate entries.	Ensured data quality.
Preprocessing	Applied a custom function to:**1. Convert text to lowercase.**2. Remove HTML tags and punctuation.**3. Tokenize text.**4. Remove English stopwords (e.g., 'the', 'a').**5. Apply Lemmatization (e.g., 'running' → 'run').	Standardized text and reduced vocabulary noise.

III. Feature Extraction

Clean text was converted into a numerical format using **TF-IDF (Term Frequency-Inverse Document Frequency)**.

- **Rationale:** TF-IDF assigns weights to words based on their frequency in a document relative to their frequency across the entire corpus. This prioritizes words that are *distinguishing* (like 'terrible' or 'superb') over common, non-sentiment words.
- **Data Split:** The final dataset was split into **70% Training** and **30% Testing** for unbiased evaluation.

IV. Model Training and Evaluation

Logistic Regression and Random Forest models were trained and evaluated on the test set.

A. Evaluation Metrics

Metric	Logistic Regression	Random Forest
Accuracy	0.5	0.5
Precision	0.5	0.5
Recall	1	1
F1-Score	≈0.67	≈0.67

B. Model Selection

Selected Model: Logistic Regression

Both models performed identically on the small test set. Logistic Regression was chosen as the final model due to its **simplicity, speed, and inherent interpretability** (allowing easy extraction of word sentiment weights), making it an excellent baseline classifier.

C. Confusion Matrix Analysis

The confusion matrices for both models were identical :

True Label \ Predicted Label	Negative	Positive
Negative (0)	0 (TN)	1 (FP)
Positive (1)	0 (FN)	1 (TP)

Insight: The models exhibited behavior typical of classifiers trained on very small, slightly imbalanced data: they defaulted to predicting the **majority class** (Positive) for all test samples. This resulted in perfect **Recall** (finding all positive cases) but poor **Precision** (mistaking the negative case as positive).

V. Reusable Pipeline and Final Insights

A streamlined **Scikit-learn Pipeline** was created, combining the TF-IDF Vectorizer and the Logistic Regression classifier. This pipeline was saved as sentiment_pipeline.joblib for future deployment.

A. Pipeline Test Results

+

Test Sample	Model Prediction
The movie was a total disaster, slow and the worst experience.'	Negative
Highly recommended! The plot twist was genuinely surprising and the acting superb.'	Positive
It was just okay, nothing special, but not bad either.'	Positive

B. Insights on Word Contributions (Feature Importance)

The coefficients of the Logistic Regression model reveal which words most strongly influence the sentiment prediction:

Sentiment	Top 5 Contributing Words (Positive Weights)		
Positive	brilliant, masterpiece, liked, cinema, great		
Negative	worst, skip, seen, year, movie		

Conclusion: The model correctly identified strong sentiment adjectives (worst, brilliant) as having the highest predictive power, successfully demonstrating the basic mechanics of sentiment classification.

Project Summary: Sentiment Analysis:

This project successfully developed and evaluated a machine learning pipeline to classify movie review sentiment as either Positive or Negative, fulfilling all requirements for a real-world NLP scenario.

1. Key Steps

1. **Data Preparation:** A small, synthetic dataset of movie reviews was loaded, cleaned (lowercase conversion, punctuation removal, stop word removal), and normalized using **Lemmatization**.
2. **Feature Extraction:** The cleaned text was converted into numerical features using the **TF-IDF** (Term Frequency-Inverse Document Frequency) method.
3. **Model Training:** Both **Logistic Regression** (selected as the final model due to its interpretability and speed) and **Random Forest** classifiers were trained.

2. Results & Deliverables

Deliverable	Outcome
Model Performance	Both models achieved an Accuracy of 50% and an F1-Score of ≈0.67 on the small test set.
Model Behavior	The Confusion Matrix showed both models acted as dummy classifiers on the small test set, predicting "Positive" for every review (resulting in 1 True Positive and 1 False Positive).
Insights	Logistic Regression coefficients revealed that words like 'worst' and 'skip' strongly contributed to negative sentiment, while 'brilliant' and 'masterpiece' contributed to positive sentiment.
Final Product	A reusable Scikit-learn Pipeline combining the TF-IDF vectorizer and the Logistic Regression model was created and saved.

3. Conclusion

The project successfully demonstrated proficiency in the core NLP concepts of **text preprocessing**, **TF-IDF feature extraction**, and **classification modeling** using industry-standard Python libraries.