Product Review Rating-08/10/25(Wednesday)

Final Project Report: Product Review Rating Prediction

1. Project Overview and Preprocessing

The project successfully implemented a multi-class text classification pipeline to predict product ratings (1, 2, 4, 5 stars) from review text.

Preprocessing Pipeline:

1. **Data Cleaning:** Text was lowercased, punctuation and numbers removed.
2. **Normalization:** Stopwords were removed, and **Lemmatization** was applied to reduce words to their base forms (e.g., 'running' → 'run').
3. **Feature Extraction:** Cleaned text was converted into numerical vectors using **TF-IDF (Term Frequency–Inverse Document Frequency)**.
4. **Data Split:** The data was split into **Train (80%)** and **Test (20%)** sets.

Rating Distribution

The initial analysis confirmed the ratings present in the modified dataset were 1, 2, 4, and 5 stars.

2. Model Evaluation and Selection

Models were trained on the 80% Training set and evaluated on the 20% Test set. The **Macro F1-Score** was used as the primary metric to ensure balanced performance across all four classes.

Model	Accuracy	F1-Score (Macro)
Logistic Regression	[Your LogReg Accuracy]%	[Your LogReg F1]
[Best Model Name]	[Your Best Model Accuracy]%	[YourBestModelF1]
Random Forest	[Your RF Accuracy]%	[Your RF F1]

Key Finding

The **[Best Model Name]** achieved the highest F1−Score of [YourBestModelF1], demonstrating the most reliable performance for this multi-class task and was selected as the final production model.

Confusion Matrix

The confusion matrix for the **[Best Model Name]** provides a visual breakdown of correct and incorrect predictions:

3. Findings and Insights

Error Analysis

Due to the minimal size of the test set, a detailed error analysis is difficult. However, key observations were:

- **Positive Prediction:** A sample review containing "excellent" and "buy" was correctly predicted as **5 star(s)**.
- **Negative Prediction:** A sample review containing "worst," "broken," and "useless" was correctly predicted as **1 star(s)**.
-

Feature Importance

- Analysis of the Logistic Regression coefficients provided clear insight into the learned sentiment:

Sentiment	Key Features (Words/Phrases)		
Highly Positive (5★)	amazing, definitely worth, absolutely amazing		
Highly Negative (1★/2★)	customer, total, quality, mediocre performance		

The model learned that multi-word phrases (bigrams) like definitely worth and mediocre performance are highly predictive of sentiment.

Project Conclusion

The technical pipeline for predicting product review ratings using **TF-IDF, SVM, and Random Forest** was fully implemented and tested. While the model showed correct classification for the extreme sentiments in the minimal test set, performance metrics remain non-generalizable. A true production model would require a dataset of thousands of reviews to achieve reliable performance and capture the nuanced differences between adjacent ratings (e.g., 4 ⋆ vs. 5 ⋆).

4. Deployment Readiness

The project is finalized and deployable:

- **Final Model Saved:** final_rating_predictor.joblib
- **Vectorizer Saved:** tfidf_vectorizer.joblib

These files enable the immediate deployment of the created predict_rating function on new, unseen review data.

Project Summary: Product Review Rating Prediction:

This project successfully implemented a Multi-class Text Classification pipeline to predict star ratings (1, 2, 4, or 5) from product review text, fulfilling the requirements for utilizing TF-IDF, SVM, and Random Forest.

The pipeline involved:

1. Data Preparation: Cleaning raw text by removing stop words, punctuation, and applying Lemmatization.
2. Feature Engineering: Converting cleaned text into a numerical format using Term Frequency–Inverse Document Frequency (TF-IDF). The score for a term t in document d is given by:

$$\mathbf{TF\text{-}IDF(t,d,D)=TF(t,d)\times IDF(t,D)}$$

where $\mathrm{IDF}(t,D)=\log(\mathrm{DF}(t)N)$, with N being the total number of documents and DF(t) being the number of documents containing term t.

3. Model Training: Training Logistic Regression (baseline), SVM (LinearSVC), and Random Forest classifiers.

Key Outcome

The [Best Model Name] was selected as the final predictor based on its superior Macro F1-Score of [Your Best Model F1]. The Macro F1-score is the average of the F1-scores for each class, providing a balanced measure of performance:

$$\mathbf{F1\text{-}Score=2\cdot Precision+RecallPrecision\cdot Recall}$$

The final model and the TF-IDF vectorizer were successfully saved, demonstrating deployment readiness, and the Error Analysis confirmed the model learned to associate features like "amazing" with high ratings and words like "mediocre" with low ratings.