Sentiment Score Prediction

Objective

This report details the iterative process of developing models for predicting review scores. After exploring five different approaches, a final model was identified that delivered the highest accuracy on the leaderboard. The aim of the project was to predict sentiment scores for reviews based on textual data. Through a series of approaches, various methods were explored, including nearest neighbors, TF-IDF and clustering, ultimately achieving the best performance using a combination of feature engineering, clustering and a Naive Bayes model.

Best Approach: Multinomial Naive Bayes with TF-IDF and Clustering

Methodology

The best-performing approach involved feature engineering, dimensionality reduction, clustering and classification using a Multinomial Naive Bayes model.

1. **Data Cleaning**: Addressed issues like inconsistent quote usage to standardize data formatting.
2. **Feature Engineering**:
   - Extracted `HelpfulnessRatio` and `ReviewLength` to capture review informativeness and verbosity.
3. **Text Embeddings Using TF-IDF**:
   - Processed the *Text* column with `CountVectorizer`, limiting to 3,000 features for memory efficiency.
   - Used Truncated SVD to reduce dimensionality to 100 features, striking a balance between data representation and computational efficiency.
4. **Clustering**:
   - Grouped reviews into clusters using KMeans, adding a `ClusterLabel` feature to represent latent groups.
5. **Modeling with Multinomial Naive Bayes**:
   - Used `MinMaxScaler` to ensure all features were non-negative (a requirement for Multinomial Naive Bayes).
   - The combined feature set was effective for sparse data and high dimensionality.
6. **Testing and Submission**:
   - Preprocessed the test set similarly, filled missing values and predicted scores with the trained model.

Results

- **Public Score**: 0.53369
- **Private Score**: 0.53505

Analysis and Observations

This approach effectively balanced feature diversity and computational efficiency. The Multinomial Naive Bayes model performed well with sparse and high-dimensional data. Incorporating KMeans clustering added a distinct categorical feature that helped differentiate reviews, capturing latent structure in the data and improving model performance.

Additional Approaches and Conceptual Comparisons

To achieve the final model, several approaches were attempted, each with unique conceptual foundations that affected their effectiveness. Below is a summary of these approaches and their conceptual differences from the best model.

Approach 1: Nearest Neighbors with TF-IDF and SVD

1. **Concept**:
   - Used K-Nearest Neighbors (KNN) with TF-IDF embeddings to predict scores based on the most similar reviews.

- Conceptually, this approach relied on the assumption that similar reviews have similar scores.

2. **Impact on Scores**:
   - KNN struggled with sparse embeddings and high dimensionality, resulting in lower scores.
   - The model's dependency on exact text similarity was limiting, especially for short summaries.
3. **Distinct Characteristics**:
   - This approach used distance-based predictions, which did not capture broader patterns in review structure that clustering enabled in the final approach.

Configuration	Public Score	Private Score
Summary TF-IDF	0.47598	0.47459
Text TF-IDF	0.41386	0.41300

Approach 2: FAISS-based Nearest Neighbors for Fast Cosine Similarity

1. **Concept**:
   - Leveraged FAISS on GPU for efficient similarity searches, focusing on approximate nearest neighbors in TF-IDF space.
   - Reduced computation time significantly and explored both *Summary* and *Text* columns.
2. **Impact on Scores**:
   - FAISS accelerated the computation but did not improve accuracy due to similar limitations as Approach 1, such as reliance on surface-level text similarity.
3. **Distinct Characteristics**:
   - Used FAISS to optimize KNN but still lacked latent structure learning. This approach showed how even optimized methods need robust feature engineering to improve accuracy.

Configuration	Public Score	Private Score
Summary Column	0.47598	0.47459
Text Column	0.41386	0.41300

Approach 3: Gradient Boosting with TF-IDF Features and Feature Engineering

1. **Concept**:
   - Gradient Boosting combined with TF-IDF embeddings and additional features. This tree-based model attempted to capture non-linear patterns in both text and numerical features.
2. **Impact on Scores**:
   - Gradient Boosting captured more complex relationships in the data than KNN, leading to improved scores. However, the computational demand was high, and it required significant hyperparameter tuning.
3. **Distinct Characteristics**:
   - Unlike Naive Bayes, this approach leveraged complex non-linear patterns in engineered features, but with diminishing returns due to overfitting risk in a high-dimensional sparse feature space.

Public Score	Private Score
0.53369	0.53505

Approach 4: Ensemble Model with Voting Classifier

1. **Concept**:
   - Combined multiple classifiers (Gradient Boosting, Random Forest, Logistic Regression) with voting to improve robustness.
   - This model aimed to stabilize predictions by averaging across diverse algorithms.
2. **Impact on Scores**:

- Voting improved consistency but did not outperform Naive Bayes with clustering, as the ensemble required more resources and struggled with sparse features.

3. **Distinct Characteristics**:
    - While robust, ensemble models require extensive tuning and large feature sets to be effective, leading to high complexity without a proportionate increase in performance for this dataset.

Public Score	Private Score
0.53369	0.53505

Summary of Results

Approach	Public Score	Private Score
Best Approach	0.53369	0.53505
Approach 1 - Summary TF-IDF	0.47598	0.47459
Approach 1 - Text TF-IDF	0.41386	0.41300
Approach 2 - Summary FAISS	0.47598	0.47459
Approach 2 - Text FAISS	0.41386	0.41300
Approach 3 - Gradient Boosting	0.53369	0.53505
Approach 4 - Voting Ensemble	0.53369	0.53505

Conclusion

Iterative experimentation across these conceptually diverse approaches provided insights into text similarity modeling. The best approach emerged by combining clustering, TF-IDF embeddings and feature engineering, which effectively balanced model accuracy with computational efficiency. Key insights that contributed to final model's success included:

1. **Feature Engineering**: Helpfulness and review length were significant indicators of review quality.
2. **Latent Patterns**: Clustering captured underlying themes in reviews, adding depth to feature representation.
3. **Sparse Representation**: Naive Bayes excelled in sparse data settings, contrasting with the ensemble approaches which faced challenges with high-dimensional embeddings.

Ultimately, the best approach demonstrated that thoughtful feature selection, dimensionality reduction and efficient classifiers can outperform complex or computationally intensive methods for specific high-dimensional text data.