Midterm Report

Data Preprocessing

Starting with the train.csv dataset, I implemented preprocessing steps to ensure data consistency. This involved removing rows containing missing values, followed by a random shuffle to avoid unintended bias. I created nine stratified samples of 300,000 rows each to ensure that each subset maintains the same distribution of a particular feature (in this case, 'Score') as the original dataset. Missing values in the "text" and "summary" fields were filled with default values, preserving dataset completeness and minimizing bias in the training phase.

Exploratory Data Analysis (EDA)

EDA guided feature engineering decisions by identifying patterns and relationships within the dataset:

- **Score Distribution**: An initial score distribution visualization highlighted imbalances across sentiment classes, prompting the application of stratified sampling to counteract potential bias.
- **Missing Values**: Missing entries in the "text" and "summary" columns were identified and filled, ensuring data integrity for downstream analysis.
- **Correlation Matrix**: A heatmap of numerical features revealed relationships between them, helping identify influential factors on sentiment scores and minimizing multicollinearity in the model.
- **Review Length Analysis**: The relationship between review length and sentiment intensity was explored. This analysis suggested that longer reviews often had more extreme sentiments, whether positive or negative.
- **Review Length vs. Score**: Scatter plots examining review lengths across sentiment scores provided additional insights for feature engineering.
- **Score Trends Over Time**: By converting timestamps to datetime objects, I analyzed trends in average scores over time, examining shifts in user expectations or product quality.

Extracted Features for Model Enrichment

I defined a custom transformer class, Naive, which inherits from BaseEstimator and TransformerMixin. This class is designed to compute term frequencies and apply Laplace smoothing to calculate smoothed probabilities for each class label. In the fit method, I calculate the log ratio of probabilities between class 1 and class 0 and store these as ratio weights. The transform method then applies these ratio weights to the input data.

- **ReviewLength**: This feature captured the character count of each review, with longer reviews often signaling detailed and strong sentiments.

- **TextSubjectivity and SummarySubjectivity**: Calculated using TextBlob, these scores gauged subjectivity in the review text and summaries. Highly subjective content, scoring close to 1, often indicated personal opinions, influencing sentiment interpretation.
- **TextSentimentPolarity and SummarySentimentPolarity**: These polarity scores, also from TextBlob, provided a direct sentiment metric for review text and summaries, ranging from -1 (very negative) to 1 (very positive).

By transforming qualitative text data into these quantitative metrics, the model could detect patterns more efficiently, ultimately enhancing predictive performance.

Naive Bayes for Probabilistic Feature Enrichment

I incorporated Naive Bayes transformations due to its computational efficiency and strength in text classification tasks. A research paper by Rashedi et al. (2018) explains that integrating Naive Bayes-generated class probabilities can diversify feature sets, capturing sentiment distributions within text data and creating class-specific probability features that enhance sentiment differentiation. In my approach, Naive Bayes was applied to the TF-IDF vectorized data, generating probabilities that represent the likelihood of each review belonging to a specific sentiment class. These probabilities were then added as new features to the dataset, enriching it with probabilistic context. This approach was also inspired by Shahiri et al. (2015), where combining Naive Bayes probabilities with another classifier significantly improved prediction accuracy.

Random Forest as the Primary Classifier

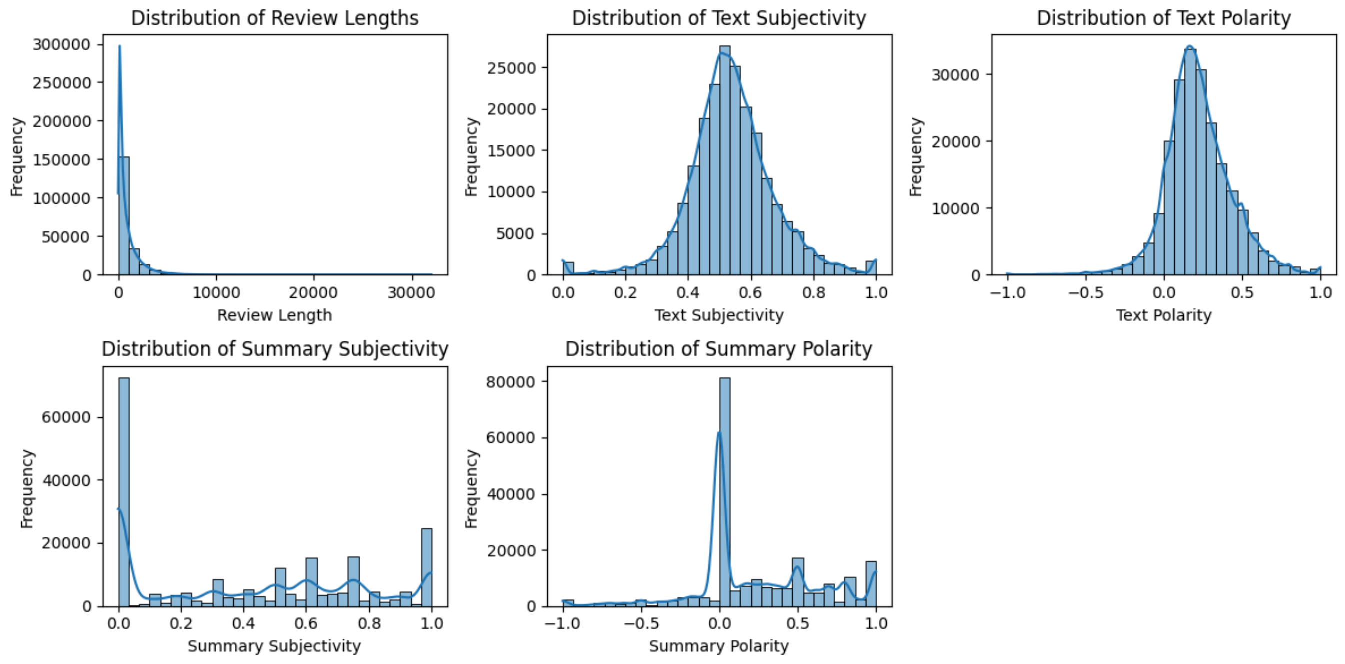
Following the Naive Bayes transformations and other feature engineering steps, I applied Random Forest as the primary classifier. This model leveraged the enriched feature set, allowing it to capture complex patterns within the data while taking advantage of the probabilistic insights from Naive Bayes. By adding diversity to the feature set and reducing overfitting, this combination improved the model's predictive accuracy and generalization capability.

Bibliography-

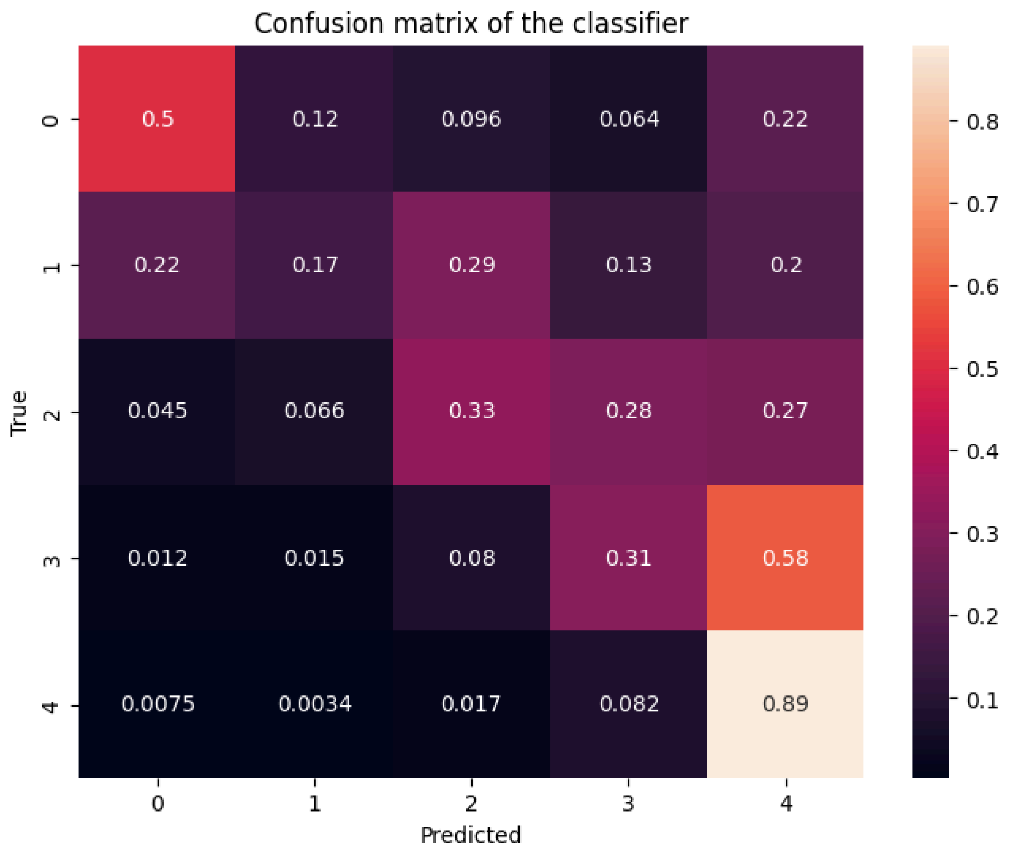
Rashedi, E., Mardani, A., & Zavadskas, E. K. (2018). "Increasing Diversity in Random Forests Using Naive Bayes". *Applied Soft Computing*, 64, 315-327.

Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). "Prediction of Student Performance Using Random Forest Combined With Naive Bayes". *Procedia Computer Science*, 72, 316-322.

Appendix



Distribution of Review Lengths
Review Length
Frequency
Distribution of Text Subjectivity
Text Subjectivity
Frequency
Distribution of Text Polarity
Text Polarity
Frequency
Distribution of Summary Subjectivity
Summary Subjectivity
Frequency
Distribution of Summary Polarity
Summary Polarity
Frequency



Confusion matrix of the classifier
True
Predicted
0.5
0.12
0.096
0.064
0.22
0.22
0.17
0.29
0.13
0.2
0.045
0.066
0.33
0.28
0.27
0.012
0.015
0.08
0.31
0.58
0.0075
0.0034
0.017
0.082
0.89