Model Selection and Hyperparameter Tuning Report

1. Introduction

Sentiment analysis is a crucial task in natural language processing (NLP), and selecting the appropriate model significantly impacts performance. In this project, we experimented with multiple models for text classification and ultimately chose the **Bernoulli Naive Bayes (BNB) classifier** due to its efficiency, interpretability, and suitability for text-based classification tasks.

2. Why Choose Bernoulli Naive Bayes?

The **Bernoulli Naive Bayes (BNB) classifier** is well-suited for binary text classification tasks, where text is represented as presence or absence of words (binary features). Several key reasons influenced our decision:

- **Efficient and Fast:** BNB is computationally efficient and scales well for large datasets, making it a good choice for handling the 160,000 samples in our dataset.
- **Effective for Text Data:** This algorithm assumes features are binary, which aligns well with text data transformed using TF-IDF or CountVectorizer.
- **Robust to High-Dimensional Data:** Since text data usually results in high-dimensional feature spaces, Naive Bayes remains effective because it assumes feature independence, avoiding overfitting.
- **Baseline for NLP Classification:** Many NLP problems, including spam filtering and sentiment analysis, use BNB as a strong baseline model.

3. Hyperparameter Tuning: Alpha Selection

Hyperparameter tuning is essential for optimizing model performance. The primary hyperparameter for **Bernoulli Naive Bayes** is **alpha**, which controls Laplace smoothing to prevent zero probabilities for unseen words.

- We conducted **GridSearchCV** to test different values of alpha: [0.1, 0.5, 1, 1.5, 2].
- The best-performing value was **alpha = 1**, which provided an **accuracy of 80%**.
- This value balances the model by preventing over-smoothing (high alpha) and avoiding zero probabilities (low alpha).

4. Model Performance Evaluation

The **Bernoulli Naive Bayes model (with alpha = 1)** achieved the following results:

Metric	Class 0 (Negative)	Class 4 (Positive)	Macro Avg	Weighted Avg
Precision	0.80	0.80	0.80	0.80
Recall	0.79	0.81	0.80	0.80
F1-score	0.80	0.80	0.80	0.80
Accuracy	-	-	-	0.80

The **precision, recall, and F1-score** are balanced across both classes, indicating that the model does not heavily favor one sentiment over the other.

5. Conclusion

The **Bernoulli Naive Bayes model** was chosen because of its efficiency, simplicity, and effectiveness in handling binary text features. The hyperparameter tuning process determined **alpha = 1** as the optimal value, resulting in an **80% accuracy**. This makes the model suitable for real-world sentiment analysis applications where speed and interpretability are important considerations.

In future improvements, additional techniques such as ensemble learning, feature engineering, or deep learning approaches could be explored to further enhance model performance.

The various steps involved in the **Machine Learning Pipeline** are:

- Import Necessary Dependencies
- Read and Load the Dataset
- Exploratory Data Analysis
- Data Visualization of Target Variables
- Data Preprocessing
- Splitting our data into Train and Test sets.
- Transforming Dataset using TF-IDF Vectorizer
- Function for Model Evaluation
- Model Building and Hyperparameter Tuning
- Model Evaluation