Name: Parsa Mazaheri

Data Science and Machine Learning Fundamentals

---

1 Sentiment Analysis on Stanford Movie Corpus

In this section, we analyze Stanford's movie review corpus to classify reviews as either positive or negative using a range of machine learning models.

1.1 Data Processing

Before feeding the data into our models, it's essential to clean and preprocess the raw text data for better performance. The following preprocessing steps were implemented:
- HTML Tag Removal: removes HTML elements
- URL Removal: Erases web links to keep reviews distraction-free.
- Text Contractions Expansion: Transforms abbreviations to their full forms, ("can't" to "cannot").
- Special Characters and Number Removal: Eradicates non-alphabetical chars.
- Text Lemmatization: Standardizes words (root form); "running" to "run".
- Stopword Removal: removing stopwords such as 'and', 'the', 'is'.
- Single Character Removal: Gets rid of isolated characters that don't convey any sentiment.
- Whitespace Removal: merges consecutive spaces into one.

1.2 Vectorization

To convert our cleaned text data into a format suitable for machine learning, we used the n-gram method (unigram and Bigram) to capture context and semantics. for that I used scikit-learn's `TfidfVectorizer` for this purpose.

1.3 Models

After splitting the dataset into a training set (90%) and a validation set (10%), then we go to train five different machine learning models:

Linear SVM:
Linear Support Vector Machine tries to find the best hyperplane that separates the data into the respective classes. It's widely used for text classification due to its efficiency and ability to handle high dimensional data.

Naive Bayes:
Specifically, we used the Multinomial variant suitable for discrete features like word counts. It's based on Bayes theorem and calculates the probability of each class given a particular feature.

Logistic Regression:
A statistical model that uses the logistic function to model a binary dependent variable. It's a go-to method for binary classification problems like ours.

Random Forest:
An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification. It's known for its high accuracy, ability to handle large datasets with higher dimensionality, and its ability to handle missing values.

Gradient Boosting:
Another ensemble technique that builds multiple decision trees but in a sequential manner where each

tree corrects the errors of the previous one. It often provides higher accuracy compared to other methods but it's computationally expensive.

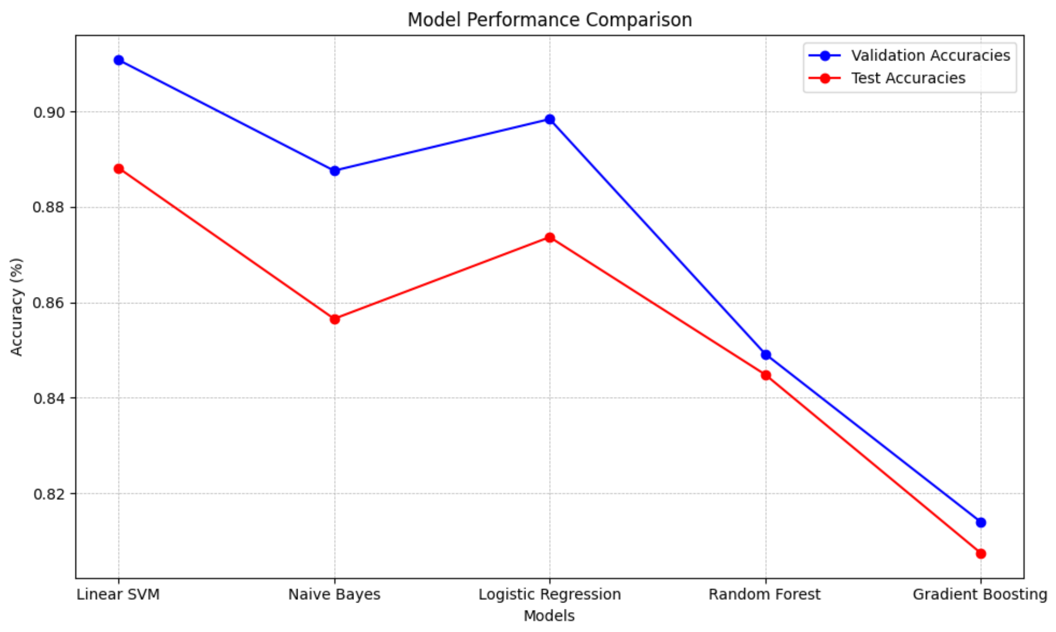
For each model, I trained on the training dataset and evaluated its performance on both validation and test datasets which is shown in the table below.

Model	**Val Acc (%)**	**Test Acc (%)**
Linear SVM	91.08	88.82
Naive Bayes	88.76	85.66
Logistic Regression	89.84	87.37
Random Forest	84.92	84.49
Gradient Boosting	81.40	80.75

1.4 Model Performance Comparison:

Linear SVM notably outperforms other models, achieving a test accuracy of 88.82%. Its efficiency with high-dimensional text data is evident. Logistic Regression follows closely with a test accuracy of 87.37%, demonstrating its reliability for binary classification tasks.

While Naive Bayes posts a decent test accuracy of 85.66%, ensemble methods like Random Forest and Gradient Boosting lag behind with 84.49% and 80.75% respectively. This suggests that for sentiment analysis on this dataset, SVM and Logistic Regression might be more apt choices compared to tree-based ensemble techniques.


Model Performance Comparison
Validation Accuracies
Test Accuracies
Accuracy (%)
Linear SVM
Naive Bayes
Logistic Regression
Random Forest
Gradient Boosting
Models