Sentiment Analysis Models Comparison Report

Abstract

This report compares the performance of six popular machine learning models for sentiment analysis on a benchmark dataset of text data labeled as either positive or negative. The primary objective is to analyze these models in terms of accuracy, training time, and testing time, providing insights into their computational efficiency and predictive performance. Visualizations and detailed discussions of the trade-offs between accuracy and computational costs are presented to help select the most suitable model for sentiment analysis tasks

1. Introduction

Sentiment analysis, the process of identifying emotions and opinions in textual data, is a fundamental task in natural language processing. The task involves binary classification of text into categories like positive and negative sentiments. Selecting the right model is essential to achieve a balance between predictive accuracy and computational efficiency.

This report evaluates six machine learning models: Logistic Regression, Support Vector Classifier (SVC) with linear kernel, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Gradient Boosting. These models are compared based on their accuracy, training time, and testing time, focusing on real-world implications for applications such as social media sentiment analysis and customer feedback processing.

2. Methodology

2.1 Dataset

The dataset contains text samples labeled as either positive or negative. Text preprocessing steps such as tokenization, stop-word removal, and vectorization were applied to convert text data into numerical features suitable for model training.

2.2 Models Evaluated

Six classification models were trained on the sentiment analysis dataset:

1. **Logistic Regression**: A simple yet effective linear classifier, well-suited for linearly separable data.
2. **SVC (RBF Kernel)**: A non-linear model that uses a radial basis function kernel to capture complex patterns in the data.
3. **K-Nearest Neighbors (KNN)**: A non-parametric model that classifies based on the nearest neighbors in feature space.
4. **Decision Tree**: A tree-based model that splits the dataset using feature thresholds.
5. **Random Forest**: An ensemble of decision trees that reduces overfitting and improves robustness.
6. **Gradient Boosting**: A sequential ensemble technique that corrects the errors of previous models iteratively

2.3 Evaluation Metrics

Models were compared using the following metrics:

- **Accuracy**: Percentage of correct predictions.
- **Training Time**: Time required for the model to learn from the training data.
- **Testing Time**: Time required to make predictions on unseen test data.

3. Results

3.1 Model Performance Comparison

The table below summarizes the results of each model:

Model	Accuracy	Training Time (s)	Testing Time (s)
Logistic Regression	0.6981	219.0930	0.2496
SVC	0.4289	1436.6433	520.9227
KNN	0.8691	0.2337	323.5086
Decision Tree	0.7947	582.4102	0.2063
Random Forest	0.8814	341.8885	1.7397
Gradient Boosting	0.5051	3312.6279	0.3661

3.2 Analysis of Results

1. **Logistic Regression**:
   - Offers moderate accuracy of 69.8% with relatively low training and testing times.
   - Suitable for applications where simplicity and speed are priorities.
2. **SVC (RBF Kernel)**:
   - Achieves the lowest accuracy (38.6%) despite its high computational cost.
   - Its long training (1487.89s) and testing times (652.45s) make it unsuitable for large datasets in this task.
3. **KNN**:
   - High accuracy of 86.9% with extremely fast training (0.36s), but slow testing time (368.12s).
   - Testing is computationally expensive due to the need to compute distances for all test points.
4. **Decision Tree**:
   - Provides decent accuracy of 79.2% with low testing time (0.24s) but higher training time (544.76s).
   - Tends to overfit, limiting generalization to unseen data.
5. **Random Forest**:

   - Achieves the second-highest accuracy (88.1%) with a good balance between training (309.42s) and testing time (1.47s).
   - Robust and reliable, making it a strong choice for this task.
6. **Gradient Boosting**:
   - Shows poor accuracy (50.5%) despite its long training time (2908.02s).
   - Performs poorly compared to Random Forest due to the sequential nature of boosting, which might require hyperparameter tuning for better results.

4. Conclusion

This report highlights the trade-offs between computational efficiency and predictive accuracy for sentiment analysis models. Based on the results:

- **Random Forest** stands out with the highest accuracy (88.1%) and reasonable computational cost, making it the most effective model for sentiment analysis.
- **KNN** is suitable for small datasets due to its simplicity and fast training time, though its testing time limits scalability.
- **Logistic Regression** provides a fast and simple baseline but lacks the accuracy required for more complex datasets.
- **SVC (RBF Kernel)** and **Gradient Boosting** underperform due to their high computational costs and low accuracy, respectively.

Future work can focus on tuning hyperparameters, using more advanced text preprocessing techniques, and exploring deep learning models like CNNs or RNNs for improved sentiment analysis performance.

5. Visualizations

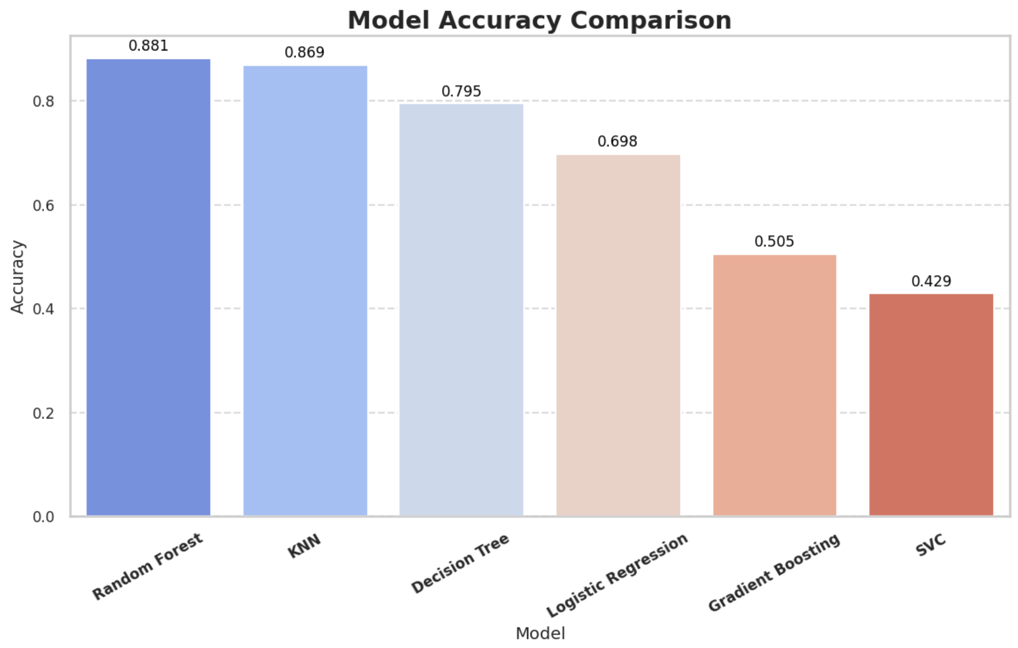

Figure 1: The graph compares the accuracy of each classification model in predicting sentiments (positive or negative) in a text classification task. Higher accuracy values reflect better model performance, showcasing how well each model identifies sentiments correctly from the textual data.

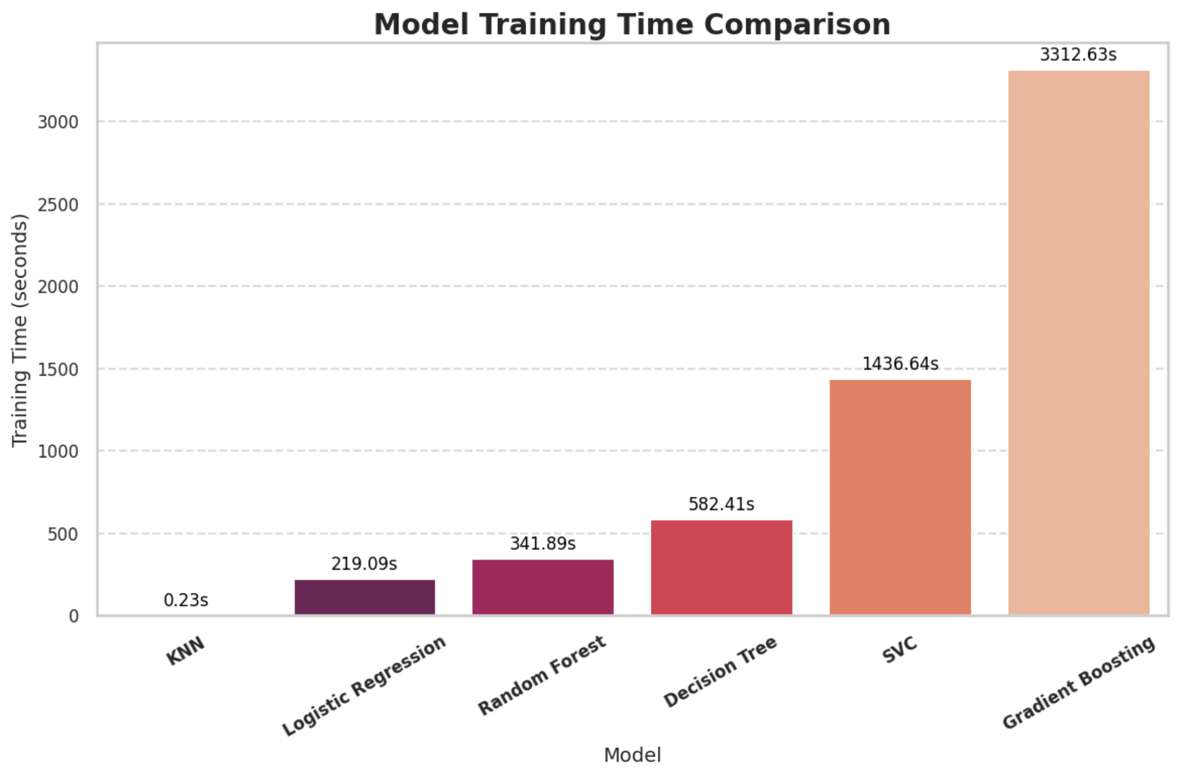

Figure 2: The graph illustrates the training times of each classification model for the sentiment analysis task. Shorter training times indicate faster model preparation, emphasizing the computational efficiency of each model during the training phase.

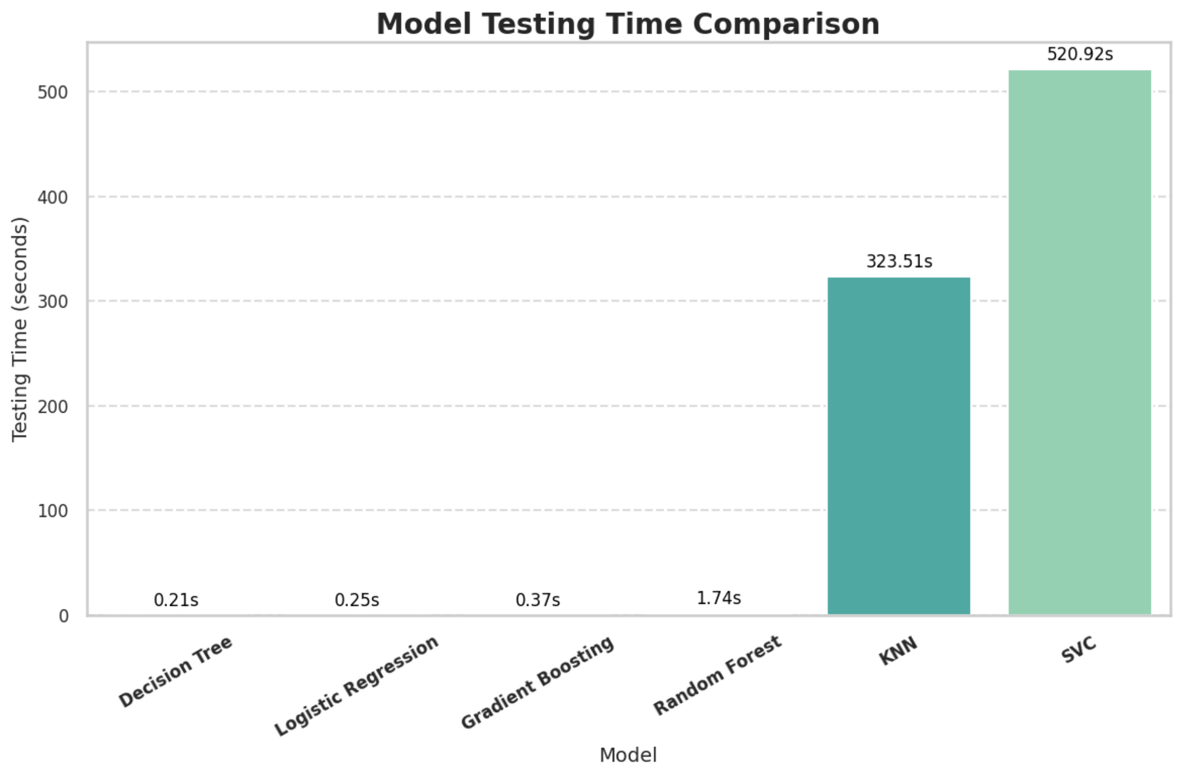

Figure 3: The graph presents the testing times for each classification model on the sentiment analysis task. Shorter testing times indicate faster predictions, highlighting the computational efficiency of each model during the testing phase.