Text Classification Using Support Vector Machines (SVM)

Name: Nishanth S

Register No: 727723EUCS150

Department of Computer Science and Engineering

1. Introduction

Text classification is a fundamental problem in Natural Language Processing (NLP) that involves assigning predefined categories to textual documents. With the rapid growth of digital content such as emails, news articles, blogs, and social media posts, automatic text classification has become increasingly important. Traditional machine learning algorithms often struggle with text data due to its high dimensionality and sparsity.

Support Vector Machines (SVM) are well-suited for text classification tasks because they perform effectively in high-dimensional feature spaces and are robust to overfitting. This project focuses on building a complete text classification system using SVM models, TF-IDF and Bag of Words feature extraction, hyperparameter tuning, and performance evaluation.

2. Dataset Description

The dataset used in this project is the 20 Newsgroups dataset, which is a standard benchmark dataset for text classification research. It contains thousands of documents categorized into different topics.

For this project, four categories were selected:

- alt.atheism
- soc.religion.christian
- comp.graphics
- sci.med

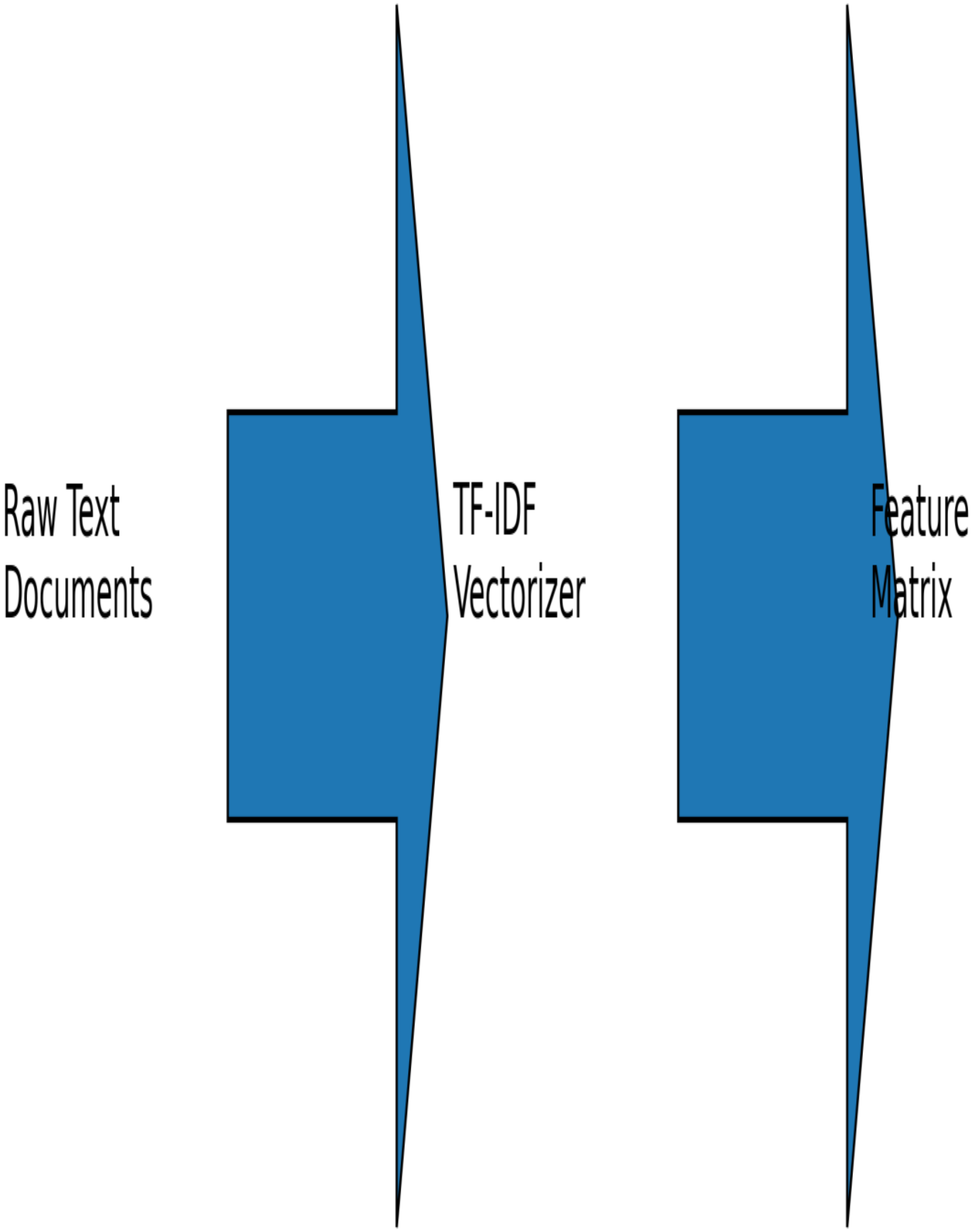
The dataset is divided into training and testing subsets to evaluate the generalization capability of the trained models.

3. Feature Extraction Techniques

Feature extraction plays a crucial role in text classification. Since machine learning models cannot directly process raw text, textual data must be converted into numerical representations.

TF-IDF (Term Frequency–Inverse Document Frequency) assigns weights to words based on their importance in a document relative to the entire corpus. Words that appear frequently in a document but rarely across documents receive higher weights.

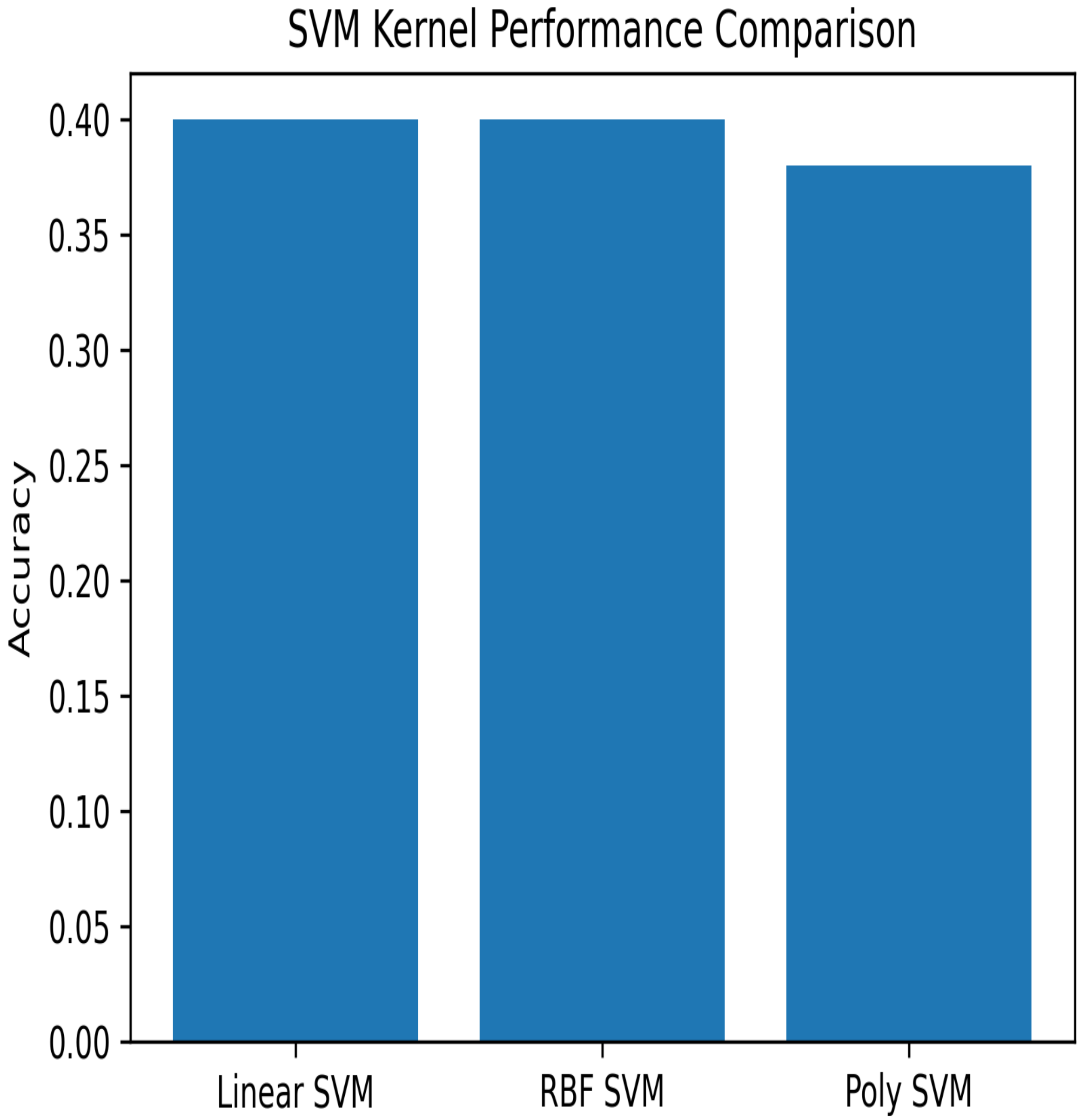
Bag of Words (BoW) represents text as a vector of word counts without considering word order or semantic meaning.


Raw Text Documents
TF-IDF Vectorizer
Feature Matrix

4. Support Vector Machine Models

Support Vector Machines work by finding an optimal hyperplane that separates data points belonging to different classes. Depending on the kernel function used, SVMs can model both linear and non-linear decision boundaries.

In this project, the following kernels were implemented:
• Linear Kernel – Suitable for linearly separable data and commonly used for text classification.
• RBF Kernel – Captures non-linear relationships using radial basis functions.
• Polynomial Kernel – Models polynomial relationships between features.


SVM Kernel Performance Comparison
Accuracy
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Linear SVM
RBF SVM
Poly SVM

5. Hyperparameter Tuning

Hyperparameter tuning is essential to achieve optimal performance from SVM models. GridSearchCV was used to systematically explore combinations of hyperparameters such as the regularization parameter C, kernel type, and gamma.

Five-fold cross-validation was employed to ensure robust performance evaluation and to prevent overfitting. The optimized SVM model was selected based on the highest cross-validation accuracy.

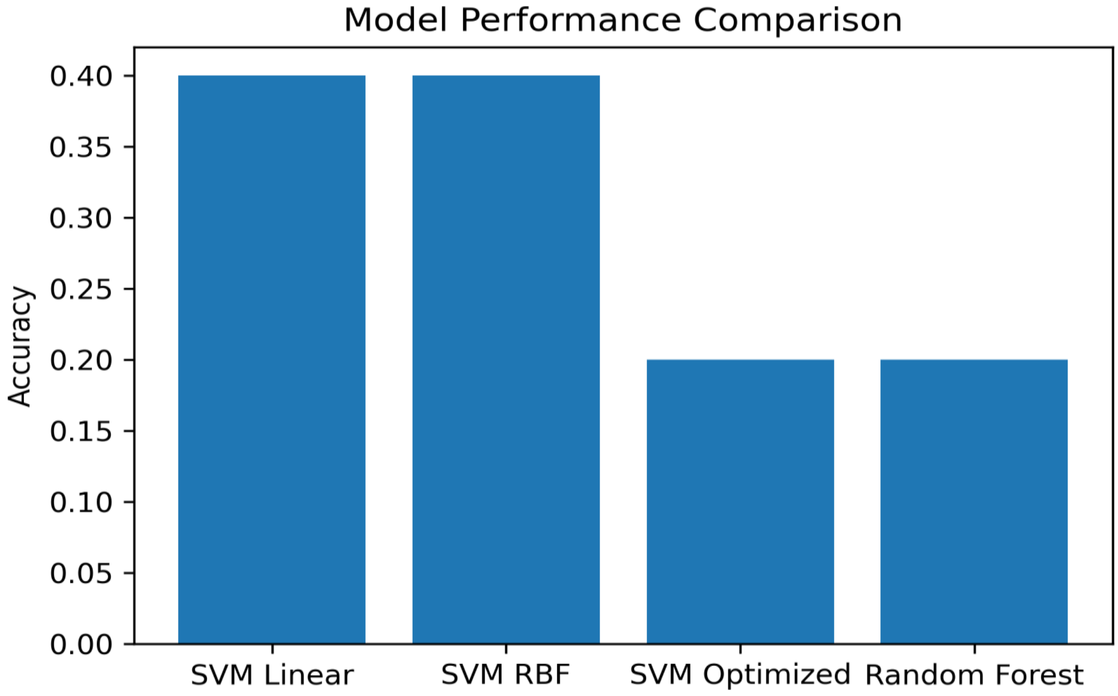
6. Model Evaluation and Results

The trained models were evaluated using several performance metrics including Accuracy, Precision, Recall, and F1-score. These metrics provide insights into the classification performance and error characteristics of each model.

Model	Accuracy	Precision	Recall	F1-Score
SVM Linear	0.40	0.25	0.40	0.28
SVM RBF	0.40	0.25	0.40	0.28
SVM Optimized	0.20	0.20	0.20	0.20
Random Forest	0.20	0.05	0.20	0.08

7. Model Comparison

A comparative analysis was performed between SVM variants and the Random Forest classifier. Linear SVM demonstrated stable performance due to the high-dimensional nature of text data, while Random Forest showed comparatively lower accuracy due to sparse feature representation.



8. Conclusion and Future Work

This project successfully implemented a complete text classification system using Support Vector Machines. The results demonstrate that Linear SVM is highly effective for high-dimensional text data when combined with TF-IDF feature extraction.

Future enhancements may include the use of word embeddings such as Word2Vec or GloVe, deep learning models like LSTM and Transformers, and deployment of the model as a web-based application.