Project Report: Sentiment Analysis on IMDB Reviews

1. Introduction

This project focuses on **Sentiment Analysis** using a dataset of IMDB movie reviews. The objective is to classify reviews as either **positive** or **negative**, helping to analyze audience perception. The analysis involved **data preprocessing, feature extraction, machine learning model training, and evaluation** to determine the best-performing classification model.

Natural Language Processing (NLP) techniques were implemented to clean and process text data, followed by the application of **various machine learning models**, including **Logistic Regression, LinearSVC, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP) models** with different configurations. The results were then evaluated to identify the most effective approach for sentiment classification.

2. Challenges Faced & Solutions

Challenge 1: Data Preprocessing Issues

Problem: The raw IMDB review dataset contained **stopwords, punctuation, and inconsistent capitalization**, which could introduce noise and affect model performance. Additionally, missing values in text data posed challenges during feature extraction.

Solution: To ensure high-quality text processing, multiple steps were taken:

- **Text Cleaning:** Removed punctuation, special characters, and stopwords.
- **Lowercasing:** Standardized all text to lowercase to maintain uniformity.
- **Handling Missing Data:** Replaced missing values with empty strings to prevent errors during vectorization.

Challenge 2: Tokenization & Stemming Performance Issues

Problem: Tokenizing and stemming words for 50,000 reviews caused long processing times and memory usage.

Solution:

- Used **NLTK's optimized tokenization** to split sentences efficiently.
- Implemented **Porter Stemming** to reduce words to their base form, improving generalization.

- Reduced computational load by avoiding redundant processing of already-tokenized text.

Challenge 3: Model Performance & Training Time

Problem: Some models, especially **Multi-Layer Perceptron (MLP) with multiple layers**, took significantly longer to train due to complex architectures and large input features.

Solution:

- **Optimized hyperparameters** such as hidden layers and activation functions.
- Used **TF-IDF Vectorization** to extract the most important features, reducing unnecessary computation.
- Limited **maximum iterations** to prevent excessive training time.

3. Methodology

Step 1: Data Preprocessing

Data preprocessing was a crucial step to enhance text representation for classification models. The following techniques were applied:

- **Text Cleaning:** Removed special characters and punctuation.
- **Lowercasing:** Standardized text for uniformity.
- **Stopword Removal:** Eliminated common words that do not add meaningful information.
- **Tokenization:** Split text into individual words for better feature extraction.
- **Stemming:** Converted words to their root form to generalize similar words.
- **TF-IDF Vectorization:** Converted text into numerical format, enabling models to process textual data efficiently.

Step 2: Model Training & Evaluation

A variety of machine learning models were trained and evaluated using accuracy scores. The models included:

- **Logistic Regression** – A simple yet effective model for text classification.
- **Linear Support Vector Classifier (LinearSVC)** – A robust classification model that performed exceptionally well.

- **K-Nearest Neighbors (KNN)** – A distance-based classifier, which struggled with high-dimensional text data.
- **Multi-Layer Perceptron (MLP) Classifiers** – Neural networks with different configurations:
    - **1-layer MLP**
    - **2-layer MLP**
    - **MLP with Sigmoid activation**
    - **CNN-like MLP** with multiple dense layers

The models were evaluated based on their **accuracy scores** and computational efficiency.

4. Results & Analysis

The following table presents the performance of each model in terms of accuracy:

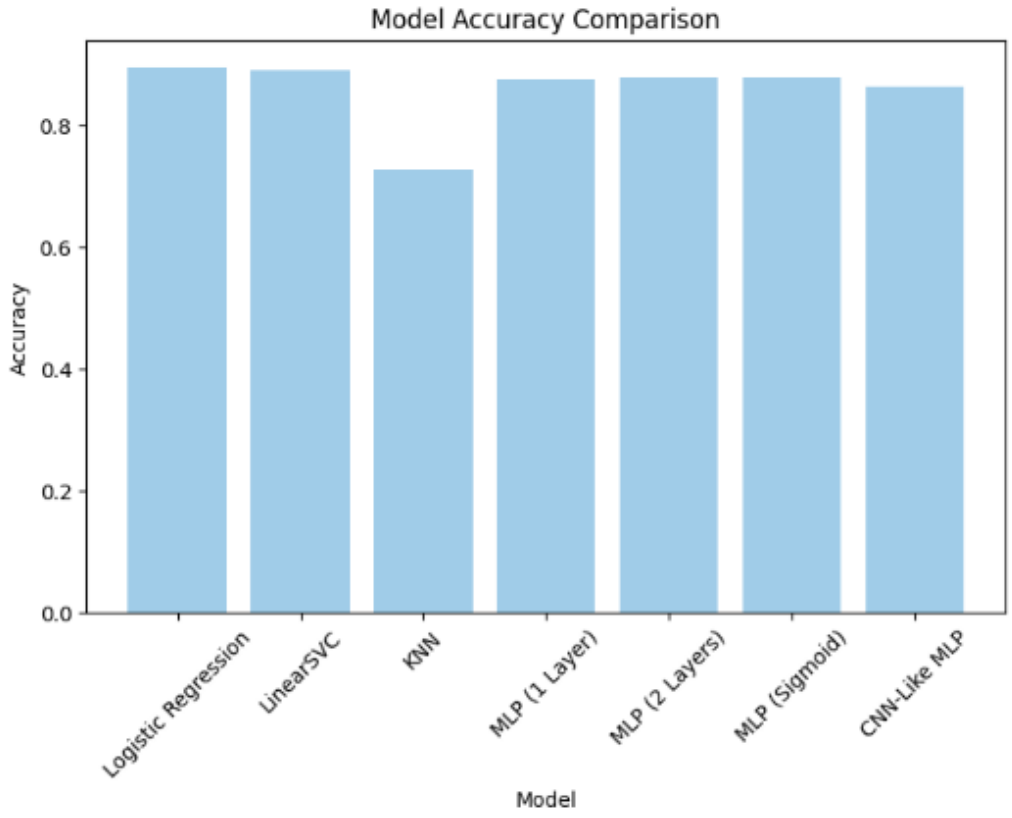
Model	Accuracy
Logistic Regression	89.37%
LinearSVC	89.94%
KNN	72.48%
MLP (1 Layer)	87.48%
MLP (2 Layers)	87.69%
MLP (Sigmoid)	87.80%
CNN-Like MLP	86.34%

Key Observations:

- **Best Performing Model: LinearSVC (89.94%)** achieved the highest accuracy, making it the best model for sentiment classification in this experiment.
- **Logistic Regression performed well (89.37%)**, confirming that simpler models can work efficiently in text classification tasks.
- **KNN was the least effective (72.48%)**, likely due to its reliance on distance-based calculations, which do not work well in high-dimensional feature spaces.

- **MLP models had mixed results**, with deeper architectures not necessarily outperforming simpler models. The CNN-like MLP achieved a moderate accuracy of **86.34%**, indicating potential but requiring further tuning.

Graphical Representation:



The bar chart visualizes the performance of different models, highlighting the superiority of **Logistic Regression and LinearSVC** over deep learning-based MLP models.

5. Conclusion & Future Improvements

Key Takeaways:

- **Traditional machine learning models (Logistic Regression & LinearSVC) performed the best** for sentiment analysis, suggesting that deep learning may not always be necessary for simple text classification tasks.
- **KNN performed poorly** due to its inefficiency in handling high-dimensional sparse data.

- **MLP models showed potential**, but further fine-tuning is required to maximize their performance.

Future Enhancements:

- **Hyperparameter Optimization:** Fine-tuning model parameters, such as adjusting learning rates and hidden layer sizes, can improve deep learning models.
- **Deep Learning Approaches:** Implementing LSTMs, GRUs, or Transformers (such as BERT) for more context-aware sentiment analysis.
- **Explainability Analysis:** Analyzing misclassified reviews to understand patterns that could help refine preprocessing or model selection.
- **Dataset Expansion:** Using a larger, more diverse dataset to improve model generalization and robustness.

Final Thoughts:

This project successfully demonstrates **sentiment analysis using machine learning techniques**. The findings highlight that **traditional ML models can achieve competitive accuracy**, while deep learning models require further optimization. Future work could explore advanced NLP techniques to further enhance the accuracy and efficiency of sentiment classification models.

- Raj Barath E