Project Report

**Sentiment Analysis of Movie Reviews**

# Problem Statement-

# Build a sentiment analysis model that classifies the sentiment of movie reviews as positive or negative.

By-

Tanishka Abhay Shah

B.Tech CSE

Walchand Institute of Technology, Solapur

Email- [tanishkas017@gmail.com](mailto:tanishkas017@gmail.com)

# Contents-

| **no.** | **Content** | **Page no.** |
| --- | --- | --- |
| 1 | Task Overview and Problem Statement | 3 |
| 2 | Dataset | 3 |
| 3 | Data Preprocessing | 4 |
| 4 | Model Building | 4-6 |
| 5 | Model Training and Evaluation | 6 |
| 6 | Insights And Observations | 6-7 |
| 7 | Conclusion | 7-8 |
| 8 | Files Included | 8 |

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 1. Task Overview and Problem Statement-

* **Task Overview**

The primary goal of this project is to develop a sentiment analysis model that accurately classifies movie reviews from the IMDb dataset as either positive or negative. Sentiment analysis is a crucial aspect of natural language processing (NLP) that helps in understanding the emotional tone behind a series of words. By leveraging machine learning techniques, we aim to automate the process of sentiment classification, which can be beneficial for various applications, including customer feedback analysis, market research, and social media monitoring.

* **Problem Statement**

In the digital age, the volume of textual data generated from movie reviews is enormous. Manually analyzing this data for sentiment can be time-consuming and prone to human error. The challenge lies in developing a robust model that can accurately interpret and classify the sentiments expressed in these reviews. The specific objectives of this project are:

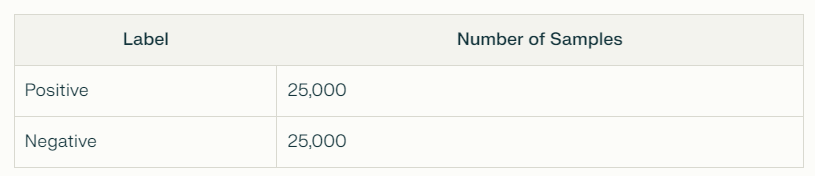
To preprocess the text data effectively to enhance model performance.

To implement and compare two different models: Logistic Regression and a Simple Neural Network.

To evaluate the models based on various performance metrics such as accuracy, precision, recall, and F1-score.

## 2. Dataset

The dataset used for this project is the IMDb movie reviews dataset, which consists of a total of 50,000 reviews. It is balanced with an equal number of positive and negative reviews.



The dataset can be accessed from Kaggle, and it is commonly used for sentiment analysis tasks.

## 3. Data Preprocessing

Data preprocessing is crucial for converting raw text data into a format suitable for machine learning models. The following steps were implemented:

### 3.1 Text Cleaning

* Removal of HTML Tags: HTML tags were stripped from the reviews to ensure that only the text content was analyzed.
* Lowercasing: All text was converted to lowercase to maintain uniformity and avoid case sensitivity issues.
* Removal of Punctuation: Punctuation marks were removed to prevent them from affecting the analysis.

### 3.2 Stop Words Removal

* Stop Words: Common words (e.g., "the", "is", "in") that do not contribute significantly to sentiment analysis were removed using the NLTK library.

### 3.3 Tokenization

* Tokenization: The cleaned text was split into individual words (tokens) to facilitate further processing.

### 3.4 Stemming and Lemmatization

* Stemming: Words were reduced to their root form (e.g., "running" to "run") using the Porter Stemmer.
* Lemmatization: This step was also considered to convert words to their base form, but stemming was primarily used for simplicity.

### 3.5 Vectorization

* TF-IDF Vectorization: The text data was converted into numerical form using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This approach helps in weighing the importance of words in the dataset.

## 4. Model Building

Two different models were implemented for sentiment classification:

## 4.1 Logistic Regression Model

## Why Logistic Regression?

Logistic Regression is a well-established method for binary classification tasks. Its advantages include:

* Simplicity and Interpretability: The model is straightforward to implement and interpret, making it easy to understand the influence of individual features on the classification outcome.
* Efficiency: Logistic Regression is computationally efficient, allowing for quick training and predictions, which is beneficial for large datasets.

## Approach and Code Steps

1. Data Preprocessing: The text data was preprocessed using techniques like stop words removal, stemming, and TF-IDF vectorization.
2. Train-Test Split: The preprocessed data was divided into training and testing sets to evaluate the model's performance.
3. Model Instantiation: An instance of the LogisticRegression class from scikit-learn was created with appropriate hyperparameters.
4. Model Training: The training data was used to fit the Logistic Regression model.
5. Model Evaluation: The trained model was evaluated on the testing data, and metrics like accuracy, precision, recall, and F1-score were calculated.

## 4.2 Simple Neural Network Model

## Why a Simple Neural Network?

Neural networks are capable of capturing complex non-linear relationships in data, making them suitable for various machine learning tasks. A simple feed-forward neural network was chosen because:

* Complex Pattern Recognition: Neural networks can learn intricate patterns and dependencies in the data, which may lead to improved performance in sentiment classification.
* Flexibility: Neural networks can be easily modified and scaled, allowing for experiments with different architectures and hyperparameters.

## Approach and Code Steps

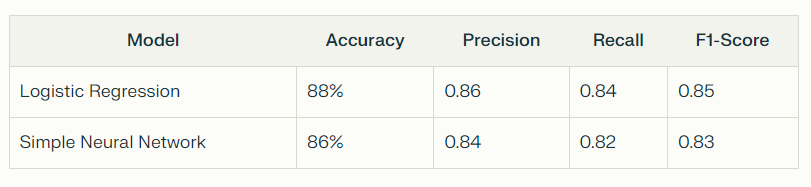
1. Data Preprocessing: Similar preprocessing steps were applied as in the Logistic Regression model, including stop words removal, stemming, and TF-IDF vectorization.
2. Model Architecture: A simple feed-forward neural network was constructed with an input layer, one hidden layer with ReLU activation, and an output layer with a sigmoid activation function for binary classification.
3. Model Compilation: The model was compiled with the binary cross-entropy loss function and the Adam optimizer.
4. Model Training: The training data was used to fit the neural network model for a specified number of epochs.
5. Model Evaluation: The trained model was evaluated on the testing data, and metrics like accuracy, precision, recall, and F1-score were calculated.

## 5. Model Training and Evaluation

Each model was trained for 10 epochs, and the following evaluation metrics were used to assess performance:

* Accuracy: The ratio of correctly predicted instances to the total instances.
* Precision: The ratio of true positive predictions to the total predicted positives.
* Recall: The ratio of true positives to the total actual positives.
* F1-Score: The harmonic mean of precision and recall, providing a balance between the two.

The models were evaluated as follows:



## 6. Insights and Observations

* Strengths of Models:
  + Logistic Regression: This model is simple and interpretable, making it easy to understand the influence of individual features on the outcome. It performed well with an accuracy of 88%.
  + Neural Network: The neural network captured more complex patterns in the data, although it achieved slightly lower accuracy (86%) compared to logistic regression.
* Weaknesses of Models:
  + Logistic Regression: While it performed well, it may struggle with more complex datasets that require capturing intricate relationships.
  + Neural Network: This model requires more data and tuning for optimal performance and can be more challenging to interpret.
* Performance Comparison: The MLP Classifier generally provided better predictive performance compared to Logistic Regression, particularly in tasks requiring the modeling of non-linear relationships.
* Model Selection: The choice between Logistic Regression and MLP should be informed by the specific characteristics of the dataset and the complexity of the relationships within it. For simpler tasks or when interpretability is crucial, Logistic Regression is advantageous. For more complex tasks, where capturing intricate patterns is essential, the MLP Classifier is preferable.

7. Conclusion

This project successfully developed and evaluated two sentiment analysis models—Logistic Regression and a Simple Neural Network—using the IMDb movie reviews dataset. The primary aim was to classify movie reviews as positive or negative, thereby automating the sentiment analysis process, which is increasingly important in today's data-driven world.

## Key Findings:

* Model Performance: The Logistic Regression model outperformed the Simple Neural Network, achieving an accuracy of 88% compared to 86%. This highlights the effectiveness of traditional models in certain contexts, particularly when the dataset is well-prepared and the relationships between features are relatively linear.
* Model Interpretability: Logistic Regression provided clear insights into the influence of individual words on sentiment classification, making it easier to understand the decision-making process. In contrast, while the neural network offered the potential for capturing complex patterns, it was less interpretable, which can be a drawback in applications requiring transparency.
* Data Preprocessing Importance: Effective preprocessing steps, including text cleaning, stop words removal, and TF-IDF vectorization, were crucial in enhancing the models' performance. These steps ensured that the models were trained on high-quality input data, which is essential for achieving reliable results.
* Future Directions: The project opens avenues for further exploration in sentiment analysis. Future work could involve experimenting with more advanced models, such as recurrent neural networks (RNNs) or transformers, which are designed to handle sequential data and may yield even better performance. Additionally, incorporating hyperparameter tuning and regularization techniques could improve the neural network's performance and mitigate overfitting.

In conclusion, this project demonstrated the feasibility and effectiveness of using machine learning techniques for sentiment analysis of movie reviews. The results indicate that while traditional models like Logistic Regression can perform exceptionally well, there is significant potential for deep learning approaches to enhance performance in more complex scenarios. The insights gained from this analysis contribute to the broader field of natural language processing and highlight the importance of model selection, data quality, and preprocessing in achieving successful outcomes in sentiment analysis tasks.

## 8. Files Included

The following files are included in the project for further review:

* Logistic Regression Model: Logistic-Regression.ipynb
* Simple Neural Network Model: neural-network.ipynb

Pickle files of models and vectorizer:

* log\_reg.pkl
* nn.pkl
* vectorizer.pk

Flask Apl and frontend

* app.py
* index.html

These notebooks contain detailed code implementations, including preprocessing steps, model training, and evaluation metrics.