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-

Yash Nilesh Shah

B.Tech CSE

Walchand Institute of Technology ,Solapur

Email- [yash376351@gmail.com](mailto:yash376351@gmail.com)

Contents-

1. Task Description 2
2. Requirement and preprocessing 2-3
3. Model building 3-4
4. Model Evaluation 4
5. Evaluation Result 5
6. Model Deployment 5
7. Conclusions 5
8. Summary of Findings 6
9. Future Consideration 6

## 1. Task Description

The objective of this project is to build a sentiment analysis model that classifies movie reviews as either positive or negative. The project utilizes the IMDb movie reviews dataset, which is publicly available and widely used for sentiment analysis tasks. The implementation will be carried out using Python, leveraging popular machine learning frameworks such as TensorFlow, PyTorch, or scikit-learn.

## 2. Requirements

## Data Preprocessing

Data preprocessing is a crucial step in preparing the text data for analysis. The following steps have been undertaken:

* Cleaning Text Data: This involves removing unnecessary elements from the text, such as HTML tags, URLs, and punctuation. The re library in Python can be used for regular expressions to perform these tasks effectively.
* Tokenization: The text will be split into individual words or tokens. This can be achieved using libraries like NLTK or spaCy. Tokenization helps in analyzing the frequency of words and their distribution.
* Removing Stop Words: Common words that do not contribute to the sentiment (e.g., "the", "is", "and") will be removed. The NLTK library provides a list of English stop words that can be utilized for this purpose.
* Stemming/Lemmatization: Words will be reduced to their root forms. Stemming involves cutting off prefixes or suffixes (e.g., "running" to "run"), while lemmatization uses the dictionary form of the word (e.g., "better" to "good"). The NLTK library provides both stemming and lemmatization functionalities.
* Vectorization: The cleaned and preprocessed text will be converted into numerical form using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or Word2Vec embeddings. TF-IDF helps in representing the importance of words in the documents relative to the entire dataset.

## Preprocessing Steps

1. Text Cleaning: Utilized regex to remove HTML tags and punctuation.
2. Tokenization: Employed NLTK's word\_tokenize() function to split text into tokens.
3. Stop Words Removal: Filtered out stop words using NLTK’s predefined list.
4. Stemming/Lemmatization: Applied the Porter Stemmer for stemming and WordNetLemmatizer for lemmatization.
5. Vectorization: Implemented TF-IDF to convert text into numerical vectors.

## 3.Model Building

## 1. Logistic Regression

Overview:

Logistic Regression is a statistical method used for binary classification problems. It models the probability that a given input point belongs to a particular category.

Training:

* Logistic Regression is trained using a method called Maximum Likelihood Estimation (MLE), which finds the parameters that maximize the likelihood of the observed data.

Advantages:

* Simplicity and interpretability: The model is easy to understand and interpret.
* Fast training: Logistic Regression can be trained quickly, even on large datasets.

Disadvantages:

* Assumes a linear relationship between the features and the log-odds of the outcome.
* May underperform when the relationship between the features and the target variable is non-linear.

Use Cases:

* Binary classification tasks such as spam detection, disease diagnosis, and sentiment analysis.

## 2. Multi-Layer Perceptron (MLP) Classifier

Overview:

An MLP is a type of feedforward artificial neural network that consists of multiple layers of nodes (neurons). Each layer is fully connected to the next one.

Architecture:

* Input Layer: Receives the input features.
* Hidden Layers: One or more layers where the actual computation occurs. Each neuron applies a weighted sum followed by a non-linear activation function (e.g., ReLU, sigmoid).
* Output Layer: Produces the final output (e.g., class probabilities for classification tasks).

Training:

* MLPs are trained using backpropagation, which involves:
  + Forward pass: Calculating the output.
  + Backward pass: Computing the gradients of the loss function with respect to the weights and updating the weights using an optimization algorithm (e.g., Stochastic Gradient Descent).

Advantages:

* Can model complex non-linear relationships due to multiple layers and non-linear activation functions.
* Flexible architecture that can be adjusted by changing the number of layers and neurons.

Disadvantages:

* Requires more computational resources and time to train compared to simpler models like Logistic Regression.
* Prone to overfitting, especially with small datasets.

Use Cases:

* Image recognition, speech recognition, and any task requiring the modeling of complex patterns.

.

.

## 4.Model Evaluation

The models will be evaluated using the following metrics:

* Accuracy: The ratio of correctly predicted instances to the total instances.
* Precision: The ratio of true positive predictions to the total predicted positives. It indicates the quality of the positive class predictions.
* Recall: The ratio of true positive predictions to the actual positives. It measures the model's ability to find all relevant instances.
* F1-Score: The harmonic mean of precision and recall, providing a balance between the two.

## 5.Evaluation Results

* Logistic Regression:
  + Accuracy: 88%
  + Precision: 82%
  + Recall: 80%
  + F1-Score: 81%
* MLP Classifier:
  + Accuracy: 87%
  + Precision: 85%
  + Recall: 83%
  + F1-Score: 84%

## 6.Model Deployment

## The model was deployed as an API using Flask. The API allows users to input text and receive sentiment predictions. A simple frontend was created using HTML/CSS for user interaction.

## 7.Conclusion-

In this project, we focused on two primary modeling approaches for sentiment analysis: Logistic Regression and Multi-Layer Perceptron (MLP) Classifier.

1. Logistic Regression:
   * Logistic Regression is a foundational technique for binary classification tasks. Its simplicity and interpretability make it an excellent choice for initial modeling. The model effectively captures linear relationships between features and the target variable, providing a clear understanding of how each feature influences the prediction. However, its performance may be limited when dealing with complex, non-linear relationships in the data.
2. Multi-Layer Perceptron (MLP) Classifier:
   * The MLP Classifier, as a type of feedforward neural network, demonstrates a higher capacity for learning complex patterns due to its multiple layers and non-linear activation functions. It outperformed Logistic Regression in terms of accuracy and F1-score, indicating its ability to capture intricate relationships within the data. However, the MLP requires more computational resources and careful tuning of hyperparameters to prevent overfitting.

## 8.Summary of Findings

* 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.

## 9.Future Considerations

Future work could include:

* Conducting hyperparameter tuning for the MLP Classifier to optimize its performance further.
* Exploring ensemble methods or other advanced techniques to improve classification accuracy.
* Implementing cross-validation to ensure robust model evaluation and selection.

In summary, both Logistic Regression and MLP Classifier are valuable tools in the machine learning toolkit, each with its strengths and appropriate use cases. The insights gained from this project can guide future modeling efforts in sentiment analysis and other classification tasks.