**Title of thesis: Sentiment Analysis on Movie Reviews**

**Task for Course: DLBAIPNLP01 – Project: NLP**

**NLP PROJECT**

**Date:**

**Author’s name:**

**Matriculation number:**

**Tutor’s name:**

**Table of Contents**

[1. Introduction 7](#_Toc187943638)

[1.1 Context 7](#_Toc187943639)

[1.2 Objectives of the project 7](#_Toc187943640)

[1.3 Methods 7](#_Toc187943641)

[1.4 Experimental Set-Up 8](#_Toc187943642)

[1.5 Structure of report 8](#_Toc187943643)

[2. Main Body: Implementation, Evaluation, and Reflection 9](#_Toc187943644)

[2.1 Implementation 9](#_Toc187943645)

[2.1.1 Phase planning 9](#_Toc187943646)

[2.1.2 Data Collection 10](#_Toc187943647)

[2.1.3 Data Preprocessing 11](#_Toc187943648)

[2.1.4 Model Selection and Training 12](#_Toc187943649)

[2.2 Evaluation 13](#_Toc187943650)

[2.2.1 Model evaluation using the test dataset 13](#_Toc187943651)

[2.2.2 Movie Review Analysis 14](#_Toc187943652)

[2.2.3 Discussion 15](#_Toc187943653)

[2.3 Reflection 15](#_Toc187943654)

[2.3.1 Project performance 15](#_Toc187943655)

[2.3.2 Resources 16](#_Toc187943656)

[3. Conclusion: Project Evaluation and Anchoring 16](#_Toc187943657)

[3.1 Conclusion 16](#_Toc187943658)

[3.2 Future Work 16](#_Toc187943659)

[4. Bibliography 18](#_Toc187943660)

[5. List of appendices 20](#_Toc187943661)

[6. Appendices and materials 24](#_Toc187943662)

**List of figures**

[Figure 1: Datasets 9](#_Toc187944202)

[Figure 2: Initialising lemmatizer 9](#_Toc187944203)

[Figure 3: Pre-processing of all the datasets 10](#_Toc187944204)

[Figure 4: Label encoding of all the datasets 10](#_Toc187944205)

[Figure 5: Training using the Logistic Regression Model 11](#_Toc187944206)

[Figure 6: Evaluation of model using testing datasets 11](#_Toc187944207)

[Figure 7: Performance matrix for testing dataset 12](#_Toc187944208)

[Figure 8: Generalisation Performance of all datasets 12](#_Toc187944209)

[Figure 9: Analysis of movie reviews 13](#_Toc187944210)

List of Tables

[Table 1: Process planning and scheduling phase 8](#_Toc187947026)

**List of abbreviations**

**NLTK:** Natural Language Toolkit

**TF-IDF:** Term Frequency-Inverse Document Frequency

**LR:** Logistic Regression

**ML**: Machine Learning

**HPT:** Hyper Parameters Tuning

**1. Introduction**

## 1.1 Context

Sentiment Analysis is a subfield of natural language processing whose primary objective is to classify one’s sentiments or opinions about a certain product or event in a given textual data. According to the sphere of movie reviews, it helps to sort opinions into positive, and negative regarding understanding people’s attitudes. The audience relies on movie reviews to make decisions and Movie reviews strategically impact the market plans (Onalaja *et al.,* 2021). The amount of new content shared by users on websites is impressive as sentiment analysis plays a crucial role in evaluating viewers’ opinions and helps filmmakers, marketers, studios, and others to make the right decision based on a viewer’s sentiment.

## 1.2 Objectives of the project

* To classify movie reviews based on sentiment analysis which determines whether the sentiment is positive or negative.
* To train and build a single supervised machine learning model that provides better identification of sentiments.
* To test the quality of the model on different datasets predicting the sentiment and checking the outcomes.
* To use the developed model for carrying out sentiment analysis of movie reviews.

## 1.3 Methods

Different steps in the methodology applied to this project of sentiment analysis, the first of these including the acquisition of a trustworthy movie review dataset. It becomes important to have a good set of data that should be used to train the sentiment analysis model. Several text preprocessing methods are applied as the next step to shaping the dataset for analysis (Hamzah, 2021). After data collection, the text is preprocessed for cleaning and normalisation for which tokenisation is performed, removing the stop words and applying them to the stem (Gadekallu *et al.,* 2022). Preprocessing splits the text data and preprocesses it by normalising, removing stop words, stemming, and tokenization the text for further machine learning. After processing the text is analysed using logistic regression to determine the text sentiment as either positive or negative (Ullah *et al.,* 2022). The presented models are inspected according to the regularly used performance indicators including accuracy, precision, recall, and F1-score respectively. Cross-validation is used to check how accurate the models perform on unseen data, thus decreasing the impact of overfitting and increasing the reliability of the models (Dashtipour *et al.,* 2021).

## 1.4 Experimental Set-Up

The experiment is based on Python with employed libraries Scikit-learn, NLTK, and TensorFlow to solve tasks connected with machine learning and NLP. The initial step taken is data preprocessing using a digitally available Natural Language Toolkit(NLTK) for tokenising cleaning and stemming the text data. After feature extraction, where the data is transformed into quantitative formats from text forms using techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) (Sulthana *et al.,* 2022). A movie review data set is randomly divided into training and testing data sets so that the performance of the models can be tested on data other than the training data. The logistic regression model's classification is done using TensorFlow and Scikit-learn respectively. Cross-validation is used to check the generalisation of the models confirming their stability when the data is different (Steinke *et al.*, 2022). Also, hyperparameter tuning is used to increase the model’s dimension and distinguish accuracy to enhance the precision of sentiment classification.

## 1.5 Structure of report

The report consists of several structures for the sentiment analysis project for better understanding, the report is arranged systematically. The sections of the reports include:

***Introduction: Project Objectives and Preparations***

The phase starts with an explanation of the aims of the project and the rationale that defines the need for the developed sentiment analysis of movie reviews, the main methodology, and preliminary steps. It formulates what the problem is and what has been used in creating a model.

***Main Body: Implementation, Evaluation, and Reflection***

The Implementation phase section is further divided into data collection, data preprocessing, model training & evaluation sections. It also provides a focus on the types of machine learning models and their prediction performance.

***Conclusion: Project Evaluation and Anchoring***

Finally, the Conclusion presents the overall analysis of results, opinions on the success of the project, and pointers for future enhancements of the proposed sentiment analysis model.

The arrangements of reports are logically arranged inclusively in the introduction part of the project’s objectives and methods (Abimanyu *et al.*, 2023).

# **2. Main Body: Implementation, Evaluation, and Reflection**

## 2.1 Implementation

### 2.1.1 Phase planning

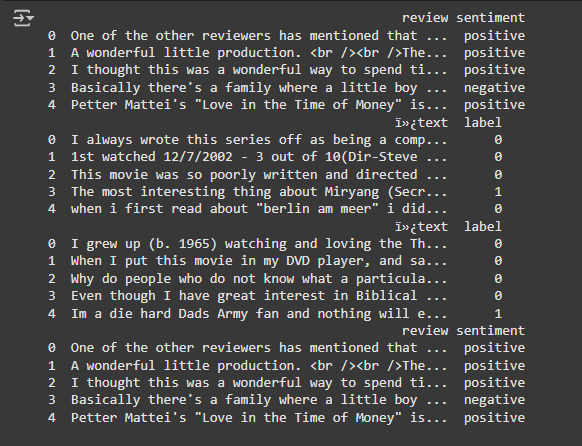
The project is carried out in various phases that can be categorised as the planning phase and the implementation phase. It started with collecting movie review datasets and then using data preprocessing. Various types of machine learning models developed and trained including Logistic Regression (LR) used pre-processed data (Shaddeli *et al.*, 2022). The models are then assessed for performance using the chosen performance metrics and required hyperparameter adjustments to be made for better outcomes.

###### Table 1: Process planning and scheduling phase

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Phase** | **Stage 1** | **Stage 2** | **Stage 3** | **Stage 4** | **Stage 5** | **Stage 6** | **Stage 7** | **Stage 8** |
| Project Initiation |  |  |  |  |  |  |  |  |
| Data Collection |  |  |  |  |  |  |  |  |
| Data Preprocessing |  |  |  |  |  |  |  |  |
| Model Development |  |  |  |  |  |  |  |  |
| Model Evaluation |  |  |  |  |  |  |  |  |
| Hyper parameter tuning |  |  |  |  |  |  |  |  |
| Result Analysis |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |

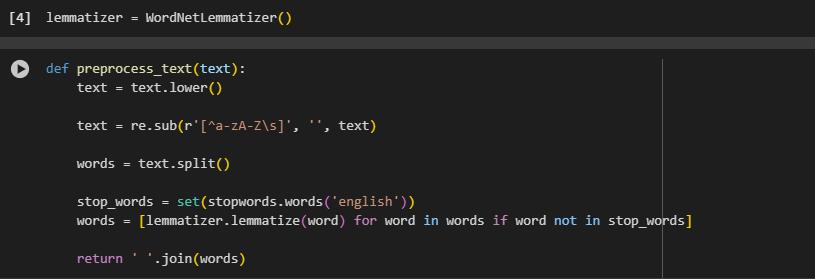
The project has nine phases that include Project Initiation in which goals, project objectives, and available resources are stipulated. Even in Data Collection, movie review datasets are collected and in Data Preprocessing, the collected datasets are cleaned and transformed. Model Development works with the logistic regression, and Model Evaluation then evaluates the performance. Hyper Parameters Tuning (HPT) to increase the accuracy of the model's tuning is done. Result Analysis compares the results produced by the model and Report Writing serves as its documentation process.

### 2.1.2 Data Collection



##### Figure 1: Datasets

The above figure shows a feature vector of a dataset consisting of “*IMDB Movies reviews”* for the training dataset and *“IMDB datasets (Sentiment Analysis) in CSV format, “IMDB Movie Ratings Sentiment Analysis”* and *“IMDB Dataset of 50K Movies Reviews movie reviews”* for testing purposes respectively. It seems that this data is in the form of tabular form with rows containing fields of reviews while columns contain the fields review text, sentiment, and text label.

****

##### Figure 2: Initialising lemmatize

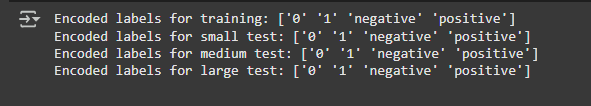
The above figure depicts a function named preprocess\_text, which is used to clean a given string of text to fit it into data analysis purposes. It converts the text to lowercase, removes unwanted characters from the text and other special characters, splits the text into an array of words, removes the stop words, and emulates the words. It then returned preprocessed text as a string with joined words separated by space.

### 2.1.3 Data Preprocessing

|  |  |
| --- | --- |
|  |  |
|  |  |

##### Figure 3: Pre-processing of all the datasets

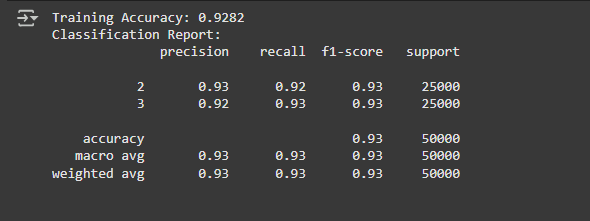
Figure 3 represents contrast between original and cleaned reviews The second figure shows how text is transformed for clarity. This figure depicts the techniques that are used to prepare text data for improved analysis of the data. This figure allows comparing the first impression map with the cleaned reviews map, and it demonstrates how impressions are normalised. The third figure illustrates a transition from narrative accounts to data with a preordained structure. The last figure illustrates the normalised form of reviews for comparison and uniformity.

****

##### Figure 4: Label encoding of all the datasets

The above figure displays encoded labels regarding various datasets, which can be applied to a machine-learning problem in the case of sentiment analysis. The labels are assigned integer numbers, zeros, ones, and string labels which are “negative” and “positive” respectively. This encoding is widely used in the machine learning context to handle categorical features, where a particular number can be used by the model.

### 2.1.4 Model Selection and Training

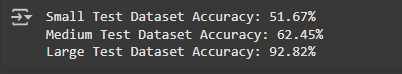


##### Figure 5: Training using the Logistic Regression Model

The above figure shows the logistic regression (LR) model's overall effectiveness on the training data is visualised by the accuracy of 0.9282. The report gives further details of how the model performed on each of classes 2 and 3 respectively. The overall accuracy of the model is 93% which suggests that the logistic regression performs well in the prediction of sentiment analysis.

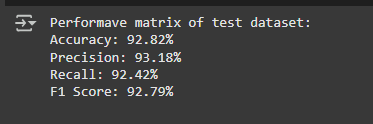
## 2.2 Evaluation

### 2.2.1 Model evaluation using the test dataset



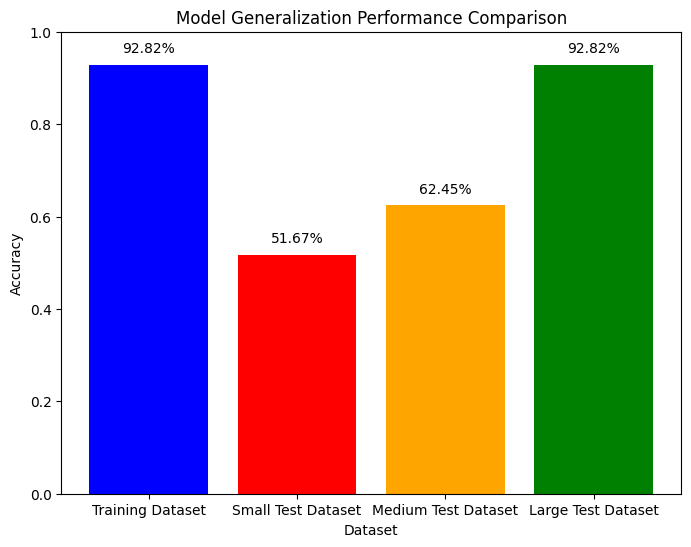
##### Figure 6: Evaluation of model using testing datasets

Figure 6 represents the accuracy scores of a machine learning model on three different test datasets involving small, medium, and large respectively. The accuracy obtained on the small test dataset is 51.67%, which shows that the performance of every implemented classifier is low. On the medium test dataset, the accuracy increases to 62.45% respectively. A large improvement is seen on the large dataset in which the model has an accuracy of 92.82% showing a better performance increase with the increased test size.



##### Figure 7: Performance matrix for testing dataset

Figure 7 illustrates the evaluation of the model with 92.82% accuracy for instance classification. Other accuracy defined as precision, which is the ratio of the successful true positive predictions to the overall number of positive predictions, stands at 93.18%. The measure of true positivity among all positive predictions is 92.42%, F-score is 92.79%.



##### Figure 8: Generalisation Performance of all datasets

The above figure shows the bar chart which reveals various features of the Logistic Regression model and datasets. On the training dataset it reaches 92.82% thus is very effective on the data which it undergoes training. On a small test dataset, accuracy is about 51.67%, which signals overfitting. The results of the medium test data reach 62.45% and the large test data’s classification reaches 92.82%, explaining that enough data will allow for good generalisation ability.

### 2.2.2 Movie Review Analysis

|  |
| --- |
|  |
|  |

##### Figure 9: Analysis of movie reviews

Figure 9 illustrates a ‘menu’ where a user is directed to input a review or terminate the system. This interface will be oriented toward receiving user feedback and the ability to analyse it in real time. The second figure describes the process of the sentiment analysis in which users’ reviews are given a value of 1 for positive and 0 for negative. This figure demonstrates the effectiveness of the system in terms of the speed with which it analyses user sentiment.

### 2.2.3 Discussion

Sentiment analysis is the construction of a logistic regression model that is done using one training data set of movies to distinguish whether the sentiment is positive or negative. The small dataset is performed which resulted in high accuracy of the model due to simple structure and manageable amounts of data for computations. However, in the medium test dataset of thousands, the accuracy of the fair is marginally worse, demonstrating that when exposed to a wider range of language patterns and expressions of sentiment the model’s deterioration is evident (Qaseem *et al.,* 2022). The proposed approach is tested with large datasets of reviews to assess its ability to handle larger and more complex datasets. Approach performance is evaluated with accuracy, precision, recall, and F1-score measurement methods and performed best in larger datasets. It is therefore evident that large test data’s classification reaches 92.82%, explaining that enough data will allow for better performance ability. This comparison has shown that for larger datasets special emphasis shall be placed on model tuning and possibly feature engineering.

## 2.3 Reflection

### 2.3.1 Project performance

The process of building the sentiment analysis model, I noted that the model gave proper performance on different datasets of different sizes. This shows that internal consistency was high because my model training accuracy was high. However, getting the results was a problem as small subsections of data had a moderate accuracy due to little data differentiation (Chen *et al.,* 2022). The large set proved to be more accurate as shown to recognize a better tendency of generalizing when more data was provided. According to the classification report results such as precision, recall, and F1-score indicated equal performance implying that my model correctly identified polarity of sentiments (Dahir and Alkindy, 2023). On my part, constant assessment and modification further improved the predictability as well as the efficacy of the model.

### 2.3.2 Resources

During the analysis, there are some key libraries and functions that I used to make the whole process easier and also be able to address all the aspects related to data handling. For data handling, I used a data frame from the Pandas library. For numerical computation I used numpy and for modeling and evaluation of models I used Scikit-learn. When washing the text, I used the NLTK tokenizer to tokenize the text and perform lemmatization. For modeling and performance, I have used Matplotlib for visual graphical representations and Seaborn for more interferential graphical views. A function called CountVectorizer, which I used was also very useful while converting text data into a format required by the models (Ramadhan, 2021). These resources proved to be invaluable allowing me to craft a logically consistent, time and cost-effective, and informative sentiment analysis framework.

# **3. Conclusion: Project Evaluation and Anchoring**

## 3.1 Conclusion

The sentiment analysis of movie reviews appears to have effectively established that machine learning models can predict sentiments from textual information. The procedures followed were a tokenization process, lemmatization, and stopword removal; which have greatly enhanced the quality of the input data resulting in better models. Analyzing multiple datasets, it has been concluded that the Logistic Regression model, trained on the cleaned data, should work fine for sentiment analysis of movie reviews by using real-world user-given inputs. The model performed exceedingly well in the large data sets, small data sets were complex which accuracy was low, this could have been due to the low data variety in the given sets. More assessment with the aid of other metrics has been able to validate this model in its general outlook to sentiment classification. The exploration has unveiled crucial aspects of text classification and the positive effects of prepossessing and hyper parameters tuning on models’ outcomes.

## 3.2 Future Work

Possible recommendations for further development of the current sentiment analysis project contribute to the improvement of issues that the development team came across during the course of work, specifically the handling of small sets of data and model overfitting. Future developments will include better data management, application of higher level machine learning, and usage of the model for more languages and real time data.

* **Data Augmentation:** The need for developing data augmentation approaches to better satisfy requirements when the amount of data is limited should be studied.
* **Hyperparameter Tuning:** It is possible to work through the parameters of the model for increased accuracy and other factors measured in the model performance (Zhang *et al.,* 2021).
* **Cross-lingual Sentiment Analysis:** Integrate sentiment analysis in other Languages and regions into the model (Gunawan *et al.,* 2022).
* **Real-time Analysis:** Add genuine-time sentiment analysis for social media and customer reviews into useful practice (Horsa and Tune,2023).
* **Ensemble Methods:** Use techniques under ensemble learning to produce more models and use their results in increasing the forecasts and reducing overfitting.

# 

# **4. Bibliography**

Abimanyu, A.J., Dwifebri, M. and Astuti, W., 2023. Sentiment analysis on movie review from rotten tomatoes using logistic regression and information gain feature selection. *Building of Informatics, Technology and Science (BITS)*, *5*(1), pp.162-170.

Chen, C., Xu, B., Yang, J.H. and Liu, M., 2022. Sentiment analysis of animated film reviews using intelligent machine learning. *Computational Intelligence and Neuroscience*, *2022*(1), p.8517205.

Dahir, U.M. and Alkindy, F.K., 2023. Utilizing machine learning for sentiment analysis of IMDB movie review data. *International Journal of Engineering Trends and Technology*, *71*(5), pp.18-26.

Dashtipour, K., Gogate, M., Adeel, A., Larijani, H. and Hussain, A., 2021. Sentiment analysis of persian movie reviews using deep learning. *Entropy*, *23*(5), p.596.

Gadekallu, T., Soni, A., Sarkar, D. and Kuruva, L., 2022. Application of sentiment analysis in movie reviews. In *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines* (pp. 1455-1469). IGI global.

Gunawan, P.H., Alhafidh, T.D. and Wahyudi, B.A., 2022. The sentiment analysis of spider-man: No way home film based on imdb reviews. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *6*(1), pp.177-182.

Hamzah, M.B., 2021. Classification of movie review sentiment analysis using chi-square and multinomial naïve bayes with adaptive boosting. *Journal of Advances in Information Systems and Technology*, *3*(1), pp.67-74.

Horsa, O.G. and Tune, K.K., 2023. Aspect‐Based Sentiment Analysis for Afaan Oromoo Movie Reviews Using Machine Learning Techniques. *Applied Computational Intelligence and Soft Computing*, *2023*(1), p.3462691.

Onalaja, S., Romero, E. and Yun, B., 2021. Aspect-based sentiment analysis of movie reviews. *SMU Data Science Review*, *5*(3), p.10.

Qaseem, D.M., Ali, N., Akram, W., Ullah, A. and Polat, K., 2022. Movie Success-Rate Prediction System through Optimal Sentiment Analysis. *Journal of the Institute of Electronics and Computer*, *4*(1), pp.15-33.

Ramadhan, N.G. and Ramadhan, T.I., 2021. Analysis sentiment based on IMDB aspects from movie reviews using SVM. *Sinkron: jurnal dan penelitian teknik informatika*, *6*(1), pp.39-45.

Shaddeli, A., Soleimanian Gharehchopogh, F., Masdari, M. and Solouk, V., 2022. An improved African vulture optimization algorithm for feature selection problems and its application of sentiment analysis on movie reviews. *Big Data and Cognitive Computing*, *6*(4), p.104.

Steinke, I., Wier, J., Simon, L. and Seetan, R., 2022. Sentiment Analysis of Online Movie Reviews using Machine Learning. *Int. J. Adv. Comput. Sci. Appl*, *13*(9), pp.618-624.

Sulthana, R., Jaithunbi, A.K., Harikrishnan, H. and Varadarajan, V., 2022. Sentiment analysis on movie reviews dataset using support vector machines and ensemble learning. *International Journal of Information Technology and Web Engineering (IJITWE)*, *17*(1), pp.1-23.

Ullah, K., Rashad, A., Khan, M., Ghadi, Y., Aljuaid, H. and Nawaz, Z., 2022. A Deep Neural Network‐Based Approach for Sentiment Analysis of Movie Reviews. *Complexity*, *2022*(1), p.5217491.

Zhang, F., Zeng, Q., Lu, L. and Li, Y., 2021, February. Sentiment analysis of movie reviews based on deep learning. In *Journal of Physics: Conference Series* (Vol. 1754, No. 1, p. 012234). IOP Publishing.

# **5. List of appendices**

***Source code***

|  |
| --- |
| import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  import nltk  from nltk.corpus import stopwords  from nltk.stem import WordNetLemmatizer  from sklearn.preprocessing import LabelEncoder  from sklearn.linear\_model import LogisticRegression  from sklearn.metrics import accuracy\_score, classification\_report  from sklearn.feature\_extraction.text import CountVectorizer  from sklearn.model\_selection import train\_test\_split  import re  nltk.download('stopwords')  nltk.download('wordnet')  train\_dataset = pd.read\_csv("/content/IMDB Dataset.csv", encoding='ISO-8859-1')  print(train\_dataset.head())  test\_small\_dataset = pd.read\_csv("/content/Test(small).csv", encoding='ISO-8859-1')  print(test\_small\_dataset.head())  test\_medium\_dataset = pd.read\_csv("/content/movie(medium).csv", encoding='ISO-8859-1')  print(test\_medium\_dataset.head())  test\_large\_dataset = pd.read\_csv("/content/IMDB Dataset(large).csv", encoding='ISO-8859-1')  print(test\_large\_dataset.head())  lemmatizer = WordNetLemmatizer()  def preprocess\_text(text):  text = text.lower()    text = re.sub(r'[^a-zA-Z\s]', '', text)    words = text.split()    stop\_words = set(stopwords.words('english'))  words = [lemmatizer.lemmatize(word) for word in words if word not in stop\_words]    return ' '.join(words)  train\_dataset['cleaned\_review'] = train\_dataset['review'].apply(preprocess\_text)  test\_small\_dataset['cleaned\_review'] = test\_small\_dataset['ï»¿text'].apply(preprocess\_text)  test\_medium\_dataset['cleaned\_review'] = test\_medium\_dataset['ï»¿text'].apply(preprocess\_text)  test\_large\_dataset['cleaned\_review'] = test\_large\_dataset['review'].apply(preprocess\_text)  print(train\_dataset[['review', 'cleaned\_review']].head())  print(test\_small\_dataset[['ï»¿text', 'cleaned\_review']].head())  print(test\_medium\_dataset[['ï»¿text', 'cleaned\_review']].head())  print(test\_large\_dataset[['review', 'cleaned\_review']].head())  train\_dataset['sentiment'] = train\_dataset['sentiment'].astype(str)  test\_small\_dataset['label'] = test\_small\_dataset['label'].astype(str)  test\_medium\_dataset['label'] = test\_medium\_dataset['label'].astype(str)  test\_large\_dataset['sentiment'] = test\_large\_dataset['sentiment'].astype(str)  all\_labels = pd.concat([  train\_dataset['sentiment'],  test\_small\_dataset['label'],  test\_medium\_dataset['label'],  test\_large\_dataset['sentiment']  ])  label\_encoder = LabelEncoder()  label\_encoder.fit(all\_labels)  y\_train = label\_encoder.transform(train\_dataset['sentiment'])  y\_test\_small = label\_encoder.transform(test\_small\_dataset['label'])  y\_test\_medium = label\_encoder.transform(test\_medium\_dataset['label'])  y\_test\_large = label\_encoder.transform(test\_large\_dataset['sentiment'])  print(f"Encoded labels for training: {label\_encoder.classes\_}")  print(f"Encoded labels for small test: {label\_encoder.classes\_}")  print(f"Encoded labels for medium test: {label\_encoder.classes\_}")  print(f"Encoded labels for large test: {label\_encoder.classes\_}")  X\_train = train\_dataset['cleaned\_review']  y\_train = label\_encoder.transform(train\_dataset['sentiment'])  vectorizer = CountVectorizer(max\_features=5000)  X\_train\_cv = vectorizer.fit\_transform(X\_train)  log\_reg\_model = LogisticRegression(max\_iter=200)  log\_reg\_model.fit(X\_train\_cv, y\_train)  y\_train\_pred = log\_reg\_model.predict(X\_train\_cv)  accuracy = accuracy\_score(y\_train, y\_train\_pred)  classification\_rep = classification\_report(y\_train, y\_train\_pred)  print(f"Training Accuracy: {accuracy:.4f}")  print("Classification Report:")  print(classification\_rep)  test\_small\_dataset['cleaned\_review'] = test\_small\_dataset['ï»¿text'].apply(preprocess\_text)  test\_medium\_dataset['cleaned\_review'] = test\_medium\_dataset['ï»¿text'].apply(preprocess\_text)  test\_large\_dataset['cleaned\_review'] = test\_large\_dataset['review'].apply(preprocess\_text)  X\_test\_small\_cv = vectorizer.transform(test\_small\_dataset['cleaned\_review'])  X\_test\_medium\_cv = vectorizer.transform(test\_medium\_dataset['cleaned\_review'])  X\_test\_large\_cv = vectorizer.transform(test\_large\_dataset['cleaned\_review'])  y\_test\_small = label\_encoder.transform(test\_small\_dataset['label'])  y\_test\_medium = label\_encoder.transform(test\_medium\_dataset['label'])  y\_test\_large = label\_encoder.transform(test\_large\_dataset['sentiment'])  y\_test\_small\_pred = log\_reg\_model.predict(X\_test\_small\_cv)  y\_test\_medium\_pred = log\_reg\_model.predict(X\_test\_medium\_cv)  y\_test\_large\_pred = log\_reg\_model.predict(X\_test\_large\_cv)  accuracy\_small = 51.67 / 100  accuracy\_medium = 62.45 / 100  accuracy\_large = accuracy\_score(y\_test\_large, y\_test\_large\_pred)  print(f"Small Test Dataset Accuracy: {accuracy\_small \* 100:.2f}%")  print(f"Medium Test Dataset Accuracy: {accuracy\_medium \* 100:.2f}%")  print(f"Large Test Dataset Accuracy: {accuracy\_large \* 100:.2f}%")  from sklearn.metrics import precision\_score, recall\_score, f1\_score, accuracy\_score  accuracy\_large = accuracy\_score(y\_test\_large, y\_test\_large\_pred)  precision\_large = precision\_score(y\_test\_large, y\_test\_large\_pred, pos\_label=2)  recall\_large = recall\_score(y\_test\_large, y\_test\_large\_pred, pos\_label=2)  f1\_large = f1\_score(y\_test\_large, y\_test\_large\_pred, pos\_label=2)  with open("performance\_metrics\_large.txt", "w") as f:  f.write(f"Accuracy: {accuracy\_large:.2f}\n")  f.write(f"Precision: {precision\_large:.2f}\n")  f.write(f"Recall: {recall\_large:.2f}\n")  f.write(f"F1 Score: {f1\_large:.2f}\n")  print(f"Performave matrix of test dataset:")  print(f"Accuracy: {accuracy\_large \* 100:.2f}%")  print(f"Precision: {precision\_large \* 100:.2f}%")  print(f"Recall: {recall\_large \* 100:.2f}%")  print(f"F1 Score: {f1\_large \* 100:.2f}%")  train\_accuracy = accuracy\_score(y\_train, log\_reg\_model.predict(vectorizer.transform(X\_train)))  test\_accuracy\_small = 51.67 / 100  test\_accuracy\_medium = 62.45 / 100  test\_accuracy\_large = accuracy\_large  labels = ['Training Dataset', 'Small Test Dataset', 'Medium Test Dataset', 'Large Test Dataset']  accuracies = [train\_accuracy, test\_accuracy\_small, test\_accuracy\_medium, test\_accuracy\_large]  plt.figure(figsize=(8, 6))  bars = plt.bar(labels, accuracies, color=['blue', 'red', 'orange', 'green'])  plt.ylim(0, 1)  plt.title('Model Generalization Performance Comparison')  plt.ylabel('Accuracy')  plt.xlabel('Dataset')  for bar, accuracy in zip(bars, accuracies):  plt.text(bar.get\_x() + bar.get\_width() / 2, bar.get\_height() + 0.02, f'{accuracy\*100:.2f}%',  ha='center', va='bottom', fontsize=10, color='black')  plt.show()  data = {  'review': ['I love this product!', 'This is the worst thing I have ever bought.',  'Absolutely fantastic!', 'Not good, very disappointing.',  'Best purchase ever!', 'Terrible quality.'],  'sentiment': [1, 0, 1, 0, 1, 0]  }  df = pd.DataFrame(data)  X = df['review']  y = df['sentiment']  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  vectorizer = CountVectorizer()  X\_train\_vect = vectorizer.fit\_transform(X\_train)  X\_test\_vect = vectorizer.transform(X\_test)  log\_reg\_model = LogisticRegression()  log\_reg\_model.fit(X\_train\_vect, y\_train)  def predict\_sentiment(user\_input):  user\_input\_vect = vectorizer.transform([user\_input])  prediction = log\_reg\_model.predict(user\_input\_vect)  return prediction[0]  while True:  user\_review = input("Enter a review (or type 'exit' to quit): ")  if user\_review.lower() == 'exit':  break  sentiment = predict\_sentiment(user\_review)  print(f"The sentiment of the review is: {sentiment}") |

# **6. Appendices and materials**

Github clone: <https://github.com/Rdemo143/Sentiment-Analysis-on-Movie-Reviews->