Project

Part A: IMDb Movie Review Sentiment Analysis

1. Overview

Sentiment analysis is a natural language processing (NLP) task that involves determining whether a given text expresses a positive or negative sentiment. In this project, we will analyze movie reviews from the IMDb dataset and predict the sentiment (positive or negative) based on the text of the reviews. By leveraging various text preprocessing techniques, feature extraction methods, and classification algorithms, this project will develop a machine learning model capable of accurately predicting the sentiment of movie reviews. The insights derived from this analysis can be useful for movie producers, critics, and platforms like IMDb to understand public opinion and tailor marketing or content strategies accordingly.

2. Problem Statement

The primary objective of this project is to build a machine learning classification model that can predict the sentiment of IMDb movie reviews. The dataset contains a collection of movie reviews, and each review is labeled as either positive or negative.

Using text preprocessing, feature extraction techniques (such as TF-IDF), and various classification algorithms, the project will aim to develop a model that can effectively classify the sentiment of movie reviews. The model's performance will be evaluated using standard classification metrics, such as accuracy, precision, recall, and F1-score.

3. Dataset Information

The IMDb dataset contains a large number of movie reviews, each labeled with either a positive or negative sentiment.

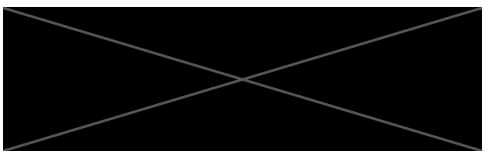
- **Text of the review**: The actual review provided by the user.
- **Sentiment label**: The sentiment of the review, either "positive" or "negative."

Dataset: **Imdb**

4. Deliverables

1. Data Exploration and Preprocessing (5 Marks)

- **Analyze the dataset for trends, missing values, and outliers**.
    - Perform basic data exploration, such as checking for missing values, identifying imbalanced classes (positive/negative), and analyzing the length of reviews.
- **Perform data cleaning and text preprocessing**.
    - Steps will include:
        - Removing stop words, punctuation, and special characters.



- Tokenization of text (splitting text into words).
- Lemmatization and stemming.
- Vectorization using techniques like Bag-of-Words and TF-IDF.

2. Feature Engineering (10 Marks)

- **Feature extraction using techniques like TF-IDF, Word2Vec, or embeddings**.
  - Transform the textual data into numerical features that can be used by machine learning models.
- **Textual features**: Word count, character count, average word length, etc.

3. Model Development (20 Marks)

- **Build and train classification models to predict the sentiment of reviews**.
  - Experiment with various classification algorithms such as Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Random Forest, and Neural Networks (e.g., LSTM, BERT, etc.).

4. Model Evaluation (5 Marks)

- Evaluate the model's performance using appropriate metrics.

5. Final Report and Presentation (10 Marks)

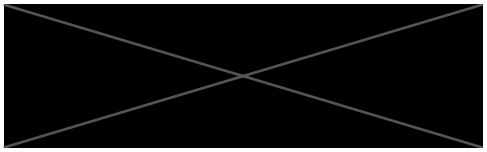
- **Create a final report** that documents the entire process, from data exploration and preprocessing to model evaluation.
- **Video presentation** (maximum 5 minutes) summarizing the key findings, model development process, and insights derived from the project.

5. Success Criteria

The project will be deemed successful if:

- The classification model achieves an acceptable performance on the test data using metrics like accuracy, F1-score, and ROC-AUC.
- Insights regarding the factors influencing sentiment (such as word frequency, review length, etc.) are clearly communicated.
- Predictions for new movie reviews can be made with a reasonable degree of accuracy.
- The final presentation effectively communicates the results and analysis of the project.
- **Visualizations**: Use plots, such as bar charts, confusion matrices, and word clouds, to clearly present data trends and model results.

6. Tools Required



- **Python Libraries**: pandas, numpy, scikit-learn, matplotlib, seaborn, NLTK, spaCy, TensorFlow/Keras, XGBoost, etc.
- **Jupyter Notebook**: For developing and documenting the code.
- **Text Preprocessing Libraries**: NLTK, spaCy, scikit-learn (for vectorization), etc.
- **Visualization Libraries**: matplotlib, seaborn (for visualizing data distributions, model performance, etc.).

Part B : News Article Classification

1. Overview

In today's digital world, news articles are constantly being generated and shared across different platforms. For news organizations, social media platforms, and aggregators, classifying articles into specific categories such as sports, politics, and technology can help improve content management and recommendation systems. This project aims to develop a machine learning model that can classify news articles into predefined categories, such as sports, politics, and technology, based on their content.

By automating this process, organizations can efficiently categorize large volumes of news articles, making it easier for readers to access relevant information based on their interests.

2. Problem Statement

The primary objective of this project is to build a classification model that can automatically categorize news articles into different predefined categories. The model will be trained using a labeled dataset of news articles and will output the most likely category (e.g., sports, politics, or technology) for any given article.

The goal is to:

- Develop a robust classifier capable of handling articles from multiple categories.
- Preprocess the text data, extract meaningful features, and train models to classify the articles.
- Evaluate the model performance and provide actionable insights on how well it classifies articles.

3. Dataset Information

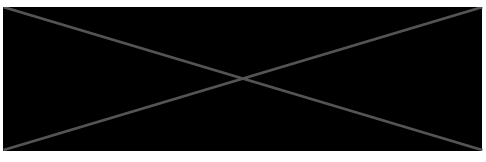
The dataset can be used from **data_news.**

4. Deliverables

1. Data Collection and Preprocessing (5 Marks):

- Collect a dataset of labeled news articles (sports, politics, technology etc).
- Clean and preprocess the text data.
- Handle missing data, if any, and ensure the text is ready for feature extraction.

2. Feature Extraction (10 Marks):



- Use methods like TF-IDF, word embeddings (e.g., Word2Vec, GloVe), or bag-of-words to convert text data into numerical features.
- Perform exploratory data analysis (EDA) to understand the distribution of different categories.

3. Model Development and Training (20 Marks):

- Build classification models using algorithms like Logistic Regression, Naive Bayes, Support Vector Machines (SVM).
- Train the models on the preprocessed text data, tuning hyperparameters as necessary.
- Use cross-validation to ensure robust evaluation of model performance.

4. Model Evaluation (5 Marks):

- Evaluate the models using appropriate metrics.
- Compare the performance of different models and select the best one for classification.

5. Final Report and Presentation (10 Marks):

- Create a report summarizing the entire process, from data collection to model evaluation, and present the findings.
- Include visualizations of model performance and feature importance, if applicable.
- Prepare a video or slide presentation (not exceeding 5 minutes) explaining the methodology, models, and results.

5. Success Criteria

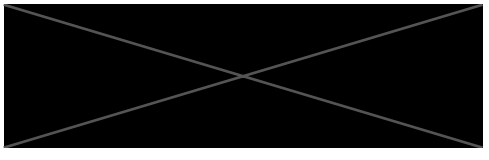
The project will be deemed successful if:

- The classification model achieves good performance metrics (accuracy, F1-score, etc.).
- The model can successfully classify new, unseen news articles into the correct categories (sports, politics, technology).
- Insights regarding the most important features or keywords driving classification are derived from the model.
- The process and methodology are clearly documented and presented.

Submit Guidelines

- **Submit in jupyter notebook file (.ipynb file), report.**
- **Create a video of maximum of 5 mins explaining the analysis and share the drivelink.**

How to ZIP a folder:



● Put all files you want to compress into a new folder.

● Right click on that folder.

● Select the “Compress to ZIP file” option and then click “Compressed (Zipped) folder.”

● A new .ZIP file will be created that contains your document(s). Upload this folder.

Note:

● Plagiarism will result in a penalty, including possible project disqualification.

● The project will be evaluated based on the quality of analysis, depth of insights, and feasibility of recommendations.

● Remember to keep the video length less than 5 minutes with your face clearly visible.