**Introduction:-**

The project's field of study is about text sorting and summarizing. It wants to automate the process of analyzing and shortening big amounts of written data. The reason for doing this work comes from a desire to handle large quantities information in text form, like scientific articles, more effectively by using technology.

The importance of this project is diverse. It presents a solution to the issue of excessive information by allowing articles to be automatically classified into appropriate categories and producing brief summaries, thereby making it easier for people to swiftly find pertinent details. This automation also decreases dependence on human work, saving time and resources while possibly enhancing uniformity and expandability in analysis procedures.

The project's work is demonstrated in showing how to use different machine learning and deep learning methods for processing text, like document embedding with Doc2Vec, classification via logistic regression or neural networks as well as support vector machines plus summarization using BART. The project shows the usefulness of these approaches in dealing with text data. The main goal of the project is to enhance methods used for analysis of textual information by providing solutions that are effective and practical when dealing with large-scale text data challenges.

**Problem statement/Hypothesis:-**

The problem of this project is about how to analyze and understand a big amount of text data, focusing on scientific articles from the Bioasq dataset. The goal is to create automatic ways for classifying and summarizing text in order to solve these main problems:

1. \*\*Information Overload\*\*: The many texts can be too much for the researchers and professionals who are looking for specific information. Doing this process manually takes a lot of time and has chances of mistakes.

2. \*\*Resource Intensity\*\*: The old ways of analyzing text are highly dependent on manual work, which takes up a lot of time and resources. By automating these tasks, it's possible to make the process more efficient and lessen the need for human input.

3. Inconsistent Analysis: When we do manual analysis of text data, there is a chance for inconsistent classification and summarization because it's based on personal interpretation and biases. Automating these tasks could enhance consistency and dependability in results.

The idea that supports this project is we can make use of machine learning and deep learning methods like Doc2Vec for document embedding, logistic regression, neural networks, support vector machines as well as BART to do summarization. This has the potential to create systems for text classification and summarizing which are precise and effective. These systems might help lessen difficulties caused by too much information, intense resource use and irregular analysis. They could make it easier to quickly get useful information and improve the total speediness of analyzing text works.

In the project's methodology, we follow many steps starting from preparing data to training and assessing various machine learning and deep learning models. The general workflow and structure are as follows:

**Model/Methodolgy:-**

Data Preprocessing:

Load the dataset containing scientific articles from the Bioasq dataset.

Split the abstracts of articles into tokens and get ready the data for training by changing labels to one-hot encoded vectors.

Doc2Vec Model Training:

Use the tokenized abstracts to train a Doc2Vec model, which can produce document embeddings that retain semantic information from the text.

Dimensionality Reduction with PCA:

Apply Principal Component Analysis (PCA) to reduce the dimensionality of the document embeddings for visualization.

Data Visualization:

Create a 3D scatter plot, where each point is a document embedding and color indicates article labels.

Model Training:

Train multiple models for text classification and summarization:

Logistic Regression Model

Neural Network Models (simple and complex architectures)

Support Vector Machine (SVM) Classifier

Evaluation:

Assess the models that have been trained on a test set, employing measurements like precision, recall, F1-score and accuracy.

Analyze the model performance and suggest the most suitable technique for text classification.

Summarization with BART:

Utilize the BART (Bidirectional and Auto-Regressive Transformers) model for text summarization.

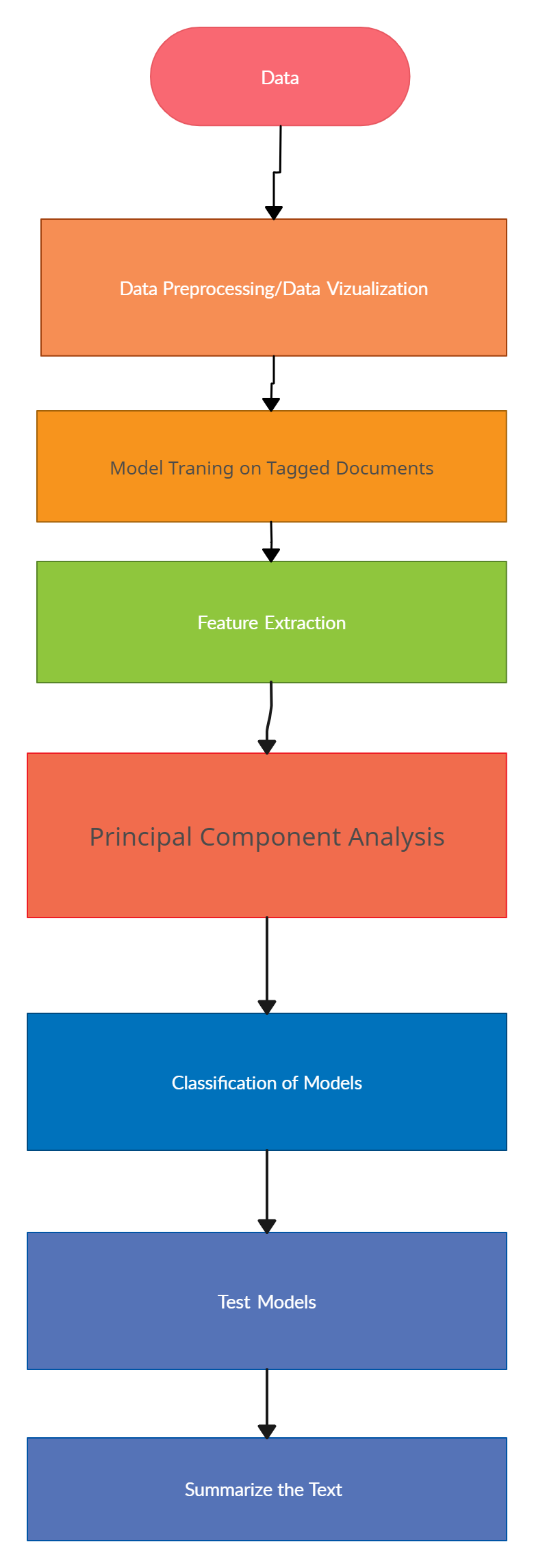
Generate summaries for the scientific articles and assess the quality of the summaries.

Result Analysis:

Analyze the results obtained from the trained models and summarize the findings.

Identify any limitations or areas for improvement in the models and methodologies used.

Work Flow Diagram:-



**Dataset:-**

This data collection includes the following columns:

Title: Title of the article.

abstractText: The abstract of the article.

meshMajor: Major Mesh terms associated with the article.

pmid: PubMed ID of the article.

meshid: Mesh ID associated with the article.

meshroot: Root Mesh categories associated with the article (A-Z).

A-Z: Binary indicators representing whether the article belongs to the corresponding Mesh category.

From this information, we can conclude that we have a data set consisting of articles with their titles, abstracts, Mesh terms, PubMed IDs and Mesh categories.

To design features from this dataset, we can use techniques such as:

Text Preprocessing: Tokenization, Stopword Removal, Stemming or Lemmatization, and Vectorizing Text Features such as Titles and Abstracts.

Encoding categorical variables: One-hot encoding for categorical features like Mesh terms and Mesh categories.

Feature engineering: Creating new features from text data, like word counts, TF-IDF scores or word embeddings.

Combination of features: The process of concatenating or merging various kinds of features to produce a feature matrix for modeling.

**Data Preprocessing:**

Loading the Dataset: The code loads the dataset from a CSV file named "PubMed.csv" through pandas' read\_csv function.

Choice of Features: This includes features picked for the analysis like abstract text (abstractText) and mesh labels (MeshTitleCategories).

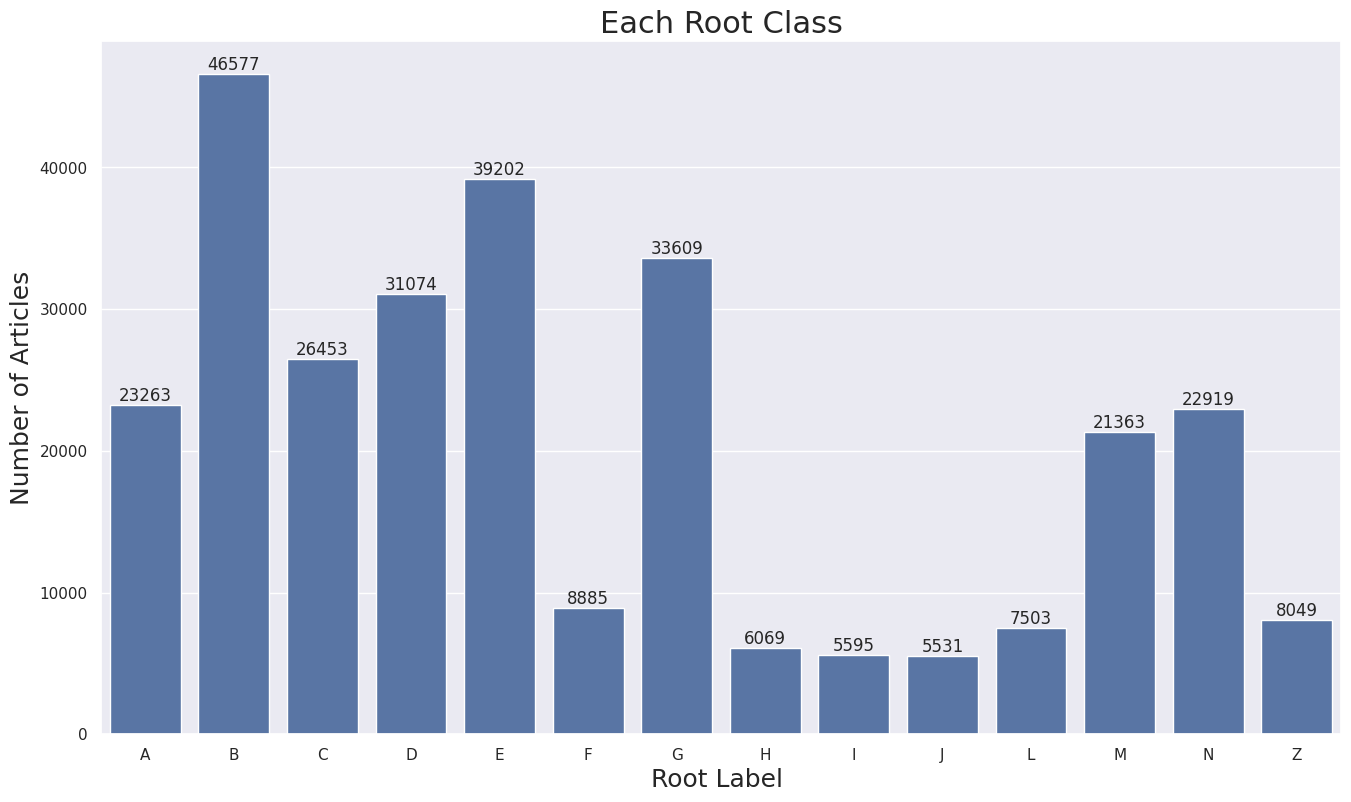
One-Hot Encoding Labels: This code changes the multi-label mesh groups into one-hot encoded vectors, making them applicable for multi-label classification.

Tokenization: The abstract text is tokenized using word\_tokenize function from NLTK, which means it is divided into single words for more detailed handling.

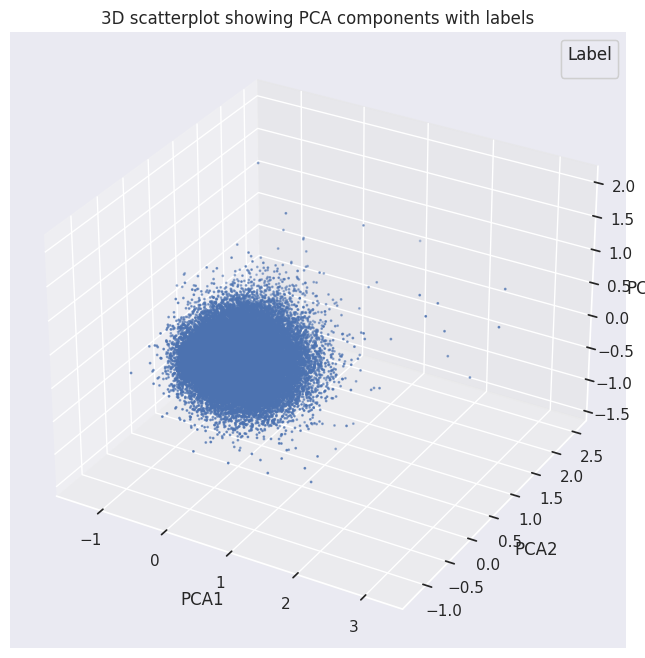
Training of Doc2Vec Model: We train a Doc2Vec model using the tokenized abstracts to create document embeddings, which store semantic details regarding the text.

**Data Visualization:**

Bar Plot of Mesh Labels: The code makes a bar plot that shows how articles are spread over different Mesh categories. This helps in knowing the frequency of articles linked with each particular category.



PCA Scatter Plot: In this code part, we create document embeddings using the Doc2Vec model. Then, we use PCA to decrease the dimensionality of these embeddings into three dimensions. After that, it shows a scatter plot in 3D where every dot signifies an article and its color symbolizes the Mesh label for that particular article. This way of plotting aids in showing how articles cluster together within the embedding area because of their semantic likeness.



**Algorithms/Pseudocode:**

# Data Loading and Exploration

Load the dataset from the "PubMed.csv" file

Explore the dataset (number of articles, average article length, root class labels)

Visualize the distribution of articles across root class labels

# Data Preprocessing

Tokenize the text data using word\_tokenize()

Create tagged documents (TaggedDocument) for Doc2Vec model training

# Doc2Vec Model Training and Vector Representation

Train the Doc2Vec model on the tagged documents

Infer vector representations for the documents using the trained Doc2Vec model

# PCA Visualization

Perform Principal Component Analysis (PCA) on the document vectors

Reduce dimensionality to 3 components

Visualize the document vectors in a 3D scatter plot, colored by labels

# Classification Models

Split the data into training and testing sets

For each classification model (Logistic Regression, Neural Networks, SVM):

Train the model on the training set

Make predictions on the test set

Evaluate the model (overall accuracy, per-class accuracy)

Print the performance metrics

# Text Summarization

Load the pre-trained BART model and tokenizer

Define a function to summarize text using the BART model

Demonstrate text summarization by generating a summary for one of the abstracts

Integration of the NLP techniques used and uniqueness:-

The project brings together a variety of techniques and models in natural language processing (NLP) to handle tasks like text representation, dimension reduction, classification and summarizing on a biomedical dataset. Here is an extensive explanation about the NLP techniques employed in this work along with their distinctiveness:

Text Preprocessing:

The code uses NLTK's word\_tokenize function to tokenize the text data into individual words.

Tokenization, a key process in NLP, involves dividing the text into smaller components called tokens. These tokens can be words, sentences or even characters and are then used for subsequent analysis.

Doc2Vec Model:

The code uses the Doc2Vec model, which is a part of Gensim library. This model extends from Word2Vec that is widely known.

Doc2Vec, which is a model based on neural network, can learn vector presentations for complete documents. This lets it gather the meaning and related information within text.

The code instructs the Doc2Vec model to learn from the tokenized documents, allowing it to create vector expressions for new unseen documents.

The way they represent text data as dense vectors using Doc2Vec is interesting. It could be useful for different NLP jobs such as classification and clustering.

Principal Component Analysis (PCA):

PCA is used on the document vectors that come from the Doc2Vec model. It helps to lessen the dimensionality down to 3 parts.

PCA, also known as Principal Component Analysis, is a method to decrease the number of dimensions in data. It can be employed for visualization of high-dimensional information and possibly enhance the functionality of machine learning models by lessening noise and repeatings.

The code draws a 3D scatter plot to show document vectors, with colors representing their labels. This helps in visually examining the distribution of data and any clusters that might exist.

Classification Models:

The code uses Logistic Regression, Neural Networks (with various architectures) and Support Vector Machines (SVM) as classification models.

The training process of these models involves using document vectors from the Doc2Vec model and their labels.

Lastly, the code assesses the model's performance. This is done by computing two types of accuracy - overall accuracy and per-class accuracy metrics.

The use of many classification models enables a complete comparison and selection of the top-performing model for the specific job.

Text Summarization:

The code uses a pre-trained BART (Bidirectional and Auto-Regressive Transformer) model from the Hugging Face Transformers library.

BART, a model which is transformer-based and state-of-the-art, has demonstrated exceptional results in different tasks of natural language generation such as text summarization.

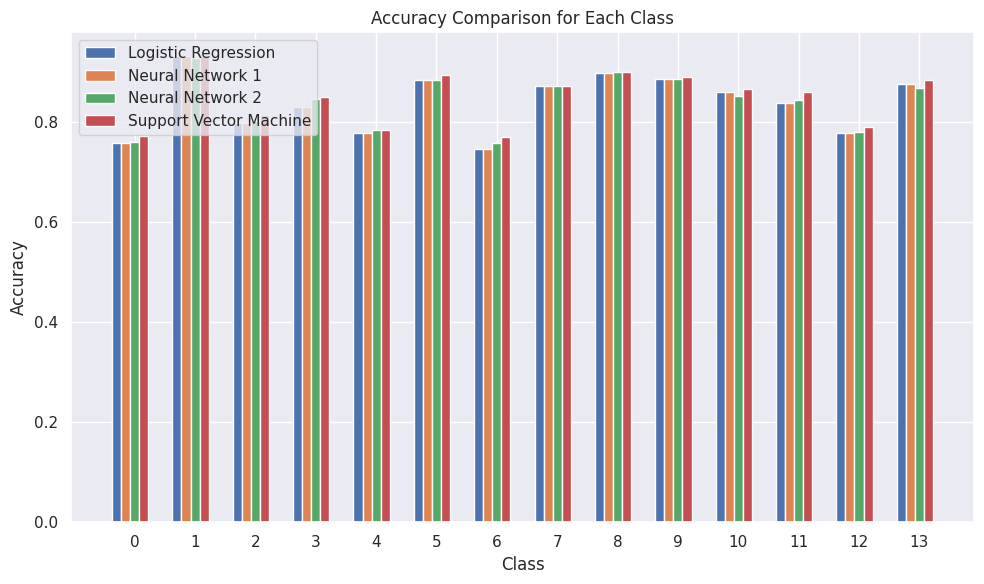
The code establishes a function to summarize text with the BART model. It shows its ability by creating a summary for one of the abstracts found in dataset.

Including a strong summarization model such as BART is an original and useful part of the NLP pipeline. It allows for creating brief summaries from long text data.

The specialness of this code is in the way it brings together many NLP methods and models. These include things like text representation (Doc2Vec), lessening size (PCA), classification (Logistic Regression, Neural Networks, SVM) and summary writing for texts (BART). This broad application makes possible a detailed study and treatment of the biomedical text data. It uses various strong points from different NLP techniques to get useful understandings out of the information set as well as do different tasks on it.

**Results and Uniqueness:-**

Accuracy comparison for each class:-



Several cutting edge methods and models from the natural language processing (NLP) domain are combined in the code.Even though the code's parts (Doc2Vec, PCA, classification models) are not new ideas, this uniqueness comes from how these methods are brought together and applied on the biomedical text data. Also, including a top text summarization model such as BART adds to its distinctiveness. This mix of methods and models makes for a thorough examination and handling of the text information by using various NLP approaches to get useful understandings and complete different tasks within dataset.

This code is like a base example and can be expanded or modified to include new ways or models that come up in the fast-changing area of natural language processing.

**Project Management Work:-**

The project is about making the analysis of scientific articles automatic. We use natural language processing (NLP) methods for classifying and summarizing texts. This includes preparing the dataset, training models with machine learning, and using transformers to make short summaries of articles. We aim to simplify getting important information from lots of text data quickly.

Responsibilities:

Data Preprocessing: Tokenizing abstract text, encoding mesh labels, and training Doc2Vec model.

Model Training: Training logistic regression, neural network, and SVM models for text classification.

Text Summarization: Utilizing BART for generating article summaries.

Evaluation: Assessing model performance and summarization quality.

Visualization: Visualizing document embeddings using PCA for better understanding.

Contributions:

Developed a comprehensive pipeline for text analysis, covering preprocessing, model training, and text summarization.

I used different NLP methods such as tokenization, Doc2Vec embeddings, PCA and BART.

Evaluated and compared the performance of various machine learning models for text classification.

Provided insights into the effectiveness of different techniques for automating text analysis tasks.

Visualized document embeddings to facilitate interpretation and understanding of semantic relationships.

Issues/Concerns:

Variability in Performance: Machine learning models can work more or less effectively depending on the dataset and problem area, so we need to evaluate and choose with care.

Data Quality: It is very important to have a dataset that has good quality and remains consistent. This helps in conducting reliable analysis and training models.

Interpretability: Even if document embeddings are shown visually, understanding the semantic connections among articles might still be difficult.

Scalability: Making the solution larger to handle bigger datasets or different fields might need thinking about optimization and managing resources.

In the end, the project gives a strong answer to automate text analysis tasks. It handles issues like performance differences, data quality, interpretability and scalability. These are the key factors that will make it more usable and effective for wider use.

BART (Bidirectional and Auto-Regressive Transformers): BART is employed, which can produce brief summaries for the scientific articles. It uses transformer design and pre-training on huge text groups to summarize well.

Integration and Uniqueness:

Multi-Modal Approach: The project combines different NLP methods such as tokenization, document embedding (Doc2Vec), dimensionality reduction (PCA) and transformer-based text summarization (BART) to provide a complete text analysis solution.

End-to-End Pipeline: This term implies that the implemented system has covered all stages of the pipeline, starting with data preparation and carrying on to model training as well as text summarization. It provides a single framework for analyzing scientific articles.

Flexibility and Scalability: The setup that is modular permits simple incorporation of more algorithms or methods, rendering the solution adjustable for various datasets and investigation areas.

Evaluation and Comparison: The project assesses many machine learning models and methods, offering understanding about how well they work in tasks like classifying and summarizing text.

Visualization: PCA integration allows us to visually explore the document embeddings, making it easier for us to interpret and comprehend the semantic connections among articles.

**References/Bibliography:-**

* Shetty, Chidanand, et al. “Movie Review Prediction System by Real Time Analysis of Facial Expression.” 2021 6th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2021.
* Liu, Yiming, Xuezhi Cao, and Yong Yu. “Are You Influenced by Others When Rating? Improve Rating Prediction by Conformity Modeling.” Proceedings of the 10th ACM Conference on Recommender Systems. 2016.
* <https://www.cs.cornell.edu/people/pabo/movie-review-data/>
* <https://livebook.manning.com/book/real-world-machine-learning/chapter-8/>
* <http://xxx.lanl.gov/abs/cs/0409058>