­December 6, 2023

**Movie Revies Sentiment Analysis**

(Final Report)

**SUBMITTED TO:**

Professor Bhavik Gandhi

**PREPARED BY:**

Abhishek Natani (863325)

Auradee Castro (866821)

Bhumika Rajendra Babu (867081)

Miraj Sinya (863371)

Olivia Deguit (878491)

Rochan Mehta (863914)

Roger Mais (863147)

Varun Sharma (863237)

**Table of Contents**

[I. Project Description 3](#_Toc152786334)

[Case Background 3](#_Toc152786335)

[Project Background 3](#_Toc152786336)

[II. Github Repository 4](#_Toc152786337)

[Project Board Walkthrough 4](#_Toc152786338)

[Best Git Practices 4](#_Toc152786339)

[III. Model Building and Evaluation 5](#_Toc152786340)

[IV. Application Features Overview 6](#_Toc152786341)

[V. Future Improvements 7](#_Toc152786342)

[VI. MECE Table 7](#_Toc152786343)

# Project Description

## **Case Background**

A screenshot of a computer

Description automatically generated

## **Project Background**

A diagram with text and images

Description automatically generated with medium confidence

# Github Repository

## **Project Board Walkthrough**

We have optimized our model development process by integrating GitHub's Project as a Kanban Board, significantly improving our project tracking and management. Our structured approach includes:

**Kanban Board Statuses**

* **To Do**: Lists tasks that are planned but not yet started.
* **In Progress:** Tracks tasks that are currently being worked on.
* **Blocker:** Identifies tasks that are facing obstacles or delays.
* **Done:** Marks tasks that have been completed and are ready for deployment.
* **For Testing**: Denotes tasks that are in the pre-production testing phase.
* **Closed:** Indicates tasks that have been successfully deployed in the production environment.

**GitHub Branch Management**

* **Tasks Branch:** For specific, individual tasks and features under development.
* **Dev Branch:** Used for ongoing development, integrating new features and updates.
* **Test Branch:** Dedicated to rigorous testing of new features and bug fixes.
* **Prod Branch:** The production branch, where fully tested and stable versions of the software are maintained.

## **Best Git Practices**

**Best Practices in GitHub Project Management**

* **Feature Breakdown:** Splitting features into smaller, manageable tasks for effective prioritization.
* **Distributed Responsibilities:** Assigning tasks to specific team members, ensuring clear accountability.
* **Project Progress Checkpoints:** Regular meetings, either via MS Teams or in-person, to discuss updates, tackle blockers, and strategize upcoming tasks.
* **Standardized Workflow:** Adhering to industry standards in GitHub usage, with clear distinctions between branches for production, testing, development, and individual tasks, and implementing thorough code reviews before any merges.

This approach, leveraging GitHub's functionalities and best practices, allows us to maintain a high level of organization and efficiency in our model development projects.

# Model Building and Evaluation

A diagram of a model

Description automatically generated

In the context of the sentiment analysis model development project, we have conducted a comprehensive model building and evaluation process as delineated below:

**Model Development Workflow**

* The workflow initiated with the data gathering of a labelled dataset for the model.
* Subsequent to dataset gathering, the model selection and training phase commenced, wherein the dataset was split into training and testing subsets to facilitate model training and evaluation.
* Post-training, the model was subjected to a rigorous evaluation to ascertain its performance metrics.

**Performance Evaluation Insights**

* The project saw the training and comparative performance analysis of three distinct algorithmic models: the **Multinomial Naïve Bayes**, **Long Short-Term Memory (LSTM)**, and **Gated Recurrent Unit (GRU)**.
* The **Multinomial Naïve Bayes** model emerged as the superior performer, demonstrating a commendable accuracy rate of **71%.** This is in contrast to the LSTM and GRU models, which exhibited accuracy rates of 59% and 58%, respectively. Hence our project decided to use the **Multinomial Naïve Bayes** model.
* An optimization procedure was undertaken employing the **GridSearchCV** for hyperparameter refinement, concentrating specifically on the 'Alpha' parameter. It is noteworthy that adjustments to the alpha parameter were found to have a consequential impact on model precision and accuracy. An optimal alpha was sought, aiming to achieve a harmonious balance across pivotal performance indicators such as accuracy, precision, recall, and the F1-score.

**Additional Considerations**

* The Receiver Operating Characteristic (ROC) and the Area Under the ROC Curve (AUC) were identified as key metrics for model evaluation.

This summarizes the model building and evaluation stages in the sentiment analysis project, detailing the methodologies adopted and the outcomes achieved in terms of model performance and optimization.

# Application Features Overview

A diagram of a software application

Description automatically generated

For the final application features, the project implemented a combination of **Naïve Bayes** and **KNN (k-nearest neighbors)** algorithms. The workflow is as follows:

* **Initial Stage:** Raw movie reviews are processed through a Tuned Sentiment Analysis Model employing Naïve Bayes to classify sentiment.
* **User Input Processing:** The application receives a movie name as input, which interacts with a Sentiment Labelled Dataset to determine if the movie exists within it.
* **Encoding and Matching:**
  + If the movie name exists, One Hot Encoding is applied for attributes such as genre, rating, director, sentiment, and cast, feeding into the Nearest Neighbor Model.
  + For new or unrecognized movie names, a TF-IDF Vectorizer is utilized.
* **Output Generation**
  + **Feature 1:** Generates review sentiment.
  + **Feature 2:** Provides aspect-based review sentiment.
  + **Feature 3:** Offers movie-based review sentiment.
  + **Feature 4:** Suggests movie recommendations and a list of top 10 movies related to the user's input based on the encoded query and review embeddings.

The model leverages sentence transformer encoding to convert reviews into embeddings, which are then matched against user queries using cosine similarity (indicated as CS in the diagram).

Overall, the system is designed to deliver insightful sentiment analysis and personalized movie recommendations based on user queries and review data.

# Future Improvements

Outlined below are proposed enhancements to the sentiment analysis application:

**Sentiment Data Integration:** Integrating movie review sentiments from Feature 1 (Review Generate Sentiment) into the application's existing sentiment-labelled dataset. This is to enrich the sentiment analysis with more comprehensive data.

**Enhanced Filtering Mechanism:** Refining the results from features 2 (Aspect-based Review), 3 (Movie-based Review Sentiment), and 4 (Recommendation) by analyzing the correlation of actors, directors, and writers to the positive sentiment derived from the current model's output.

# MECE Table

@Aura to update this part