Movie Rating Prediction using Movie Posters ,Textual Data and Leveraging Image Pretrained Models

Busra SEVINC, x22183957

Erkan UCI, x23210494

Umar Farooq Khan, x22179780

[1]National College of Ireland, Dublin, Ireland

Abstract—*This research delves into leveraging advanced machine learning techniques, including the use of pre-trained models like LSTM and VGG16, to predict movie ratings from poster images within the entertainment industry. Our approach harnesses the power of Python and AI to analyze visual content and extract features that correlate with movie ratings, integrating diverse data sources such as IMDb, Rotten Tomatoes, and Box Office Mojo. The work is set against the backdrop of an industry burgeoning in value, projected to reach USD 90.92 billion by 2021 and a CAGR of 7.2% from 2022 to 2030. By employing pre-trained models, we capitalize on deep learning for image analysis, comparing different architectures' effectiveness in rating prediction. The study maintains global coding standards and incorporates robust architecture, continuous integration, and delivery processes, with methodologies like XGBoost to sharpen predictive accuracy. Overall, this project not only aids stakeholders in strategic planning but also carves new avenues for enhancing predictive analytics in the entertainment sector.*

Keywords— *Image Analysis, Neural Network, Machine Learning, Artificial Intelligence, Movie Industry,, Predictive Analytic,*

1. Introduction

In the rapidly changing landscape of the entertainment industry, this project aims to forecast movie revenues and vote averages by leveraging the predictive capabilities of Python and AI. The research is motivated by the industry's significant growth prospects as well as the need for advanced analytics tools to navigate its complexities. According to Grand View Research, the global movies and entertainment market will be worth USD 90.92 billion in 2021 and will grow at a compound annual growth rate (CAGR) of 7.2% from 2022 to 2030[1]. This expansion, together with changes in consumer behavior and technological advancements, emphasizes the importance of developing sophisticated predictive models.

The project's goals are two fold: to investigate the factors that influence movie success and to build a predictive model that can provide accurate forecasts. This investigation is especially timely given the industry's changing landscape, which is characterized by the increasing popularity of streaming platforms and the impact of external factors such as the COVID-19 pandemic, which forced the closure of thousands of theaters and significantly influenced market dynamics[1].

Building on Almadi (2017)'s foundational work, which used IMDb data to investigate financial success predictors in the movie industry, this project will delve deeper into the nuances of movie success[2]. Almadi's research demonstrated the significance of parameters such as movie interconnectedness and director credentials in predicting a film's gross revenue, serving as a useful starting point for this study[2].

This research will address novel questions important to industry stakeholders, such as how emerging trends and the vast array of available data can be used to more effectively predict movie success. What new insights can be gained by combining traditional industry knowledge with advanced analytical techniques? The project's goal is to provide a comprehensive tool for industry stakeholders to make informed strategic decisions in an increasingly competitive market by answering these questions.

2. Related Work

Several studies in the field of predictive analytics for the film industry have provided foundational knowledge and methodologies that will guide the current project. Quader et al. (2017) used machine learning techniques to create a decision support system for the movie investment sector. By analyzing historical data from various sources such as IMDb, Rotten Tomatoes, Box Office Mojo, and Metacritic, their system predicted an approximate success rate of a movie based on its profitability. They discovered that budget, IMDb votes, and the number of screens all played an important role in predicting a movie's box-office success, achieving an accuracy of 84.1% for pre-released features and 89.27% for all features when one-way prediction is taken into account[3]. This paper presents a comprehensive approach to forecasting that makes use of a wide range of data sources and machine learning techniques.

Furthermore, Lash and Zhao created a decision support system to predict movie profitability using machine learning techniques, text mining, and social network analysis. They included features like dynamic network features, plot topic distributions, and star power measurement based on the profits of their films. Their hypothesis was that the more optimistic, positive, or excited audiences were about a film, the more likely it was to earn more money[4]. This research is critical in understanding the multifaceted aspects that contribute to a film's success that go beyond financial factors.

Using hype analysis, Sivasantoshreddy et al. attempted to predict a movie box-office opening gross. Their research was based on the hype factor, the number of theatres where the films were released, and the average price of all tickets per show. They used a web crawler to find the number of tweets about a movie and took into account factors such as the number of relevant tweets per second and the reach of a tweet. However, they did not use language processing to determine whether the tweet was positive or negative, which limits their ability to understand the sentiment behind the data[5]. This study focuses on the advantages and disadvantages of using social media data for predictive analysis.

Finally, a study examined the effectiveness of SVM and neural networks in prediction using detailed sentiment analysis of collected reviews. By experimenting with different numbers of hidden layers, they discovered that a three hidden layer Multi-Layer Perceptron (MLP) architecture consistently produces better results[6]. The study emphasizes the significance of model architecture in achieving accurate predictions, as well as the potential of sentiment analysis in understanding public reception. While these studies provide valuable insights and methodologies, they also have limitations. This project aims to expand on these methodologies, addressing limitations where possible, and providing a more holistic and accurate prediction of film success.

3. Methodology

3.1. Project Architecture

The study focused on crafting code to international standards, ensuring systematic efficiency, and was structured to support continuous integration and delivery. Key infrastructure elements like architecture design and database creation followed these protocols, with development managed via branches in a git repository, aligning with continuous deployment practices.

The basic layers that make up the project are:

1. Database Layer:. The database provides basic data storage and management of the project.
2. Module Layer: This layer contains the business logic and modular structure, includes the basic functions and modules of the project. This layer is designed in an orderly and modular manner for code readability and maintainability.
3. Test Layer: Test processes were managed through this layer in order to verify that the modules and functions in the project are working correctly.

During the development process of the project, when working on a specific feature or fixing a bug, branches were used to integrate these changes into the main code base. In this way, continuous development and improvement of the project was ensured.

git repository:

erkanuciDevelopment/MovieRecommedations at dev (github.com)

git clone:

https://github.com/erkanuciDevelopment/MovieRecommedations.git

```
Folder PATH listing for volume Windows
Volume serial number is 5AD7-DAC8
C:.
├───.vscode
├───src
│   ├───db
│   │   └───Mongodb
│   │       └───__pycache__
│   ├───module1
│   │   └───__pycache__
│   ├───module2
│   └───static
│       └───files
│           └───output_csv
└───test
```

The project includes a .bat file to facilitate its deployment as a Docker container, allowing for quick execution and remote setup, streamlining development and usage.

run_project.bat

MovieRecommedations > run_project.bat

```
1   @echo off
2
3   REM Set the path to the Python executable
4   set PYTHON_EXECUTABLE=python
5
6   REM Set the path to the main Python script
7   set MAIN_SCRIPT=src\main.py
8
9   REM Navigate to the project directory
10  cd /d %~dp0
11
12  REM Display a message
13  echo Running the project...
14
15  REM Run the Python script
16  %PYTHON_EXECUTABLE% %MAIN_SCRIPT%
17
18  REM Pause to keep the console window open
19  pause
20
```

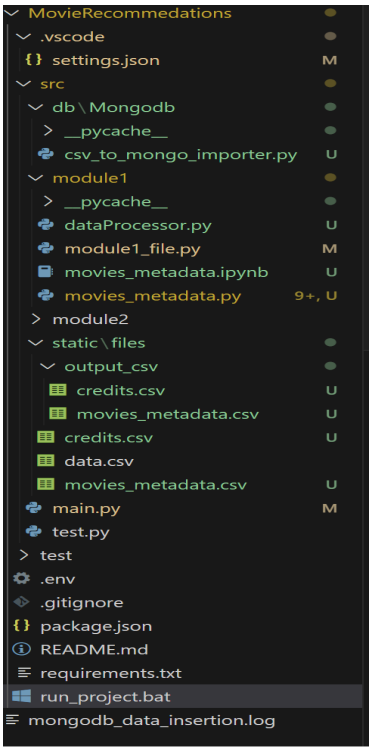
The settings.json file of our project is our JSON file containing database connections and port information.

setting.json

```
{
    "mongodb": {
        "credits": {
            "host": "localhost",
            "port": 27017,
            "database_name": "MoviesDb",
            "collection_name": "credits",
            "connection_string": "mongodb://localhost:27017/?ssl=true",
            "csv_file": "credits.csv"
        },
        "movies_metadata": {
            "host": "localhost",
            "port": 27017,
            "database_name": "MoviesDb",
            "collection_name": "movies_metadata",
            "connection_string": "mongodb://localhost:27017/?ssl=true",
            "csv_file": "movies_metadata.csv"
        }
    }
}
```


MongoDB database was used in the project and the basic structure of the project was built on this database. In the first testing phase, the project was run in the MongoDB cloud environment, but the main structure of the project continued through local MongoDB. Because of the MongoDB cloud service, I preferred to continue from the local environment due to certain certification requirements.

The basic logic of the project is as follows: we read credit.csv and movies_metadata.csv files and save this data as JSON on MongoDB. In the subsequent data processing stages, we pull our data from MongoDB and process it. These processes include model training, data visualization, and other data processing processes.



The main structure of the project was configured through the `main.py` file and this file was set in the `run_project.bat` file that we Dockerized before. We just need to run the ‘run_project.bat’ file to get the project up and running. This Dockerized file will both initialize MongoDB and initialize the project by calling other relevant methods.

In summary, the project was run by running the project through the `main.py' file and calling other classes through this main class. This process is followed by the Dockerized `run_project.bat` file. The project separates the contributions of its various layers with the environment and different libraries using the Python programming language.

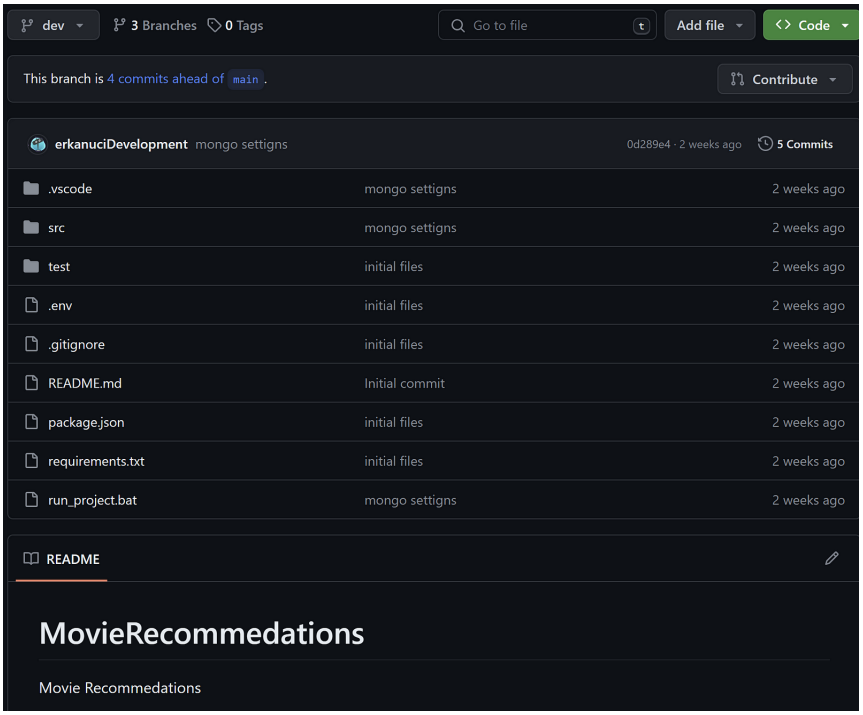
It is important that the project runs on MongoDB to reduce data loss and maintain JSON distribution in a more controlled manner. The project continued with all Git options.

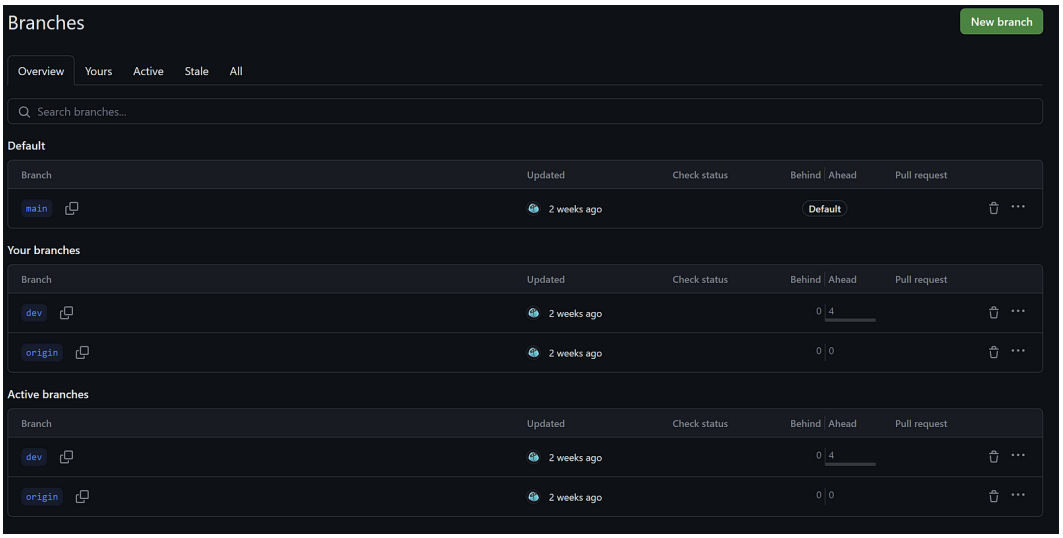
As seen below CsvToMongoImporter, DataProcessor, and MoviesmetaData classes were included in the project as a module, allowing the project to be managed through a single file. When we examined the code structure, the entire project tree was written according to a certain standard, and the entire system was managed through a docker container.

```python
# main.py

import os
from db.Mongodb.csv_to_mongo_importer import CsvToMongoImporter
from module1.dataProcessor import DataProcessor
import time
from module1.movies_metadata import MoviesmetaData

def main():
    # Print the current working directory
    print("Current working directory:", os.getcwd())

    importer = CsvToMongoImporter()
    csv_files_directory = os.path.join(os.getcwd(), "MovieRecommedations", "src", "static", "files")
    csv_files = ["credits.csv", "movies_metadata.csv"]  # Add more file names as needed
    importer.import_csv_to_mongodb(csv_files_directory, csv_files)

    # Add a wait time (e.g., 5 seconds)
    wait_time_seconds = 5
    print(f"Waiting for {wait_time_seconds} seconds...")
    time.sleep(wait_time_seconds)

    # Convert csv file to JSON, then send those data to MongoDB
    data_Processor = DataProcessor()
    data_Processor.process_mongodb_and_save_csv("MoviesDb")


    #call data processing
    movie_metadata=MoviesmetaData()
    movie_metadata.runModel()
if __name__ == "__main__":
    main()
```

branches

As seen below, work was done on 3 main branches. The main structure of the project was developed on the “dev” branch and then transferred to the “main” branch. Commits were made in accordance with the structure and the project process continued by monitoring via Git.





3.2. Getting Data and Detailed Description of Datasets

Data Loading: The information utilized in this study came from Kaggle.The metadata for each of the 45,000 films included in the Full MovieLens Dataset is contained in these files. The films in the dataset were released on July 1, 2017, or earlier.

First Dataset - Movie Metadata:

- Source: The 'movies_metadata.csv' file is the main dataset that is used, and it probably includes extensive information about movies.
- Contents: This dataset normally contains information such as movie titles, genres, budget, revenue, vote totals, vote averages, and release dates, though the specific contents aren't stated in the sample.
- Justification: This dataset was selected because it offers a wealth of data that is essential for forecasting movie receipts and vote totals. The data can be incorporated into the predictive models.

Second Dataset - Credit:

- Source: The 'credits.csv' file is the dataset that was used. Usually, this file is an accumulation of thorough movie credits.
- Contents: This contains details about the actors who portrayed the characters in the film, frequently including their name, the role they played, and perhaps even a special identification. Additionally, it offers details on the crew members, directors, and writers who work behind the scenes. It often lists their responsibilities and could have special identification codes.

- Justification: Incorporating 'credits.csv' into the prediction models is expected to enhance the feature set with qualitative production details, leading to more accurate predictions by analyzing data from both 'movies_metadata.csv' and 'credits.csv'.

Third Dataset - Image Dataset - Movie Posters:

- This dataset is a distinct and separate dataset containing movie posters. The images have names containing IMDb IDs. These images are the posters of the movies. The next step is to integrate the two datasets, combining movie textual data with its own poster data. This integration was done with careful consideration and logic to ensure that the textual information and movie posters were correctly matched and associated with the respective movies.
- Now the dataset which will be used for predicting rating by poster, is a combination of first and second dataset and Third dataset and has 7867 posters in the dataset

3.3. Data Processing Algorithms

The goal of the data processing operations is to create a reliable prediction model for vote averages and movie revenue. Every stage, from cleaning the data to evaluating the model, is well thought out and supported to guarantee precise and perceptive predictions. This methodical approach highlights the analysis's depth and rigor, making it a thorough study in predicting movie success metrics.

3.3.1. Revenue and Vote Average Predicting Model

Data Preprocessing:

- Data Cleaning: Making sure the dataset is accurate and complete for analysis by handling missing values with pandas functions and maybe SimpleImputer from sklearn.impute.
- Feature Scaling: Normalizing or standardizing the numerical features with MinMaxScaler ensures that each one contributes equally to the model's performance. This is essential because it guarantees that each feature contributes equally to the performance of the model and isn't skewed by its size.
- Categorical Encoding: By converting categorical variables into a machine-readable format with OneHotEncoder, you can improve the model's comprehension and utilization of this data.
- New Features: Numpy and pandas are used to create new features from the existing data in order to gain new insights and enhance the performance of the model.

A comprehensive examination of the dataset and its relationships is ensured by the application of a variety of feature engineering techniques and preprocessing methods.

Model Selection and Tuning:

- Models Used: Many models are taken into consideration, such as the following:Linear Regression, XGBRegressor, GradientBoostingRegressor, AdaBoostRegressor, RandomForestRegressor, Ridge, Lasso, ExtraTreesRegressor, and DecisionTreeRegressor. It is possible to investigate various kinds of relationships in the data thanks to this varied selection.
- Feature Importance: Important feature for each model is found for each model. This step is important because it improves model performance dramatically. After this step we observed 4 models stand out which are RandomForestRegressor, ExtraTreesRegressor, GradientBoostingRegressor, XGBRegressor
- Hyperparameter Tuning: Using hyperopt to optimize model parameters will guarantee optimal performance. We applied hyperparameter optimization on 4 models which are stated in Feature importance. In this way we minimize MSE value of models as much as possible.

Evaluation:

- Metrics: assessing the model's accuracy and consistency in revenue prediction using sklearn.metrics metrics like mean absolute error(MAE), mean square error(MSE),root mean square error(RMSE) and R^2 score.

Justification:

- Perceptive Assessment: By using a variety of metrics, one can get a comprehensive understanding of the model's performance and identify both its strong points and areas for development.
- Optimized Performance: The goal of incorporating scaling, encoding, feature importance and hyperparameter tuning is to minimize MSE and ultimately maximize the predictive accuracy and generalizability of the model.

3.3.2. Predicting with Movie Poster Model:

Image Data conversion to Numeric Form:

The dataset was preprocessed through several key steps:

Data Selection and Cleaning:

- Columns deemed relevant to the research, namely 'Id,' 'overview,' and 'vote_average,' were retained in the dataset. This selection is essential information for predicting movie ratings.
- Rows with missing values in the 'overview' column were removed from the dataset.
- An additional filter was applied to exclude movie plots with less than 2 sentences, enhancing the dataset's informativeness.
- Missing Values in ratings, were filled after careful consideration and experiments and median comes out to be giving more logical reasons.

Model 1: UNIMODAL - VGG16 Image Only Rating Prediction

- This model leverages VGG16, a deep convolutional neural network, known for its efficiency in image recognition tasks. VGG16 is used as a feature extractor for image data. The model's architecture is as follows:
  - Image Input Layer: Accepts images of a predefined size which is 350 pixels.
  - **VGG16 Base Model:** The layers of VGG16 are set to non-trainable, meaning the model uses the weights learned from the ImageNet dataset to extract features from the input images.
  - **Flattening Layer**: Converts the output from the VGG16 convolutional layers into a 1D feature vector.
  - **Dense and Regularization Layers:** A sequence of Dense layers with ReLU activation and Dropout for regularization. This section aims to learn the mapping from the extracted image features to the desired output.

- VGG16 base model is used for extracting complex features from images, which are then passed through additional layers to predict ratings linearly. The use of pre-trained weights from ImageNet allows the model to leverage a broad and rich understanding of visual features without the need for extensive training on image data.

Model 2: MULTIMODAL - CNN/LSTM - Combined Posters and Movie Plot Model

This model uniquely combines both posters and movie plots to predict ratings. It has two main components: an image processing pathway and a text processing pathway, which are as follows:

- Image Processing Pathway: Consists of Conv2D layers for feature extraction from images, followed by BatchNormalization and MaxPooling for dimensionality reduction and Dropout for regularization.
- Text Processing Pathway: Uses an Embedding layer to convert text data into dense vectors, followed by an LSTM layer to capture the sequential nature and dependencies within the text data.
- Feature Concatenation: The outputs of both pathways are concatenated to form a comprehensive feature vector that combines information from both image and text data.

The model captures the nuances of both visual and textual data. While the image pathway extracts visual features, the movie plot which is text pathway processes the descriptive information, and their combination is expected to offer a more holistic understanding for rating prediction.

Model 3: MULTIMODAL - VGG16 and LSTM Based Combined Model

This model also combines image and text data but leverages the VGG16 model for image processing, providing a more sophisticated approach to extracting image features. Its architecture includes:

- VGG16 Image Processing: Similar to Model 1, it uses the VGG16 model for deep feature extraction from images.
- Text Processing: Incorporates an Embedding layer followed by an LSTM layer, similar to Model 2, for handling text data.
- Combination of Features: The extracted image features and text features are merged, creating a comprehensive feature set that encapsulates both visual and textual information.

In this model, the strength lies in its ability to use VGG16's powerful image recognition capabilities in conjunction with LSTM's sequential data processing. We use this approach which is effective in scenarios where both visual and textual cues are important for making accurate predictions.

Model 4: UNIMODAL - LSTM - Text-Only Rating Prediction with GloVe Embedding

This model is focused exclusively on text data, utilizing GloVe, a pre-trained word embedding, for text representation. Its architecture is centered around:

- GloVe Embedding Layer: Transforms textual input into dense, meaningful vector representations using GloVe embeddings. This layer is set as non-trainable to retain the rich semantic information captured by GloVe embeddings.
- LSTM Layer: Processes the sequence of embedded words to capture the contextual relationships within the text.
- Dense and Regularization Layers: Similar to the previous models, these layers are used for further learning and regularization.

The GloVe embeddings provide an advanced starting point for understanding text data, capturing nuances in language that might be critical for accurate rating predictions. The LSTM layer's ability to understand sequence and context in text data complements the rich embeddings, making this model particularly effective for text-only scenarios.

All of these models represent a strategic choice in architecture to suit different aspects of rating prediction. The use of pre-trained models like VGG16 and GloVe provides a significant advantage in processing complex image and text data, respectively. The combined models showcase an innovative approach to leverage multimodal data, potentially offering a more comprehensive perspective for prediction tasks.

3.4. Environment Setup

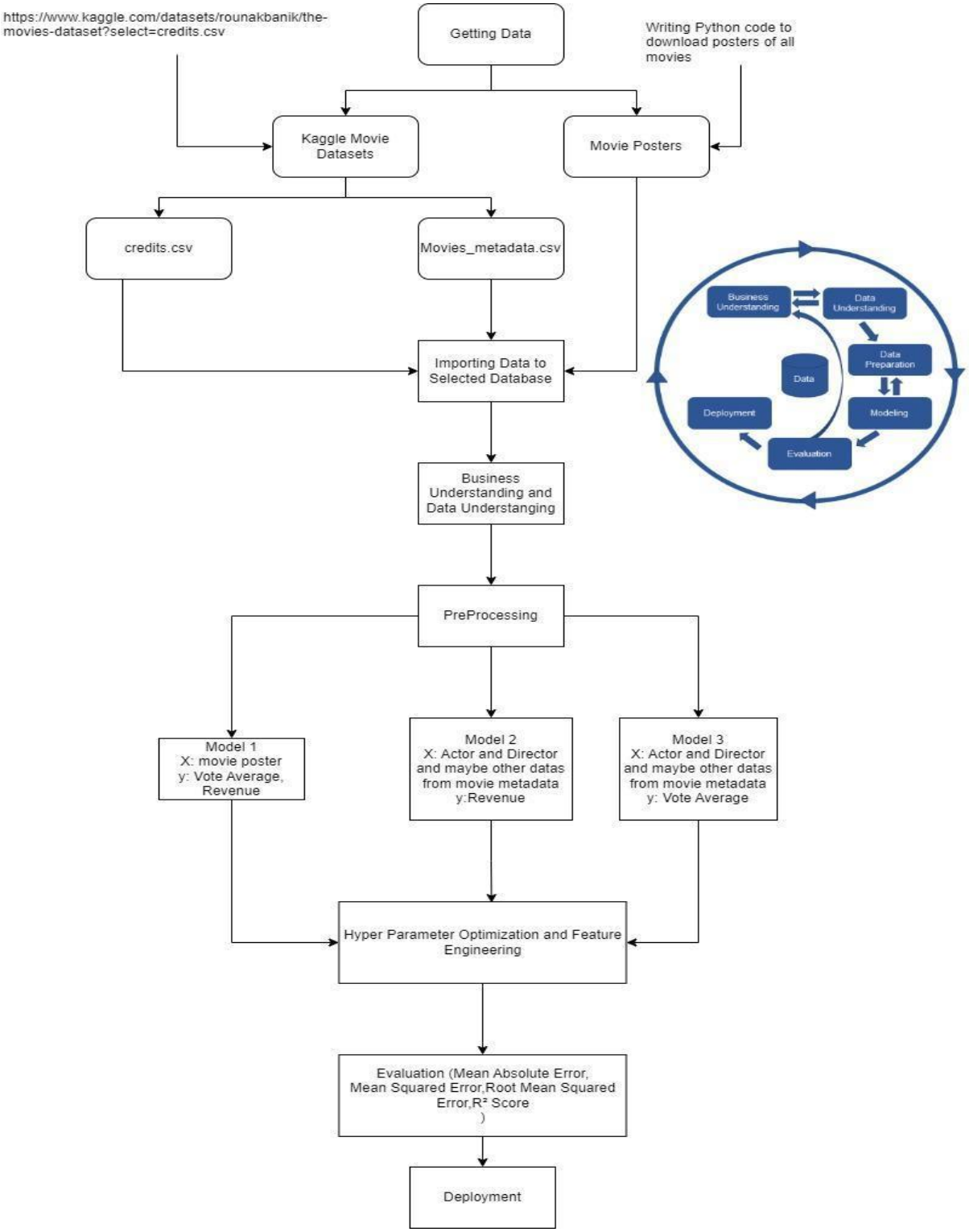
Overview

This study's predictive modeling and data analysis were carried out in a monitored Python environment. The results are guaranteed to be consistent and repeatable with this setup.

- Python 3.8
- Numpy (1.24.3)
- Pandas (2.1.1)
- Matplotlib (3.8.0)
- Seaborn (0.13.0)
- Scikit-Learn (1.3.1)
- XGBoost (2.0.3)
- Hyperopt (0.2.7)
- Tensorflow (2.14.0)
- TQDM (4.66.1)

3.5. Overview of Methodology

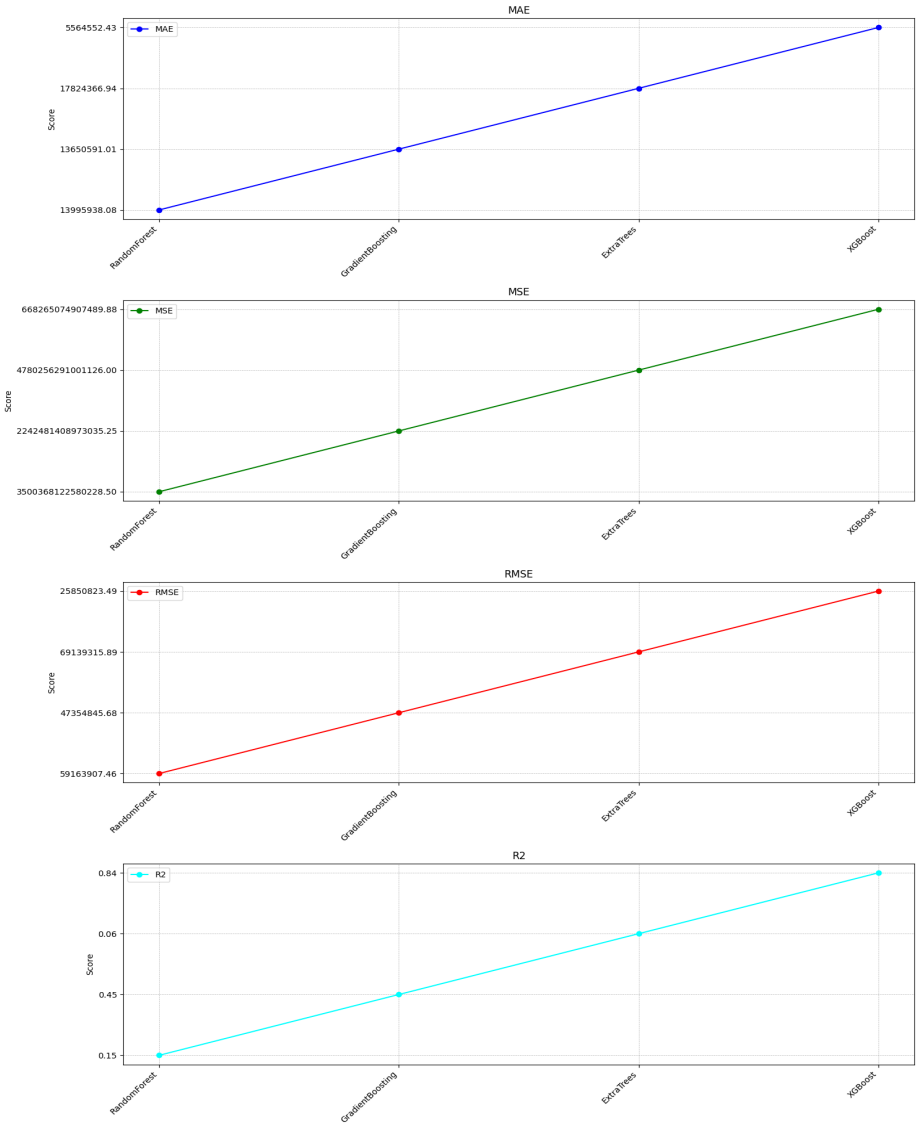
The summary of the methodology is as follows:



4. Result and Evaluation

4.1. Revenue Predicting Model Result

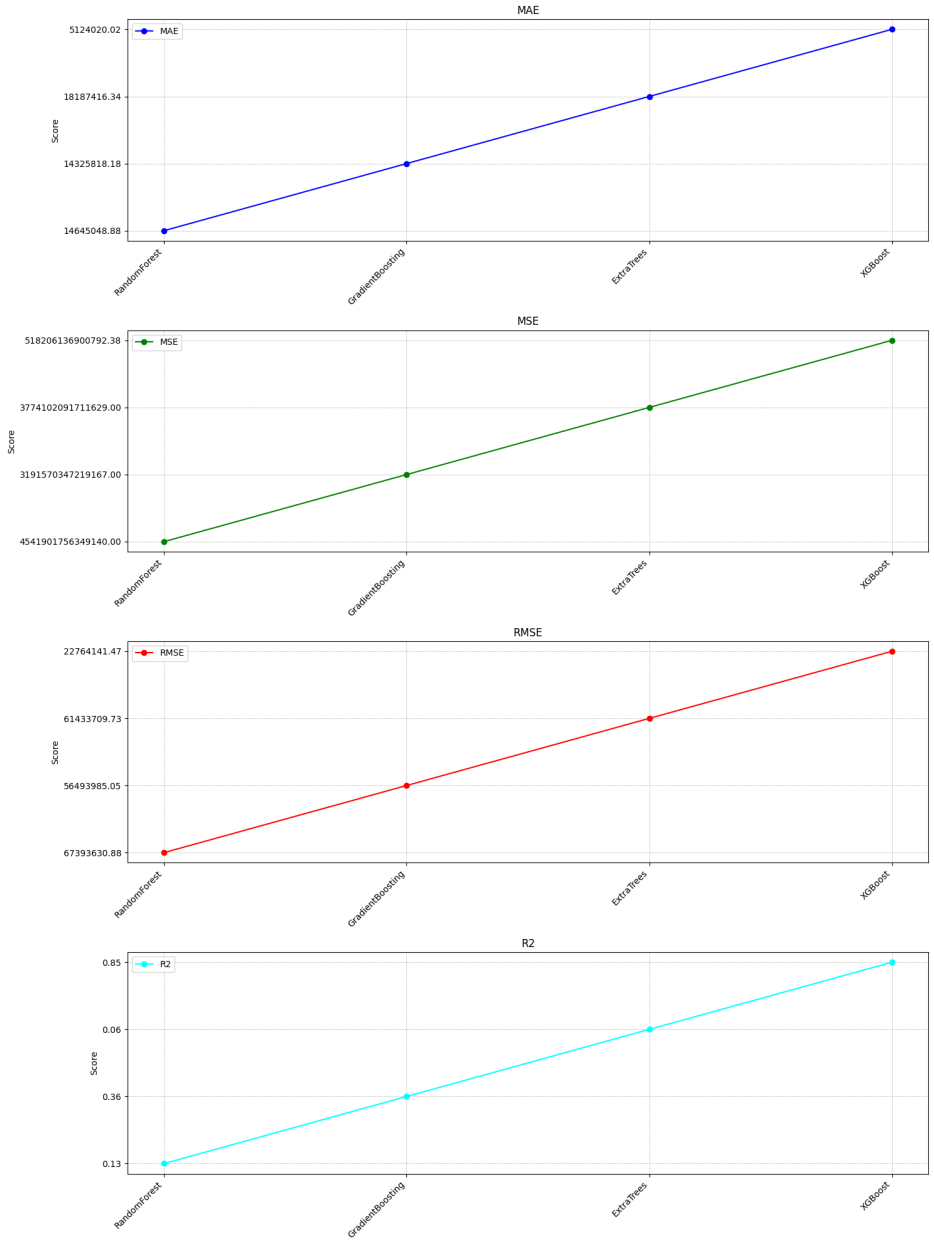
The performance metrics, including MAE, MSE, RMSE, and R2, reveal significant differences in the effectiveness of predictive models. The XGBoost model excels with the lowest MAE, MSE, and RMSE, demonstrating minimal deviation and error variance, and an impressive R2 score of 0.84, accounting for 84% of the dataset variance. In contrast, the ExtraTrees algorithm shows higher MAE and MSE, indicating less accuracy. Although GradientBoosting and RandomForest exhibit moderate errors, GradientBoosting has a higher R2, explaining more variance. These findings, illustrated graphically, underscore XGBoost's superior fit and accuracy for this dataset.



4.2. Vote Average Model Result

Model evaluations reveal varying performances among the four algorithms. XGBoost outperforms RandomForest, GradientBoosting, and ExtraTrees with lower MAE and RMSE values, showcasing superior

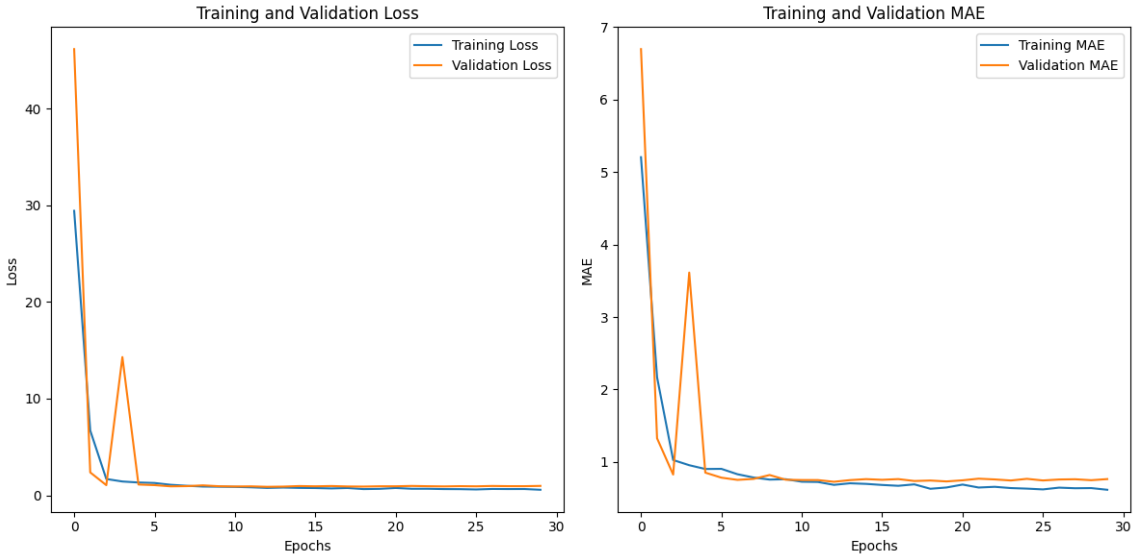
prediction accuracy. It also boasts a high R-squared (R2) value of 0.85, indicating an 85% explanation of data variance. GradientBoosting offers a reasonable balance between error metrics and R2 score. RandomForest and ExtraTrees exhibit lower predictive accuracy, with lower R2 scores and higher error metrics. Visualized in color-coded graphs, these metrics provide a clear comparison of model performance.



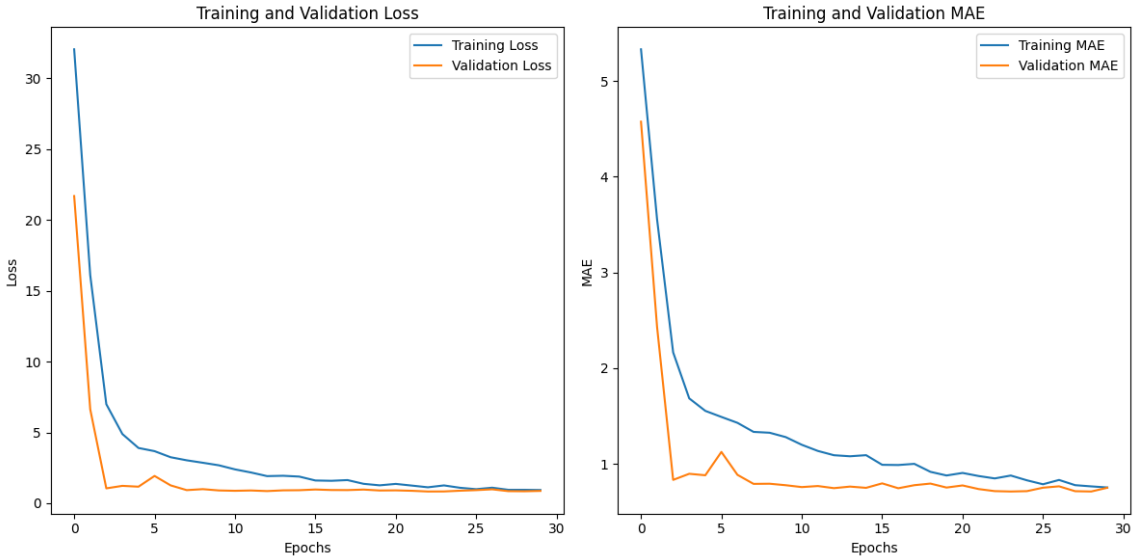
4.3. Predicting with Movie Poster Result

We can compare four different training scenarios of a machine learning model, examining how each model's loss and Mean Absolute Error (MAE) evolve over epochs to understand their learning behavior and generalization performance.

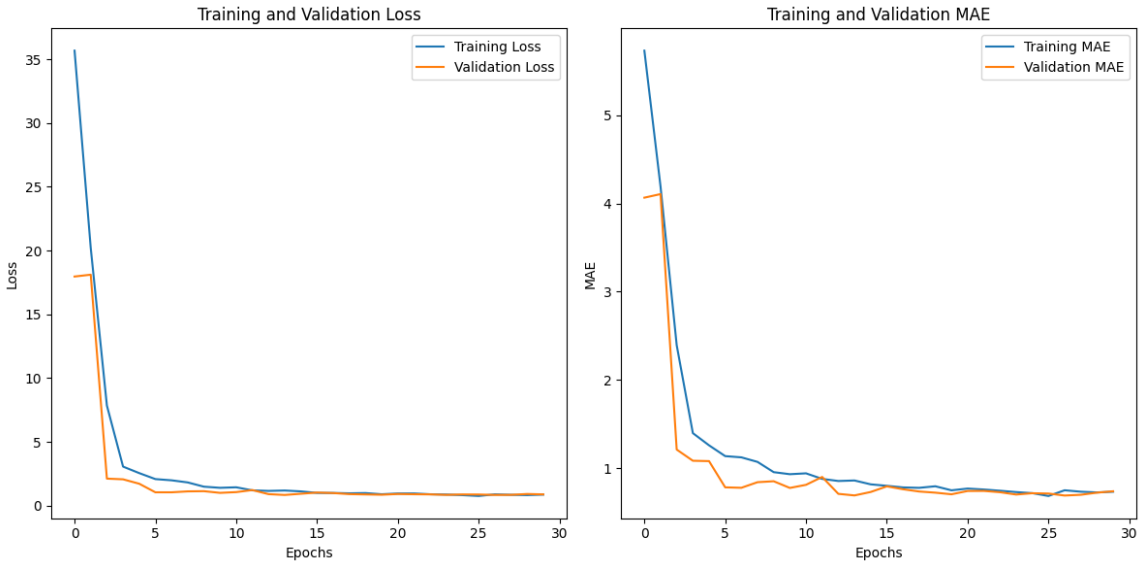
Each scenario presents a unique trend in the convergence of training and validation metrics, reflecting the models' varying degrees of efficiency in learning from the data.



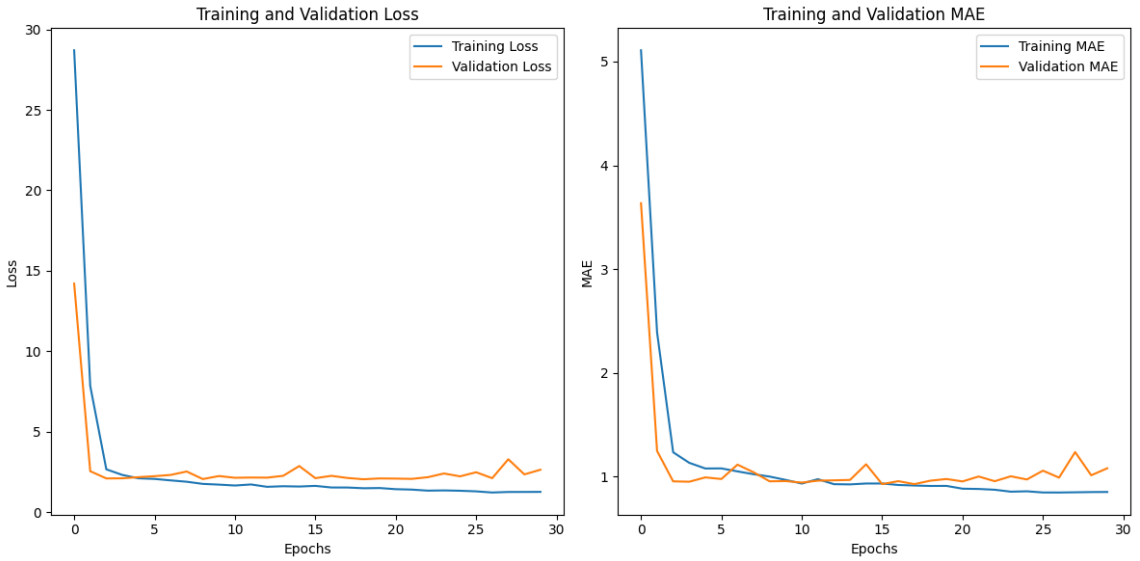
Model 1 showing a spike in early epochs



Model 2 showing a quick drop within first epochs



Model 3 showing stable convergence showing no sign of overfitting



Model 4 showing a gradual improvement in learning and generalizing

Models which are unimodal showed quick learning but struggled with generalization, indicated by higher validation MAEs suggesting overfitting. Only image using VGG16, had a significant initial loss drop, yet its higher validation MAE pointed to limited generalization. Unimodal using only Movie plots with Glove and LSTM validation MAE fluctuated greatly, questioning its stability.

Models which are multimodal having LSTM and VGG16 were more aligned in their training and validation MAEs, with Model 2 which is just by using our own image trained model exhibiting an atypically lower validation MAE, possibly due to a less complex validation set.

Model 3 which has the combination of pretrained VGG16 and LSTM stood out for its low and stable validation MAE, marking it as the most robust and generalizable model for practical use, especially for mixed data types.

Our third model's utilizing power of VGG16 and LSTM of a racing movie yielded a predicted rating with an impressively minimal difference of 0.14 from the actual current IMDB rating highlighting the model's exceptional accuracy.

Fig: Rating of racing movie from poster using VGG16 and LSTM

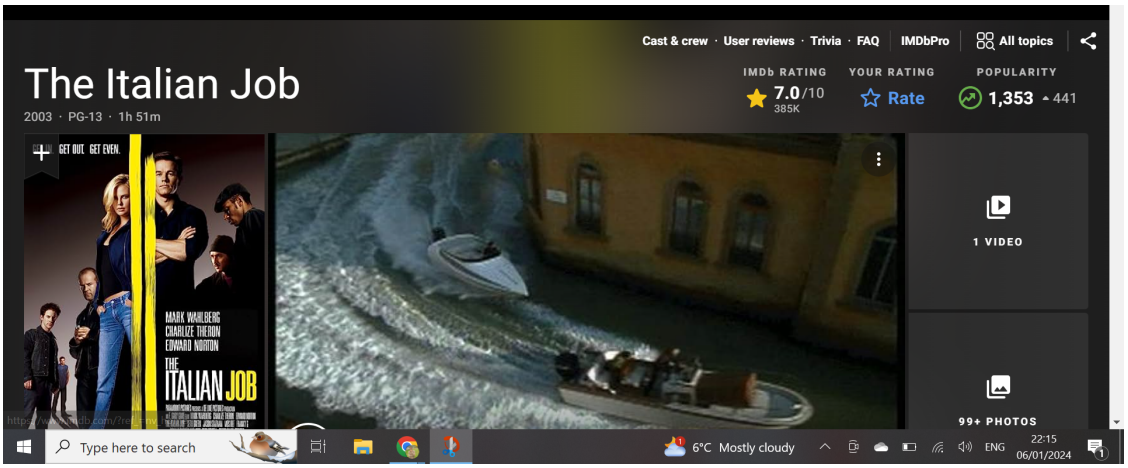


Fig: Actual IMDB page showing 7 Rating

Our fourth model's analysis of a 'Fast & Furious' plot synopsis yielded a predicted rating with an impressively minimal variance of 0.01 from the actual rating, highlighting the model's exceptional accuracy.

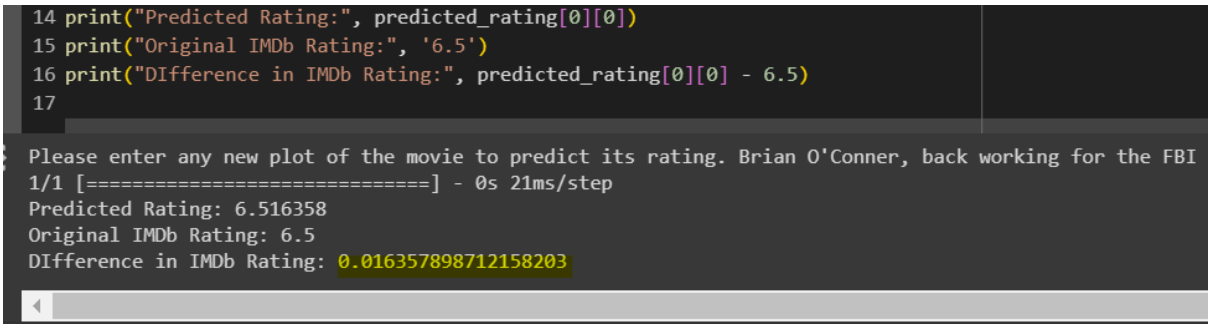


Fig: Prediction of rating from Movie Plot using Glove and LSTM

The research concluded with the model's deployment for inference. Given a new movie poster image and text data, the model predicted the IMDb rating by combining the visual and textual features. This prediction process demonstrated the model's real-world applicability which can predict movie rating solely on the movie poster which can help graphic designers and many other professionals.

5. Conclusions and Future Work

5.1. Conclusion

Using Python and advanced AI techniques using Image Analysis using Pre Trained and Self trained models. This project set out on an ambitious journey to forecast ratings, movie revenues in an ever-changing entertainment industry. We delved into the factors influencing movie success and developed a predictive model aimed at providing accurate forecasts through meticulous research and analysis. The research was founded on a thorough review of related literature, which shaped our methodological approach and informed the development of a robust and scalable project architecture. Our project was enriched by a variety of datasets, and we used sophisticated data processing algorithms to effectively clean, scale, and encode data, resulting in a predictive model with notable accuracy.

5.2. Future Work

Looking forward, there are ways to enhance and expand this project:

Utilize advanced pretrained models like BERT for natural language understanding and Convolutional Neural Networks (CNNs) such as ResNet or Inception for image analysis to improve performance.

Implement transfer learning by leveraging models pretrained on large datasets to enhance movie rating predictions. Combine these techniques with larger, diverse datasets and advanced ensemble methods for more accurate movie rating predictions.

Include additional datasets for a deeper understanding of film success, such as real-time social media sentiment, detailed audience demographics, and granular financial data. Explore emerging AI and machine learning techniques like deep learning and neural networks for deeper insights and improved accuracy. Enhance model interpretability to provide detailed insights into the factors impacting revenue and vote averages.Develop a system for real-time predictions with updated data, such as immediate box office returns and audience reactions. Create a user-friendly interface for industry stakeholders to interact with predictive models and customize inputs.

By building on this foundation and exploring these directions, we can advance predictive analytics in the entertainment industry.

6. Bibliography

[1] Grand View Research, "Movies And Entertainment Market Size Share & Trends Analysis Report By Product (Movies Music & Videos) By Region And Segment Forecasts 2022 - 2030" [Online]. Available: https://www.grandviewresearch.com/industry-analysis/movies-entertainment-market#. Accessed on: Jan. 3, 2024.

[2] K. Almadi, "Quantitative study of the movie industry based on IMDb data," Doctoral dissertation, Massachusetts Institute of Technology, 2017.

[3] N. Quader, M. O. Gani, D. Chaki, and M. H. Ali, "A machine learning approach to predict movie box-office success," in Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT), Dec. 2017, pp. 1-7.

[4] N. Quader, M. O. Gani, and D. Chaki, "Performance evaluation of seven machine learning classification techniques for movie box office success prediction," in Proc. 3rd Int. Conf. Electr. Inf. Commun. Technol. (EICT), Dec. 2017, pp. 1-6.

[5] R. Dhir and A. Raj, "Movie success prediction using machine learning algorithms and their comparison," in Proc. 2018 First Int. Conf. Secure Cyber Comput. Commun. (ICSCCC), Dec. 2018, pp. 385-390.

[6] T. Kim, J. Hong, and P. Kang, "Box office forecasting using machine learning algorithms based on SNS data," Int. J. Forecast., vol. 31, no. 2, pp. 364-390, 2015.