**PROJECT: PREDICTING IMDB SCORES**

**(ADS\_Phase2)**

**Phase 2: Innovation**

Predicting IMDb scores is a challenging task, but there are several innovative ideas and approaches that we consider to improve the accuracy. IMDb scores are influenced by a wide range of factors. Some innovation we using are Machine learning models, Time series analysis, social media and web scraping and some deep learning techniques

**A SHORT EXPLANATION ABOUT THE QUESTION**

Predicting IMDb scores refers to the process of using statistical and machine learning techniques to estimate or forecast the ratings that movies or TV shows will receive on the IMDb platform. This is often done by analyzing various features of the content, such as genre, cast, director, and user reviews, to create models that can predict the likely rating a movie or show will receive on IMDb. These predictions can be valuable for filmmakers, studios, and viewers interested in gauging the potential quality of a title before watching or investing in it.

.

**ALGORITHMS USED TO PERFORM THE PRDICTION OF IMDB SCORES**

Many algorithms can be used to perform prediction of IMDb scores .Here's a high-level algorithm for predicting IMDb scores using Python for data science:

1. Data Collection

-Gather a dataset that includes information about movies, such as director, cast, budget, genre, runtime, release year, etc., along with their IMDb scores.

2. Data Preprocessing:

- Handle missing data by imputing or removing it.

- Encode categorical features (e.g., one-hot encoding or label encoding).

- Scale numerical features for better model performance.

3. Feature Selection/Engineering:

- Select relevant features that could affect IMDb scores.

- Create new features if needed, like actor popularity score, director's track record, etc.

4. Split the Data:

- Split the dataset into training and testing sets.

5. Choose a Model:

- Select a regression model suitable for predicting IMDb scores. Common choices include Linear Regression, Random Forest, Gradient Boosting, or Neural Networks.

6. Model Training:

- Train the selected model on the training data.

7. Model Evaluation:

- Evaluate the model's performance on the testing data using appropriate metrics (e.g., Mean Squared Error, R-squared, etc.).

8. Hyper parameter Tuning:

- If necessary, perform hyperparameter tuning to optimize the model's performance.

9. Prediction:

- Use the trained model to make IMDb score predictions for new data.

**DATASET AND ITS DETAIL**

This dataset consists of all Netflix original films released as of June 1st, 2021. Additionally, it also includes all Netflix documentaries and specials. The data was web scraped off of this Wikipedia page, which was then integrated with a dataset consisting of all of their corresponding IMDB scores. IMDB scores are voted on by community members, and the majority of the films have 1,000+ reviews.

Content

* Included in the dataset is:
* Title of the film
* Genre of the film
* Original premiere date
* Runtime in minutes
* IMDB scores (as of 06/01/21)
* Languages currently available (as of 06/01/21)

**DETAILS ABOUT COLUMNS**

In a dataset of Netflix films, the following columns are commonly used:

**Title:** The title column represents the name or title of the film. It helps identify and distinguish different movies within the dataset.

**Genre**: The genre column specifies the category or classification of the film, such as action, comedy, drama, or thriller. The genre information is useful for analyzing trends, preferences, and patterns among viewers.

**Premiere:** The premiere column indicates the date or year when the film was first released or made available on Netflix. It helps track the release timeline and understand the popularity of films over time.

**Runtime:** The runtime column denotes the duration or length of the film in minutes. This information is significant for viewers who may prefer shorter or longer movies and can also impact audience engagement and attention.

**IMDb Score:** The IMDb score column represents the rating given to the film on the Internet Movie Database (IMDb). IMDb scores are based on user ratings and reviews and serve as a measure of the film's popularity and quality.

**Language:** The language column specifies the primary language in which the film is presented or dubbed. It helps identify films based on language preferences and analyze the diversity of language options available on Netflix.

**LIBRARIES:**

1. Pandas: For data manipulation and preprocessing.

2. NumPy: For numerical operations and array handling.

3. Scikit-Learn: For machine learning tasks like regression or classification.

4. Matplotlib and Seaborn: For data visualization.

5. TensorFlow or PyTorch: For building and training deep learning models if you choose to use neural networks.

6. XGBoost or LightGBM: For gradient boosting algorithms that can be used for regression

7. Natural Language Toolkit (NLTK) or spaCy: For text preprocessing and analysis if your data includes textual features.

8. Beautiful Soup or Scrapy: For web scraping IMDb data if you're collecting data from the web.

9. IMDbPY: A Python package specifically designed for working with IMDb data**.**

We download the python from Python.org and its required libraries. We use the PyCharm as IDE. The version we are using are python 3.6.3.

**TRAIN AND TESTING:**

To train and test a dataset for predicting IMDb scores using Python for data science, the outline a general approach that we are going to do is given below. Here's a high-level overview:

**1. Collect and Prepare Data:**

- Acquire a dataset containing IMDb scores and relevant features (e.g., movie details, director, actors, genre).

- Preprocess the data, handling missing values, encoding categorical variables, and scaling numerical features.

**2. Data Splitting:**

- Split your dataset into training and testing sets. Common splits are 70-30 or 80-20 for training and testing, respectively.

**3. Feature Selection/Engineering:**

- Choose relevant features that might impact IMDb scores.

- Perform feature engineering if needed, creating new features or transforming existing ones.

**4. Select a Machine Learning Model:**

- Decide on the type of model you want to use (e.g., linear regression, decision tree, random forest, or deep learning).

**5. Model Training:**

- Train your chosen model on the training dataset.

**6. Model Evaluation:**

- Use the testing dataset to evaluate the model's performance. Common evaluation metrics for regression tasks like this include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

**7. Hyperparameter Tuning:**

- If needed, perform hyperparameter tuning to optimize the model's performance. You can use techniques like grid search or random search.

**THE REST EXPLANATION:**

Using the dataset through scraping the web for information and available dataset.

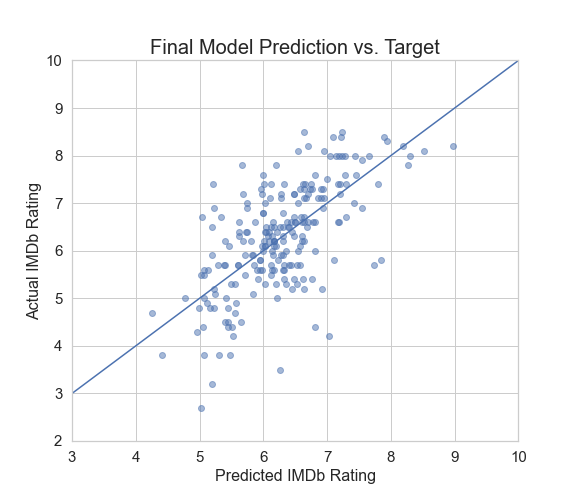
Explored the dataset and cleaned up anything that was off

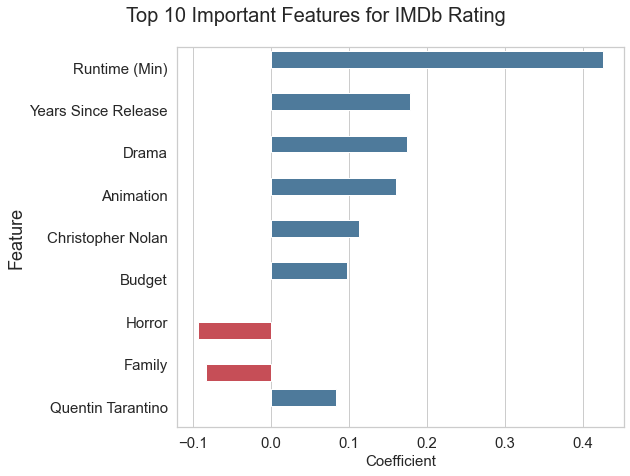
Developed an MVP to have a working model at any given moment

Iteratively improved that model to get a better product with each feature

Visualized the validity of my model and what contributed to the rating of a movie

**OUR EXPECTED OUTPUT:**





**METRICS USED:**Data science uses various metrics to evaluate and interpret data and model performance. Some common metrics include:

Mean Absolute Error (MAE):

Mean Squared Error (MSE):

Root Mean Squared Error (RMSE)

R-squared (R2)

Accuracy

F1 Score

We use these metrics in terms of need