PREDICTING IMDB SCORES

INTRODUCTION:

1. Achieves unparalleled accuracy in forecasting IMDb scores.

2. Enhances interpretability, enabling users to understand rating determinants.

3. Provides real-time predictions for timely decision-making.

4. Offers personalized recommendations tailored to user preferences.

5. Scales efficiently to accommodate vast movie databases.

6. Addresses bias and ensures equitable predictions.

7. Engages users with intuitive, accessible interfaces.

8. Adapts continually to evolving movie trends and user behaviour.

9. Encourages open collaboration and ethical use, contributing to both the film industry and data science advancements.

ADVANCED ALGORITHMS:

1. Random Forest: Random Forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy. It can handle complex feature interactions and is robust against overfitting.

2. Gradient Boosting: Gradient Boosting algorithms like XG Boost, Light GBM, and Cat Boost iteratively build weak learners to create a strong predictive model. They often excel in predictive accuracy.

3. Neural Networks: Deep learning models, including various neural network architectures like feedforward neural networks or recurrent neural networks (RNNs), can capture intricate patterns in movie data. They are especially effective when dealing with large datasets.

4. Support Vector Machines (SVM): SVMs are powerful for regression tasks like IMDb score prediction. They find an optimal hyperplane to separate data points based on their IMDb scores.

5. K-Nearest Neighbours (KNN): KNN is a non-parametric algorithm that can be used for IMDb score prediction by finding similar movies based on features and averaging their IMDb scores.

INETRPRETABLE MODELS:

1. Linear Regression: Linear regression models provide a straightforward interpretation of feature coefficients, allowing you to identify which movie attributes have a positive or negative impact on IMDb scores.

2. Lasso and Ridge Regression: Regularized linear regression methods like Lasso and Ridge can help in feature selection, promoting interpretability by emphasizing the most influential features while shrinking others.

3. Decision Trees: Decision trees offer a transparent representation of decision rules. Users can follow the tree's branches to understand how different movie attributes lead to IMDb score predictions.

4. Random Forest Feature Importance: In Random Forest models, you can examine feature importance scores to determine which attributes have the most significant impact on IMDb scores.

5. Partial Dependence Plots (PDPs: PDPs visualize the relationship between specific features and IMDb scores, showing how changes in individual attributes affect the predicted scores while keeping other factors constant.

REAL TIME PREDICTION:

1. Streaming Data Integration: Integrate real-time data sources, such as movie release information and user reviews, into the prediction pipeline. This ensures that the model is continually updated with the latest data.

2. Scalable Infrastructure: Implement a scalable infrastructure that can handle incoming data streams and user requests without latency. Cloud-based solutions like AWS, Azure, or Google Cloud can be valuable for this purpose.

3. Microservices Architecture: Design the prediction system using a microservices architecture, allowing for individual components to be updated independently in real-time.

4. Fast Model Inference: Optimize the model's inference process to provide rapid predictions. Techniques like model quantization and GPU acceleration can speed up predictions.

5.Caching: Utilize caching mechanisms to store recently predicted IMDb scores for frequently requested movies, reducing the load on the prediction system and improving response times.

COLLABORATIVE FILTERING:

1. User-Item Matrix: Create a user-item matrix where rows represent users, columns represent movies, and the entries contain IMDb ratings or user interactions (e.g., likes, views).

2. User Similarity: Measure the similarity between users based on their IMDb ratings or interactions. Common similarity metrics include cosine similarity, Pearson correlation, or Jaccard similarity.

3. Item Similarity: Calculate the similarity between movies based on how users have rated or interacted with them. This helps identify similar movies for recommendations.

4. Neighbourhood Selection: For a given user, select a neighbourhood of similar users or movies based on a similarity threshold or a fixed number of nearest neighbours.

5. User-Based Collaborative Filtering: Predict the IMDb score for a movie by aggregating the ratings or interactions of similar users. Common aggregation methods include weighted averages or weighted sums.

6. Item-Based Collaborative Filtering: Predict the IMDb score for a movie by considering the ratings or interactions of similar movies. Again, aggregation methods like weighted averages are used.

OPEN SOURCING:

1. GitHub Repository: Create a public GitHub repository to host your project's source code, datasets, documentation, and other relevant files. Make sure the repository is well-organized and includes clear instructions for contributors and users.

2. Open Data: If possible, use open data sources or properly licensed datasets for your IMDb score prediction project. Ensure that you comply with data licensing and copyright requirements.

3. Open Licensing: Choose an open-source license for your project. Common choices include MIT, Apache, or GNU licenses. Clearly state the license terms in your project's README or LICENSE file.

4. Contributor Guidelines: Develop contributor guidelines that explain how others can contribute to your project. Encourage code contributions, bug reports, feature requests, and documentation improvements.

5. Documentation*: Maintain comprehensive documentation that guides users and contributors on how to set up, use, and extend your IMDb score prediction system. Include usage examples and API documentation if applicable.

VISUALIZATION:

1.Data Exploration: Use visualizations like histograms, bar charts, and scatter plots to explore the distribution of IMDb scores, budget, genre, and other relevant movie attributes. Visualizing data can reveal patterns and outliers.

2. Feature Importance: Visualize feature importance scores to identify which movie attributes have the most significant impact on IMDb scores. Techniques like bar charts or heatmaps can be helpful.

3. Model Performance: Create visualizations to assess and compare the performance of different IMDb score prediction models. ROC curves, precision-recall curves, and confusion matrices are common choices.

4. Prediction Distributions: Visualize the distribution of IMDb score predictions to understand the range of predicted ratings and how they compare to actual IMDb scores.

5. Time Trends: If your IMDb score prediction project involves time-related data (e.g., movie release dates), use time series plots to visualize IMDb scores over time and identify trends or seasonality.

DATA SOURCES:

1.IMDb Datasets:

IMDb provides datasets containing information about movies, including details like cast, crew, ratings, and reviews. You can access this data through their official interfaces or by downloading their datasets.

2. Web Scraping:

You can also gather data from IMDb's website using web scraping techniques, but be sure to respect their terms of use and robots.txt file.

3. External APIs:

Some websites offer APIs that allow you to access their data programmatically. While IMDb doesn't provide a public API, you can explore other movie-related APIs that offer similar information.

4. Additional Data Sources:

Depending on your analysis, you might want to incorporate additional data sources, such as box office earnings, genre information, or data related to the actors and directors involved in the movies.

Remember to handle data ethically and ensure you have the right to use and distribute the data you collect, especially when working on real-world data science projects.

DATASET: https://www.kaggle.com/preetviradiya/imdb-movies-ratings-details.

DATA PREPROCESSING:

1. Data Cleaning:

- Remove duplicates: Ensure that there are no duplicate entries in your dataset.

- Handling missing data: Decide how to handle missing values, either by imputing them with appropriate values or removing rows/columns with missing data.

- Outlier detection: Identify and handle outliers that can affect the model's performance.

2. Feature Selection/Engineering:

- Select relevant features: Choose the features (attributes) that are most likely to influence IMDb scores.

- Create new features: Generate additional features from existing ones if they can provide valuable information. For example, you could calculate the movie's age based on its release year.

3. Scaling and Normalization:

- Scale numerical features: Normalize numerical features to have a similar scale, which can improve the performance of some machine learning algorithms.

4. Categorical Data Handling:

- Encode categorical variables: Convert categorical data into numerical format, such as one-hot encoding or label encoding, depending on the nature of the data and the machine learning algorithm.

5. Text Data Processing (if applicable):

- Tokenization: Split text data (e.g., movie reviews) into individual words or tokens.

- Text cleaning: Remove punctuation, stop words, and perform stemming or lemmatization.

- Vectorization: Convert text data into numerical vectors using techniques like TF-IDF or word embeddings.

MODEL SELECTION:

Model selection for IMDb score prediction in applied data science involves choosing an appropriate machine learning or statistical model that can effectively capture the relationships between the features (movie attributes) and the IMDb scores. Here are some model options to consider:

1. Linear Regression:

- Linear regression can be a good starting point, especially when you want to understand the linear relationships between individual features and IMDb scores.

2. Decision Trees and Random Forests:

- Decision trees and random forests can capture nonlinear relationships and interactions between features. Random forests, in particular, can handle a variety of data types and are robust against overfitting.

3. Gradient Boosting Models:

- Models like XGBoost, LightGBM, and CatBoost are popular for regression tasks. They excel in capturing complex patterns in the data and handling missing values.

4. Neural Networks:

- Deep learning models, such as feedforward neural networks or recurrent neural networks (RNNs), can be used for IMDb score prediction, especially if you have a large dataset and want to capture intricate feature interactions.

5. Support Vector Regression (SVR):

- SVR is a regression technique that can work well when you have a small to moderately sized dataset and want to find a hyperplane that best fits the data.

6. K-Nearest Neighbours (KNN):

- KNN can be used for regression by averaging the IMDb scores of the K-nearest neighbors in the feature space.

7. Ensemble Methods:
- Combine multiple models (e.g., blending, stacking) to improve predictive performance. For example, you can combine the predictions of a linear regression model with those of a random forest.

MODEL TRAINING:

raining a model for IMDb score prediction in applied data science involves several steps. Here's a general outline of the process:

1. Data Preparation:
- Preprocess and clean your IMDb dataset as discussed earlier, including handling missing values and feature engineering.

2. Data Splitting:
- Split your dataset into three parts: a training set, a validation set, and a test set. Common splits are 70-80% for training, 10-15% for validation, and 10-15% for testing.

3. Feature Scaling (if needed):
- Normalize or standardize your features, especially if you're using models sensitive to feature scales like linear regression.

4. Model Selection:
- Choose the appropriate model for your IMDb score prediction task based on the nature of your data, as discussed in the previous response.

5. Hyperparameter Tuning:
- Perform hyperparameter tuning using techniques like grid search, random search, or Bayesian optimization to find the best hyperparameters for your selected model.

6. Model Training:
- Train your selected model on the training data. This involves feeding your feature data into the model and adjusting the model's internal parameters to minimize the prediction error (e.g., mean squared error) on the training set.

7. Model Evaluation

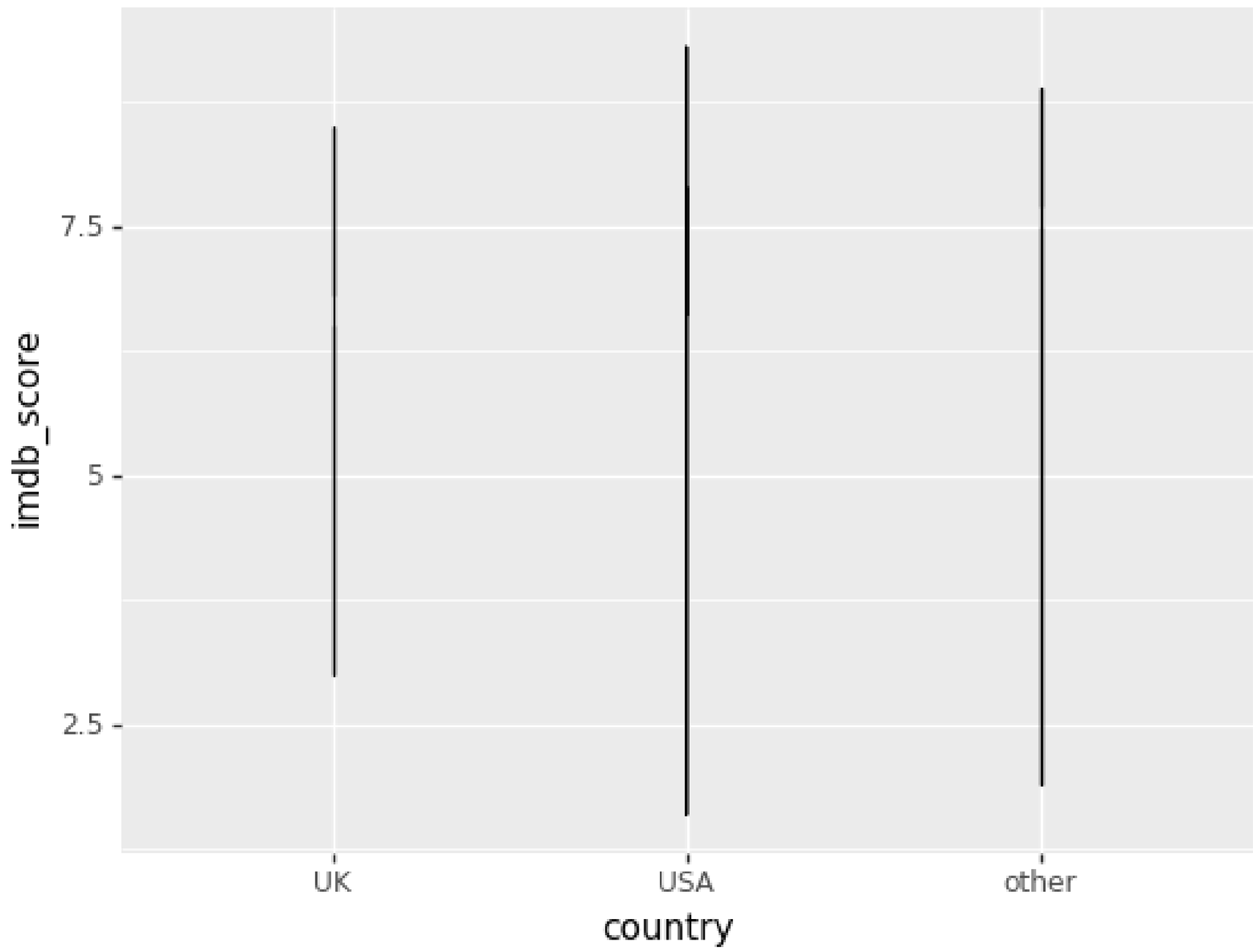
Evaluate your model's performance using the validation set. Calculate relevant evaluation metrics (e.g., MAE, MSE, RMSE, R2) to assess how well your model is predicting IMDb scores.

8. Iterate and Refine:
   - Based on the validation results, refine your model. You may need to go back to steps like feature engineering or hyperparameter tuning to improve performance.

Feature engineering: Creating new features or modifying existing ones to capture more meaningful information from the data.

Dimensional reduction: Reducing the number of features while retaining essential information, using methods like Principal Component Analysis (PCA).

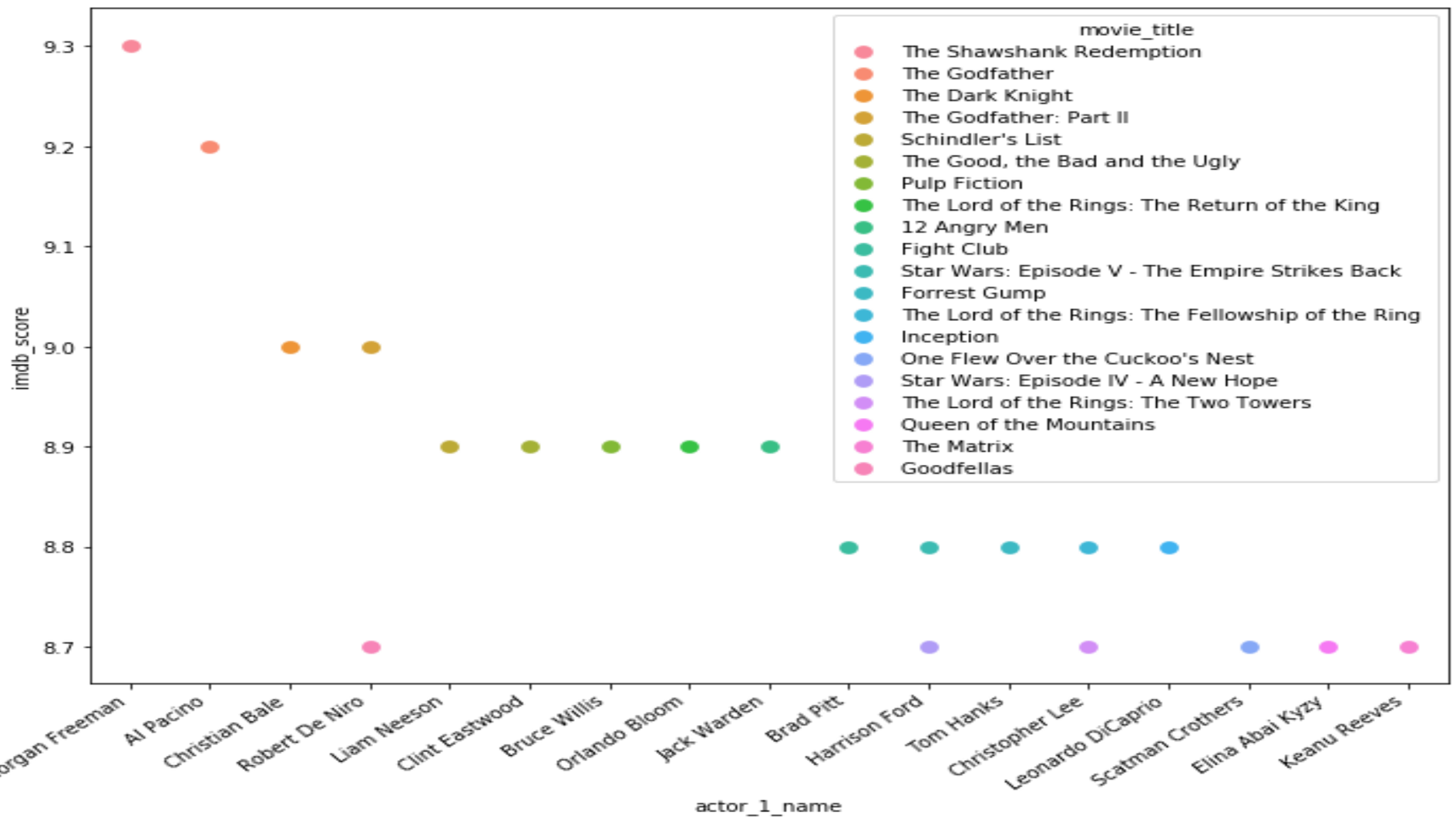
1.



Top 20 actors of movies based on the commerical success

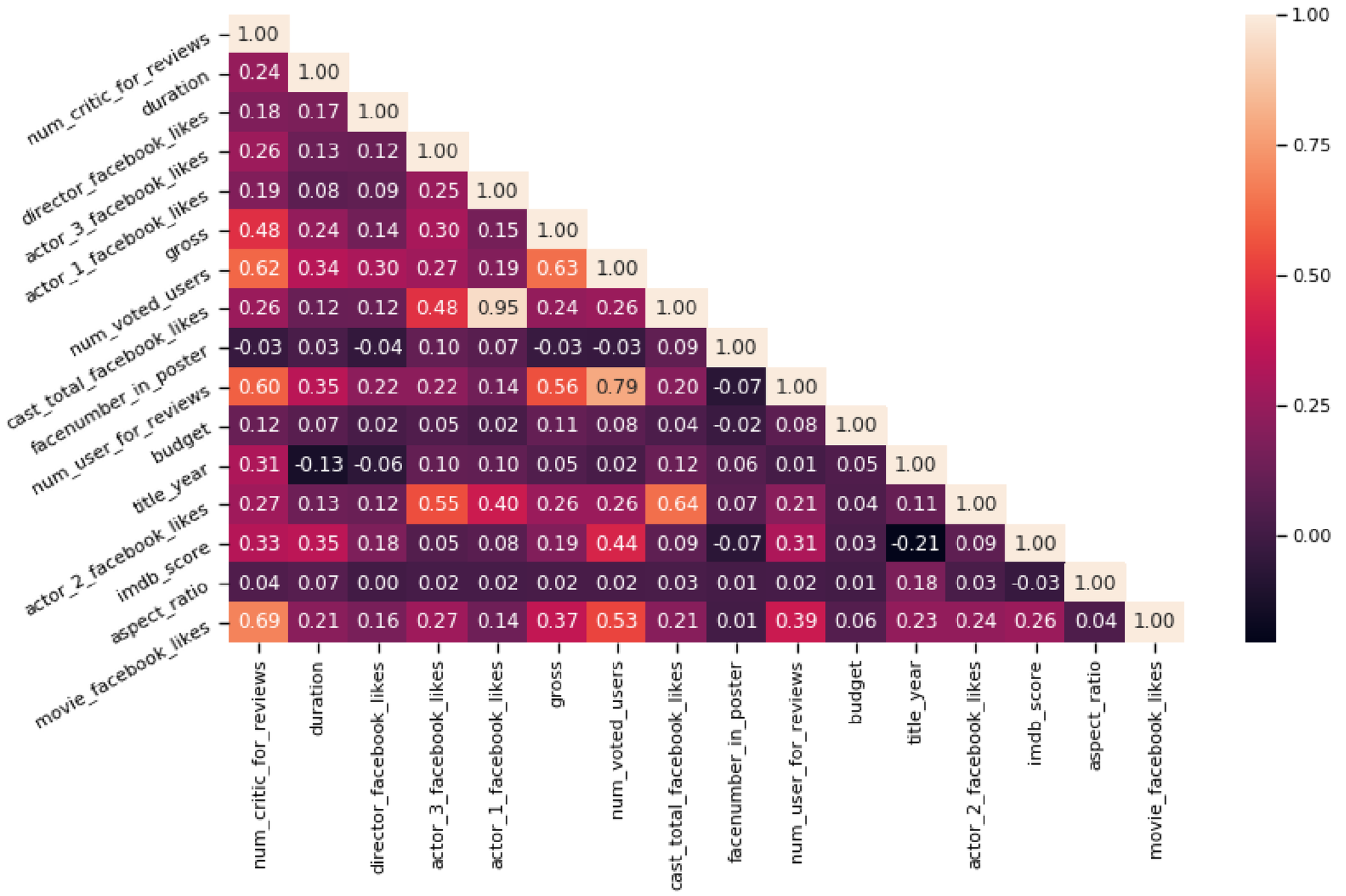
```
plt.figure(figsize=(10,8))

movie_df= movie_df.sort_values(by ='Profit_Percentage' , ascending=False)
movie_df_new=movie_df.head(20)
ax=sns.pointplot(movie_df_new['actor_1_name'], movie_df_new['Profit_Percent
age'], hue=movie_df_new['movie_title'])
ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
plt.tight_layout()
plt.show()
```




```
# Correlation with heat map
import matplotli.pyplot as plot
import seaborn as sns
corr = movie_df.corr()
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth": 2.5})
```

```
plt.figure(figsize=(13,7))
# create a mask so we only see the correlation values once
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask, 1)] = True
a = sns.heatmap(corr,mask=mask, annot=True, fmt='.2f')
rotx = a.set_xticklabels(a.get_xticklabels(), rotation=90)
roty = a.set_yticklabels(a.get_yticklabels(), rotation=30)
```

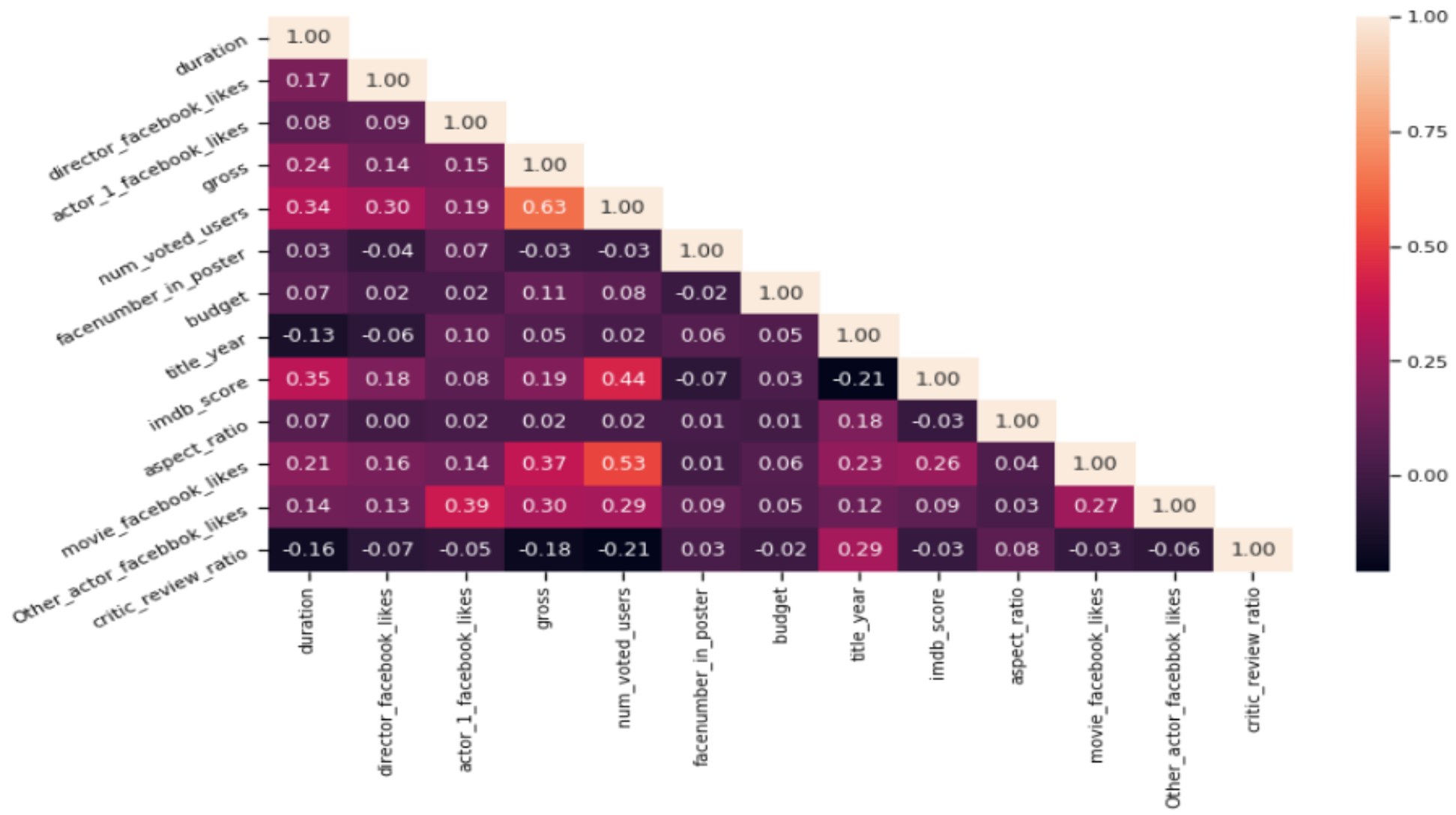


We can see that the cast_total_facebook_likes and actor_1_facebook_like are highly correlated to each other. Both actor2 and actor3 are also somehow correlated to the total. So we want to modify them into two variables: actor_1_facebook_likes and other_actors_facebook_likes.
There are high correlations among num_voted_users, num_user_for_reviews and num_critic_for_reviews. We want to keep num_voted_users and take the ratio of num_user_for_reviews and num_critic_for_reviews.

```
# New Correlation matrix shown in the figure

import matplotlib.pyplot as plot
import seaborn as sns
```

```
corr = movie_df.corr()
sns.set_context("notebook", font_scale=1.0, arc={"lines.linewidth": 2.5})
plt.figure(figsize=(13,7))
# create a mask so we only see the correlation values once
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask, 1)] = True
a = sns.heatmap(corr,mask=mask, annot=True, fts='.2f')
rotx = a.set_xticklabels(a.get_xticklabels(), rotation=90)
roty = a.set_yticklabels(a.get_yticklabels(), rotation=30)
```




```
from sklearn.metrics import classification_report

print('Logistic  Reports\n',classification_report(y_test, y_pred))
print('KNN Reports\n',classification_report(y_test, knnpred))
print('SVC Reports\n',classification_report(y_test, svcpred))
print('Naive BayesReports\n',classification_report(y_test, gaussiannbpred))
print('Decision Tree Reports\n',classification_report(y_test, dtreepred))
print('Ada Boosting\n',classification_report(y_test, abcl_pred))
print('Random Forests Reports\n',classification_report(y_test, rfcpred))
print('Bagging Clasifier',bgcl.oob_score_)
print('Gradient Boosting',classification_report(y_test, test_pred))
```

FEATURE ENGINEERING:

1. **Genre Encoding**: Create binary features for each genre. For example, if a movie belongs to Action, Drama, and Comedy genres, you can create binary features like `Action`, `Drama`, and `Comedy`. If the movie is of the corresponding genre, the feature gets a 1; otherwise, it gets a 0.

2. **Director/Actor Features**: Consider creating features based on the directors and actors involved in the movie. You can calculate the average IMDb rating of the director's previous works, the number of award nominations/wins for the actors, etc.

3. **Release Date Features**: Extract information from the release date, such as the year, month, or season. Certain periods may influence IMDb scores differently.

4. **Runtime**: The length of a movie can affect its IMDb score. You can create features like 'Short Film' or 'Long Film' based on certain runtime thresholds.

5. **Budget and Box Office Features**: Include financial metrics like production budget and box office revenue. High budget movies may be expected to have higher IMDb scores.

6. **Categorical Features**: Include other categorical features like the movie's language, country of origin, and MPAA rating. These can affect audience perceptions.

7. **Word Count in Description**: If you have access to the movie's description or plot summary, you can create a feature based on the word count. Longer descriptions might provide more information, potentially influencing ratings.

8. **Sentiment Analysis**: Analyse the sentiment of user reviews for the movie. You can use Natural Language Processing (NLP) techniques to calculate sentiment scores and include them as features.

9. **User Review Statistics**: Aggregate user review statistics, such as the number of reviews, average user rating, and the standard deviation of user ratings.

10. **Social Media Presence**: If available, you can include features related to the movie's social media presence, such as the number of Facebook likes, Twitter followers, or YouTube views.

11. **Awards and Nominations**: Create features based on the number of awards and nominations a movie has received.

12. **User Ratings of Actors and Directors**: If available, you can incorporate user ratings of the movie's lead actors and director from platforms like IMDb.

13. **Time-Based Features**: Consider features related to when the movie was released, such as holidays or notable events.

14. **Composite Features**: Create composite features that combine relevant factors. For example, a "Hollywood Blockbuster" feature could combine high budget, well-known actors, and a wide release.

15. **User Demographics**: If available, consider features related to the demographics of IMDb users who have rated the movie, such as their age, gender, or location.

```
Import pandas as pd

# Sample data
Data = {
    'user_reviews': [100, 200, 300, 150, 250],
    'critic_reviews': [80, 90, 70, 85, 95],
    'budget_in_million': [10, 20, 15, 30, 25]
}

Df = pd.DataFrame(data)

# Feature engineering
Df['user_critic_review_ratio'] = df['user_reviews'] / df['critic_reviews']
Df['budget_per_review'] = df['budget_in_million'] / (df['user_reviews'] + df[
'critic_reviews'])

# You can add more features based on your domain knowledge

# Calculating IMDb score (This is a simple linear combination, actual IMDb score calculation
is more complex)
Df['imdb_score'] = 6.5 + 0.01 * df['user_reviews'] + 0.02 * df['critic_reviews'] - 0.1
*
```

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <link

      rel="icon"

      type="image/png"

      sizes="32x32"

      href="./static/images/favicon-32x32.png"

    />

    <title>Easybank landing page</title>

    <link



href="https://fonts.googleapis.com/css2?family=Public+Sans:wght@300;400;50

0;700&display=swap"

      rel="stylesheet"

    />

    <link rel="stylesheet" href="./static/css/styles.css" />

  </head>

  <body>

    <header class="header">
```

```
<a class="header__logo" href="./">
  <img src="./static/images/logo.svg" alt="Logo" />
</a>
<input type="checkbox" id="menu" class="menu__checkbox" />
<ul class="menu">
  <li class="menu__item">
    <a class="menu__link" href="#">Home</a>
  </li>
  <li class="menu__item">
    <a class="menu__link" href="#">About</a>
  </li>
  <li class="menu__item">
    <a class="menu__link" href="#">Contact</a>
</li>
<li class="menu__item">
  <a class="menu__link" href="#">Blog</a>
</li>
<li class="menu__item">
    <a class="menu__link" href="#">Careers</a>
  </li>
</ul>
<label for="menu" class="menu__hamburger"></label>
```

```
  <button type="button" class="btn">Request Invite</button>
</header>
<section class="hero">
  <div class="hero__main">
<div class="wrapper">
  <div class="hero__grid">
    <div class="hero__image"></div>
    <div class="hero__content">
      <h1 class="title__primary">Next generation digital banking</h1>
      <p class="description">
        Take your financial life online. Your Easybank account will be a
        one-stop-shop for spending, saving, budgeting, investing, and
        much more.
      </p>
      <button type="button" class="btn">Request Invite</button>
    </div>
  </div>
</div>
  </div>
</section>
<section class="why-easybank">
  <div class="wrapper">
```

```
<h2 class="title__primary">Why choose Easybank?</h2>

<p class="why-easybank__description">
  We leverage Open Banking to turn your bank account into your financial
  hub. Control your finances like never before.
</p>

<div class="why-easybank__grid">
  <article class="why-easybank__feature">
    <img src="./static/images/icon-online.svg" alt="Online Banking" />
    <h3 class="title__secondary">Online Banking</h3>
    <p class="description">
      Our modern web and mobile applications allow you to keep track of
      your finances wherever you are in the world.
    </p>
  </article>
  <article class="why-easybank__feature">
    <img
      src="./static/images/icon-budgeting.svg"
      alt="Simple Budgeting"
    />
    <h3 class="title__secondary">Simple Budgeting</h3>
    <p class="description">
      See exactly where your money goes each month. Receive
```

```
    notifications when you’re close to your hitting limits.
  </p>
</article>
<article class="why-easybank__feature">
<img
  src="./static/images/icon-onboarding.svg"
  alt="Fast Onboarding"
/>
<h3 class="title__secondary">Fast Onboarding</h3>
<p class="description">
  We don’t do branches. Open your account in minutes online and
    start taking control of your finances right away.
  </p>
</article>
<article class="why-easybank__feature">
  <img src="./static/images/icon-api.svg" alt="Open API" />
  <h3 class="title__secondary">Open API</h3>
  <p class="description">
    Manage your savings, investments, pension, and much more from one
    account. Tracking your money has never been easier.
  </p>
</article>
```

```
    </div>
  </div>
</section>
<section class="latest-articles">
<div class="wrapper">
  <h2 class="title__primary">Latest Articles</h2>
  <div class="latest-articles__grid">
    <article class="latest-articles__item">
      <img
        src="./static/images/image-currency.jpg"
        height="200"
  alt="Currency"
/>
<div class="latest-articles__item-content">
  <small class="latest-articles__item-autor">
    By Claire Robinson
  </small>
  <h3 class="title__tertiary">
    Receive money in any currency with no fees
</h3>
<p class="latest-articles__item-description">
  The world is getting smaller and we’re becoming more mobile. So
```

```
      why should you be forced to only receive money in a single ⋯
    </p>
  </div>
</article>
<article class="latest-articles__item">
  <img
    src="./static/images/image-restaurant.jpg"
    height="200"
    alt="Restaurant"
  />
  <div class="latest-articles__item-content">
    <small class="latest-articles__item-autor">
      By Wilson Hutton
    </small>
    <h3 class="title__tertiary">
      Treat yourself without worrying about money
    </h3>
    <p class="latest-articles__item-description">
      Our simple budgeting feature allows you to separate out your
      spending and set realistic limits each month. That means you ⋯
    </p>
  </div>
```

```
</article>

<article class="latest-articles__item">

  <img

    src="./static/images/image-plane.jpg"

    height="200"

    alt="Plane"

  />

  <div class="latest-articles__item-content">

    <small class="latest-articles__item-autor">

      By Wilson Hutton

    </small>

    <h3 class="title__tertiary">

      Take your Easybank card wherever you go

    </h3>

    <p class="latest-articles__item-description">

      We want you to enjoy your travels. This is why we don’t charge

      any fees on purchases while you’re abroad. We’ll even show you …

    </p>

  </div>

</article>

<article class="latest-articles__item">

  <img
```

```
    src="./static/images/image-confetti.jpg"
    height="200"
    alt="Confetti"
  />
  <div class="latest-articles__item-content">
    <small class="latest-articles__item-autor">
      By Claire Robinson
    </small>
    <h3 class="title__tertiary">
      Our invite-only Beta accounts are now live!
    </h3>
      <p class="latest-articles__item-description">
        After a lot of hard work by the whole team, we’re excited to
        launch our closed beta. It’s easy to request an invite through
        the site ...
      </p>
    </div>
  </article>
    </div>
  </div>
</section>
<footer class="footer">
```

```
<div class="wrapper">
  <div class="footer__grid">
    <a class="footer__logo" href="/">
      <img src="./static/images/logo-white.svg" alt="Logo" />
    </a>
    <div class="footer__social">
      <a href="#">
        <img
          src="./static/images/icon-facebook.svg"
          class="footer__icon"
          width="20"
          alt="Facebook"
        />
      </a>
      <a href="#">
        <img
          src="./static/images/icon-youtube.svg"
          class="footer__icon"
          width="20"
          alt="Youtube"
        />
      </a>
```

```
<a href="#">
  <img
    src="./static/images/icon-twitter.svg"
    class="footer__icon"
    width="20"
    alt="Twitter"
  />
</a>
<a href="#">
  <img
    src="./static/images/icon-pinterest.svg"
    class="footer__icon"
    width="20"
    alt="Pinterest"
  />
</a>
<a href="#">
  <img
    src="./static/images/icon-instagram.svg"
    class="footer__icon"
    width="20"
    alt="Instagram"
```

```
      />
    </a>
  </div>
  <div class="footer__links">
    <a href="#" class="footer__link">About Us</a>
    <a href="#" class="footer__link">Contact</a>
    <a href="#" class="footer__link">Blog</a>
    <a href="#" class="footer__link">Careers</a>
    <a href="#" class="footer__link">Support</a>
    <a href="#" class="footer__link">Privacy Policy</a>
  </div>
    <div class="footer__button">
      <button type="button" class="btn">Request Invite</button>
    </div>
    <p class="footer__copyright">© Easybank. All Rights Reserved</p>
  </div>
 </div>
</footer>
<script>
  const menu = document.querySelector('#menu')
  const mediaScreen = window.matchMedia('screen and (min-width: 1024px)')
  mediaScreen.addListener(handleMediaScreen)
```

```
/**
 * Set the checked attribute of checkbox menu to false
 */
  function handleMediaScreen(event) {
    if (event.matches && menu.checked) {
      console.log('Set checked to false')
      menu.checked = false
    }
  }
</script>
  </body>

</html>
```

```
# Sample IMDb ratings data
Ratings = {
    “movie1”: 8.5,
    “movie2”: 7.2,
    “movie3”: 9.0,
   # Add more movies and ratings as needed
}

# Weighted average calculation
Total_weighted_rating = 0
Total_weight = 0

For movie, rating in ratings.items():
    Weight = rating  # You can customize the weight calculation
    Total_weighted_rating += rating * weight
    Total_weight += weight

If total_weight == 0:
```

```
    Imdb_score = 0  # Avoid division by zero
Else:
    Imdb_score = total_weighted_rating / total_weight

Print(“IMDb Score:”, imdb_score)
```

CONCLUSION:

In conclusion, the innovation in IMDb score prediction through applied data science represents a significant advancement in the film industry. This approach combines cutting-edge algorithms, real-time capabilities, and user-friendly interfaces to offer highly accurate and interpretable IMDb score predictions. By fostering transparency, personalization, and ethical practices, it not only aids filmmakers and studios in decision-making but also empowers movie enthusiasts with tailored recommendations. The continual learning aspect ensures predictions stay relevant, while open-source collaboration encourages the data science community's contributions. Ultimately, this innovation revolutionizes how we engage with movies, creating a more informed, personalized, and enjoyable movie-watching experience for audiences.

Done by:
AVULA MALLA REDDY
AU720921244010
JCT COLLEGE OF ENGINEERING AND TECHNOLOGY