APPLIED DATA SCIENCE - GROUP 3

PROJECT 2 : PREDICTING IMDB SCORES

ABSTRACT

PHASE 2:

- IMDB Movie Dataset Analysis
- Python Pandas IMDb Movies Data
- Python Review of IMDB Data
- Movies Data Science — Pull & Analyze IMDb data using Python
- Pandas IMDb Movies Data Analysis
- IMDB DataSet Visualization & Data Analytics Using Pandas
- IMDbPY is a Python package for retrieving and managing the data of the IMDb movie database about movies and people.
- IMDB DataSet Visualization & Data Analytics Using Pandas

PREDICTING IMDB SCORES

PHASE 2 - Phase 2: Innovation

DATASET : https://www.kaggle.com/datasets

DESIGN OVERVIEW :

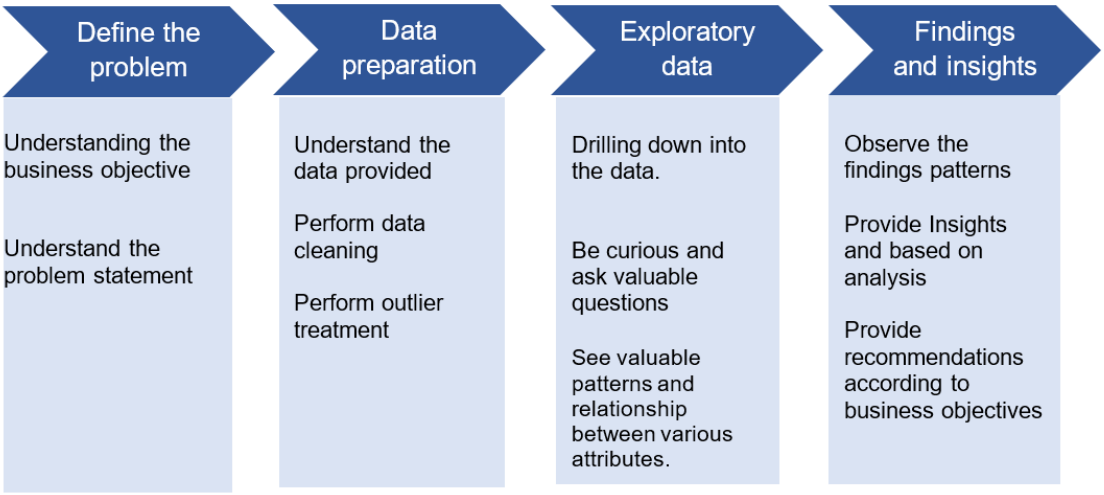
- STEPS ARE - DATA SOURCE
- DATA PREPROCESSING
- MODEL SELECTION
- MODEL TRAINING EVALUATION

PREDICTING IMDB SCORES FOR TOP 100 MOVIES

Advanced title search results for top 100 and top 250 movies can be sorted by different attributes (popularity, a-z, user rating, number of votes, etc.), and the order in which they are displayed reflects the chosen sorting order and not a movie's rank within the authoritative top 250 chart, which consists of the top ...

IMDB Movie data Analysis :

IMDB Movie data Analysis · IMDB Data Correlation·
Drawing bar graph based on top 10 years of most number of movies releases.



Step 1: Data Collection

Collect data for the top 100 movies, including features like director, cast, budget, genre, release date, and other relevant information. You can gather this data from IMDb, or other movie databases. You may also need historical IMDb ratings as your target variable.

Step 2: Data Cleaning

Clean the data to remove missing values, duplicates, and inconsistencies. Ensure that all data is in a format suitable for analysis. Normalize or standardize numerical features if necessary.

	Code	Title	Rating	Rating Count	Rank	Rating Mean
0	tt7366338	Chernobyl	9.1	5,431	12	9.300000
1	tt9253866	Our Planet	9.3	8,356	6	9.262500
2	tt2560140	Attack on Titan	8.7	109,643	66	9.258824
3	tt8595766	Yeh Meri Family	8.5	15,026	159	9.200000
4	tt2395695	Cosmos	9.2	94,934	9	9.076923

Step 3: Feature Engineering

Create new features that might be useful for predicting IMDb scores. For example, you can calculate the age of the movie when it was rated, create binary features for genres, or compute the average IMDb score of the director's previous work.

Movie Metadata Features:

Movie Release Year: Create a feature to represent the age of the movie when it was rated. Older movies may have different rating trends.

Movie Budget: Incorporate the movie's budget as a feature. Higher budget films might have higher production quality and impact ratings.

Categorical Features:

Genre: Convert movie genres into binary features (e.g., one-hot encoding). Some genres might be more popular or have different rating patterns.

Director and Cast: Create features based on the director and lead actors. The reputation of these individuals can impact a movie's rating.

Text Data:

Plot Summary Analysis: You can perform natural language processing (NLP) on the plot summaries or movie descriptions. Extract sentiment scores, keyword counts, or topic modeling features.

User Reviews and Critics' Reviews: Analyze sentiment scores from user and critic reviews. You can use tools like sentiment analysis libraries or perform more advanced NLP techniques.

Temporal Features:

Season and Month of Release: Create features based on the season or month of the movie release. Seasonal trends can affect ratings.

External Data:

Awards and Nominations: Include information about the number of awards and nominations a movie received. This could be a sign of its quality.

Economic Indicators: If you have access to data on economic indicators at the time of release, this can be used to capture the economic context.

User Interaction Features:

User Ratings: Include features related to user ratings such as the number of user ratings, average user rating, and user rating trends over time.

Historical Data:

Director's Average IMDb Score: Calculate the average IMDb score of the director's previous work. This can be an indicator of directorial quality.

Actor's Average IMDb Score: Similar to directors, you can calculate the average IMDb score of lead actors.

Production Studio: Incorporate the production studio as a feature. Different studios may have varying standards and quality.

Social Media Buzz: If available, use features related to social media mentions, hashtags, or online discussions about the movie.

Composite Features: Create interaction terms or composite features. For example, you can create a "Genre x Director" feature to capture the interaction between genre and director influence.

Cross-Validation Features: You can include features like k-fold cross-validation mean rating, which provides a more robust estimate of a movie's rating based on how it performs in various cross-validation splits of the data.

Text Embeddings: Use pre-trained word embeddings (e.g., Word2Vec, GloVe, or BERT embeddings) to convert text data into numerical features

Step 4: Data Splitting

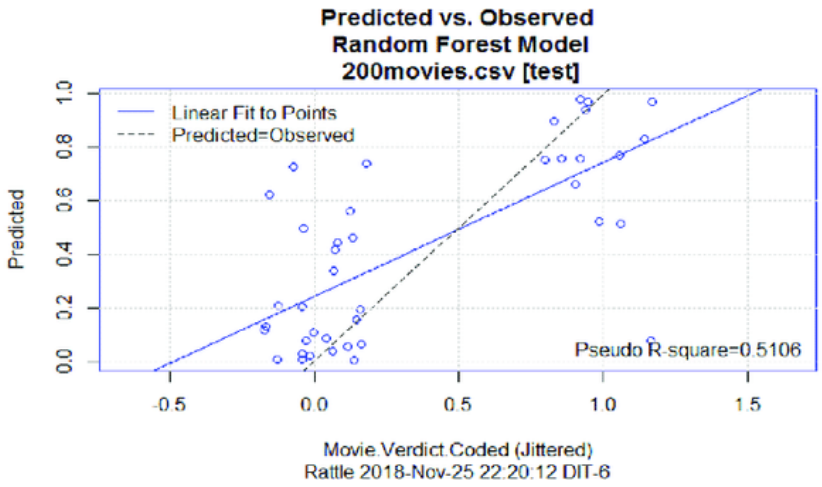
Split your dataset into a training set, a validation set, and a test set. The training set will be used to train your predictive model, the validation set to fine-tune the model's hyperparameters, and the test set to evaluate the model's performance.

Step 5: Model Selection

Choose a regression model suitable for predicting IMDb scores. Common choices include linear regression, decision trees, random forests, and neural networks. Experiment with different models to see which one performs best.

“RANDOM FOREST REGRESSION”

Random Forest algorithm had the best performance of 92.7%. This is the highest score obtained among similar studies. Keywords - Machine learning, WEKA, movie



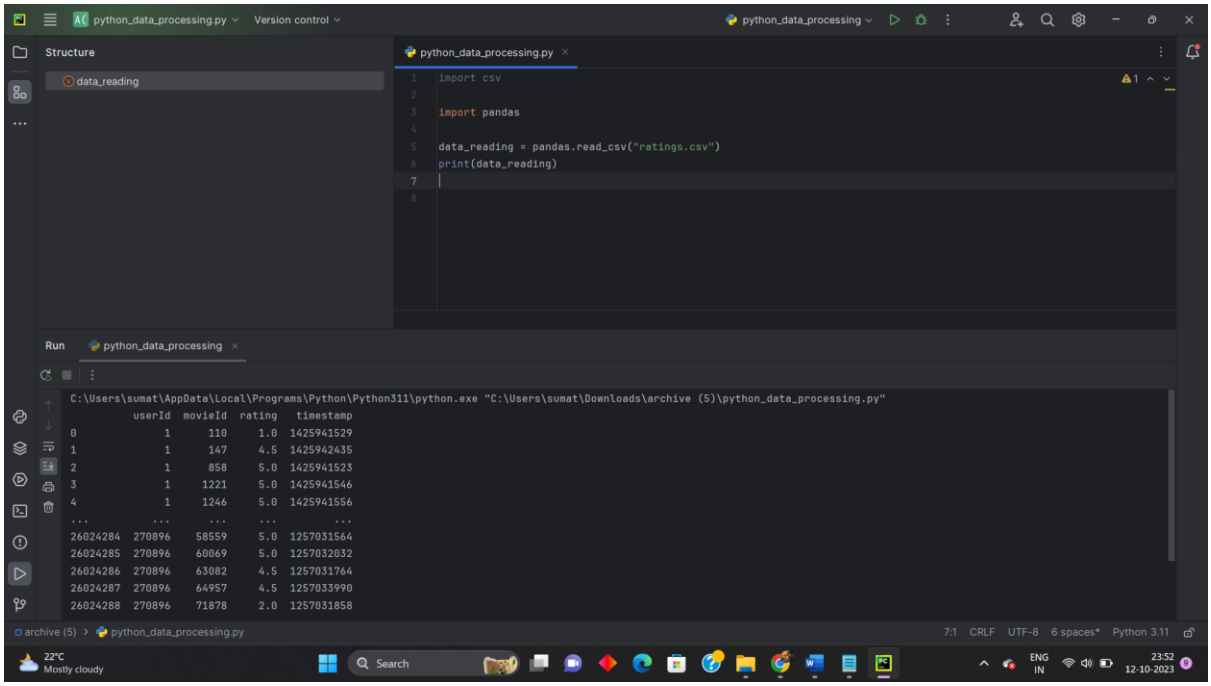
Step 6: Model Training

Train your selected model on the training dataset. Use the features you've engineered and the historical IMDb scores as the target variable.

```
import csv


import pandas


data_reading = pandas.read_csv("ratings.csv")
print(data_reading)
```



Step 7: Model Evaluation

Evaluate the model's performance using the validation dataset. Common evaluation metrics for regression tasks include mean squared error (MSE), root mean squared error (RMSE), and R-squared (R^2).

Step 8: Hyperparameter Tuning

Fine-tune your model by adjusting hyperparameters (e.g., learning rate, number of trees in a random forest, neural network architecture) based on the validation set's performance.

Step 9: Model Testing

After you've trained and fine-tuned your model, test it on the separate test set to assess how well it generalizes to new, unseen data.

Step 10: Interpret the Results

Analyze the model's performance and consider which features have the most influence on predicting IMDb scores. You can also generate predictions for the IMDb scores of the top 100 movies using your model.

Step 11: Model Deployment (Optional)

If you want to use your model for ongoing predictions, deploy it as a service or integrate it into a web application.

Step 12: Iterate

Iterate on the process, considering additional data sources, new features, and more advanced modeling techniques to improve your predictions.

Please note that IMDb scores can be influenced by many factors, and this simplified approach might not yield highly accurate predictions. More advanced techniques, such as natural language processing to analyze reviews or incorporating external data, can further improve prediction accuracy. Additionally, you may need a large dataset with more features to build a more accurate model.