APPLIED DATA SCIENCE – GROUP 3

Project 2 : PREDICTING IMDB SCORES

ABSTRACT

PHASE 1:

- PROBLEM DEFINITION
- DESIGN THINKING

- ➢ DATA SOURCE
- ➢ DATA PREPROCESSING
- ➢ MODEL SELECTION
- ➢ MODEL TRAINING
- ➢ EVALUATION

PREDICTING IMDB SCORES

PROBLEM DEFINITION :

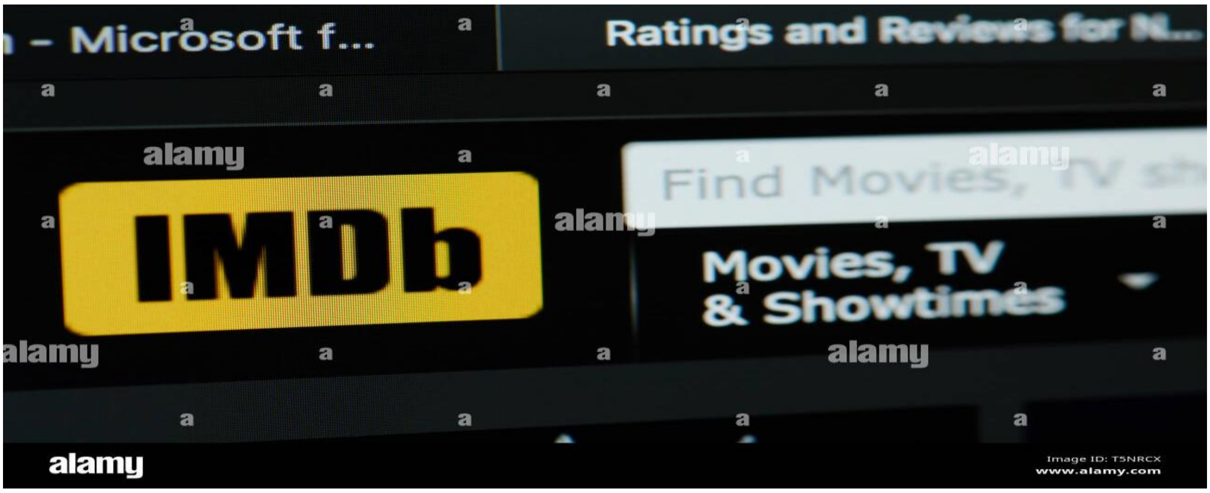
The problem is to develop a machine learning model that predicts IMDb scores of movies available on Films based on features like genre, premiere date, runtime, and language. The objective is to create a model that accurately estimates the popularity of movies, helping users discover highly rated films that match their preferences. This project involves data preprocessing, feature engineering, model selection, training, and evaluation.

- In this project, we take IMDB scores as response variable and focus on operating predictions by analyzing the rest of variables in the IMDB 5000 movie data. The results can help film companies to understand the secret of generating a commercial success movie.

Step 1: Data acquisition & cleaning ·

Step 2: Models and features ·

Step 3: Testing and training / the results.

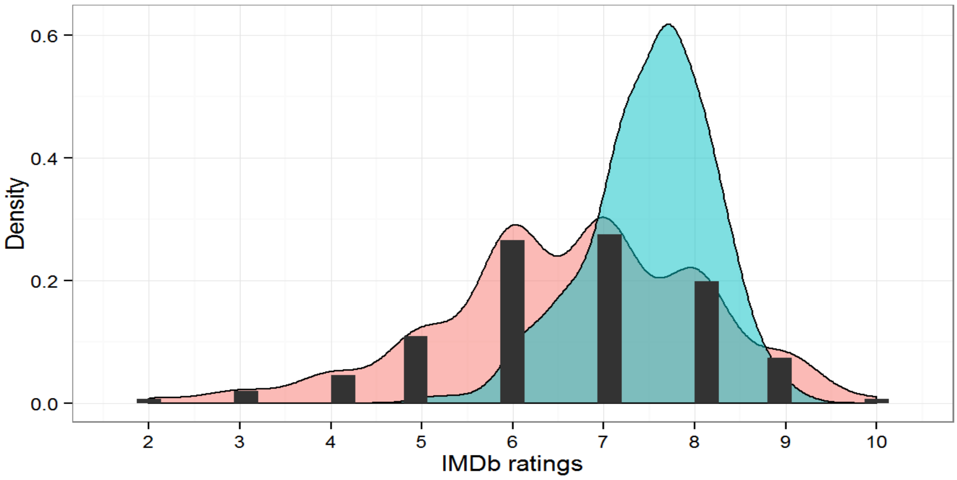


- This is done by taking all of the ratings for a movie or TV show and then weighting them according to how many votes each rating has. For example, if a movie has 100 ratings and 90% of those ratings are 5 stars, then the weighted average rating would be 4.5 stars.

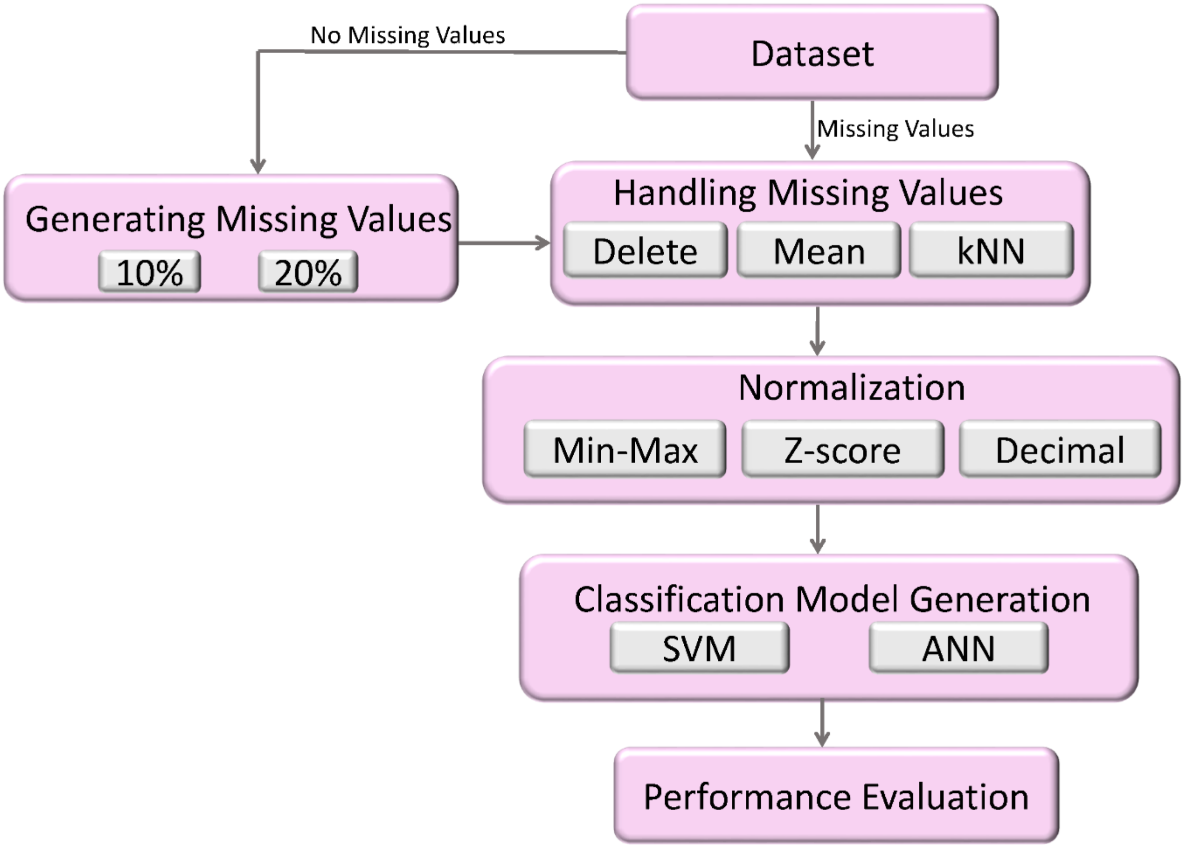
DESIGN THINKING

1. Data Source: Utilize a dataset containing information about movies, including features like genre, premiere date, runtime, language, and IMDb scores**.**

    - IMDb registered users can cast a vote (from 1 to 10) on every released title in the database. Individual votes are then aggregated and summarized as a single IMDb rating, visible on the title's main page.



2. Data Preprocessing: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.



- Data cleaning and preprocessing refer to the process of identifying and rectifying errors, inconsistencies, and inaccuracies in raw data. It involves several steps, such as removing duplicate records, handling missing values, dealing with outliers, standardizing formats, and resolving inconsistencies.

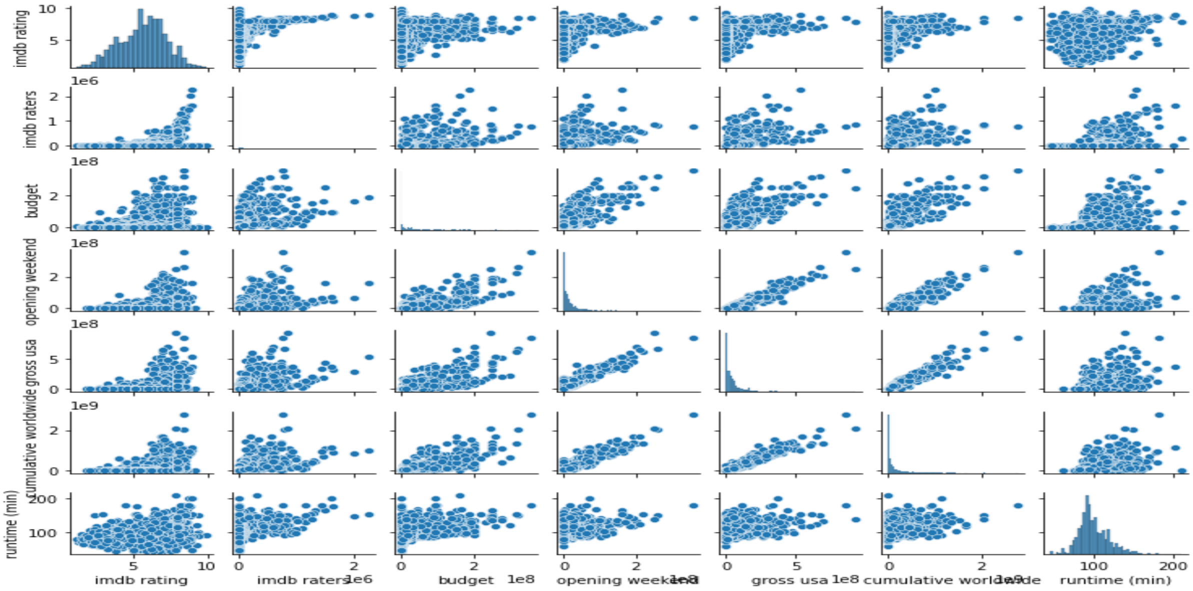
3. Feature Engineering: Extract relevant features from the available data that could contribute to predicting IMDb scores.

In its entirety, this project explored a few critical skills required of a data scientist:

- Web scraping (requests, HTML, Beautiful Soup)
- EDA (pandas, numpy)
- Linear regression (scikit-learn)
- Data visualization (seaborn, matplotlib)

- IMDb has an API available to download bulk data, but a primary requirement for this project was to obtain data through web scraping; so, I went along and got the information from IMDb using requests and Beautiful Soup. Requests is the module required to take the webpage and turn it into an object in python. Beautiful Soup takes that object, which is the HTML information behind the webpage, and makes searching and accessing specific information within the HTML text easy. You really need both in order to fully complete the process of web scraping.

```
sns.pairplot(movies_df_drop, height=1.2, aspect=1.25)
```



4. Model Selection: Choose appropriate regression algorithms (e.g., Linear Regression, Random Forest Regressor) for predicting IMDb scores.

In general, Random Forest tends to perform better than Linear Regression when: The data has a large number of features. The data has complex, non-linear relationships. The data contains missing values or outliers.

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

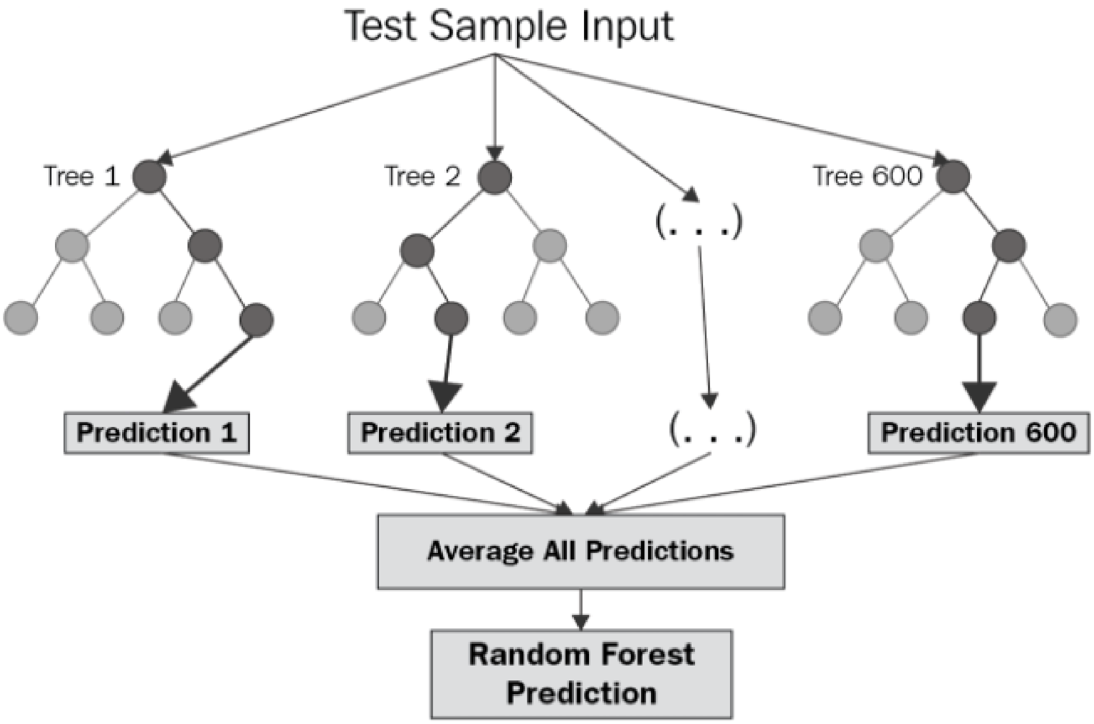
Let's see Random Forest Regression in action!

Step 1: Identify your dependent (y) and independent variables (X) ...

Step 2: Split the dataset into the Training set and Test set. ...

Step 3: Training the Random Forest Regression model on the whole dataset. ...

Step 4: Predicting the Test set results.



5. Model Training: Train the selected model using the preprocessed data.

The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let's walk through the steps:

- Pick at random k data points from the training set.

- Build a decision tree associated to these k data points.

- Choose the number N of trees you want to build and repeat steps 1 and 2.
- For a new data point, make each one of your N-tree trees predict the value of y for the data point in question and assign the new data point to the average across all of the predicted y values.

    - on many problemA Random Forest Regression model is powerful and accurate. It usually performs great s, including features with non-linear relationships. Disadvantages, however, include the following: there is no interpretability, overfitting may easily occur, we must choose the number of trees to include in the model.

6. Evaluation: Evaluate the model's performance using regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

There are three error metrics that are commonly used for evaluating and reporting the performance of a regression model; they are:

- Mean Squared Error (MSE).
- Root Mean Squared Error (RMSE).
- Mean Absolute Error (MAE)

Dataset Link: https://www.kaggle.com/datasets/luiscorter/netflix-original-films-imdb-scores