# PREDICTING IMDb SCORES

## PROBLEM STATEMENT:

The problem is to develop a machine learning model that predicts IMDb scores for movies available. This predictive model will utilize features such as genre, premiere date, runtime, and language to estimate the popularity or rating of movies. The primary objective is to assist users in discovering highly rated movies that align with their preferences, ultimately enhancing their movie-watching experience.

## OBJECTIVE:

The main objective of this project is to create a machine learning model that can accurately estimate the popularity of movies. This estimation is based on certain features, including genre, premiere date, runtime, and language. In other words, to build a model that can predict how well a movie is likely to be rated on IMDb.

## PROBLEM SOLUTION:

We propose to develop a machine learning model to predict IMDb scores of movies. The model will be trained on a dataset of movies with known IMDb scores and features such as genre, premiere date, runtime, and language. Once trained, the model can be used to predict the IMDb scores of new movies, helping users discover highly rated films

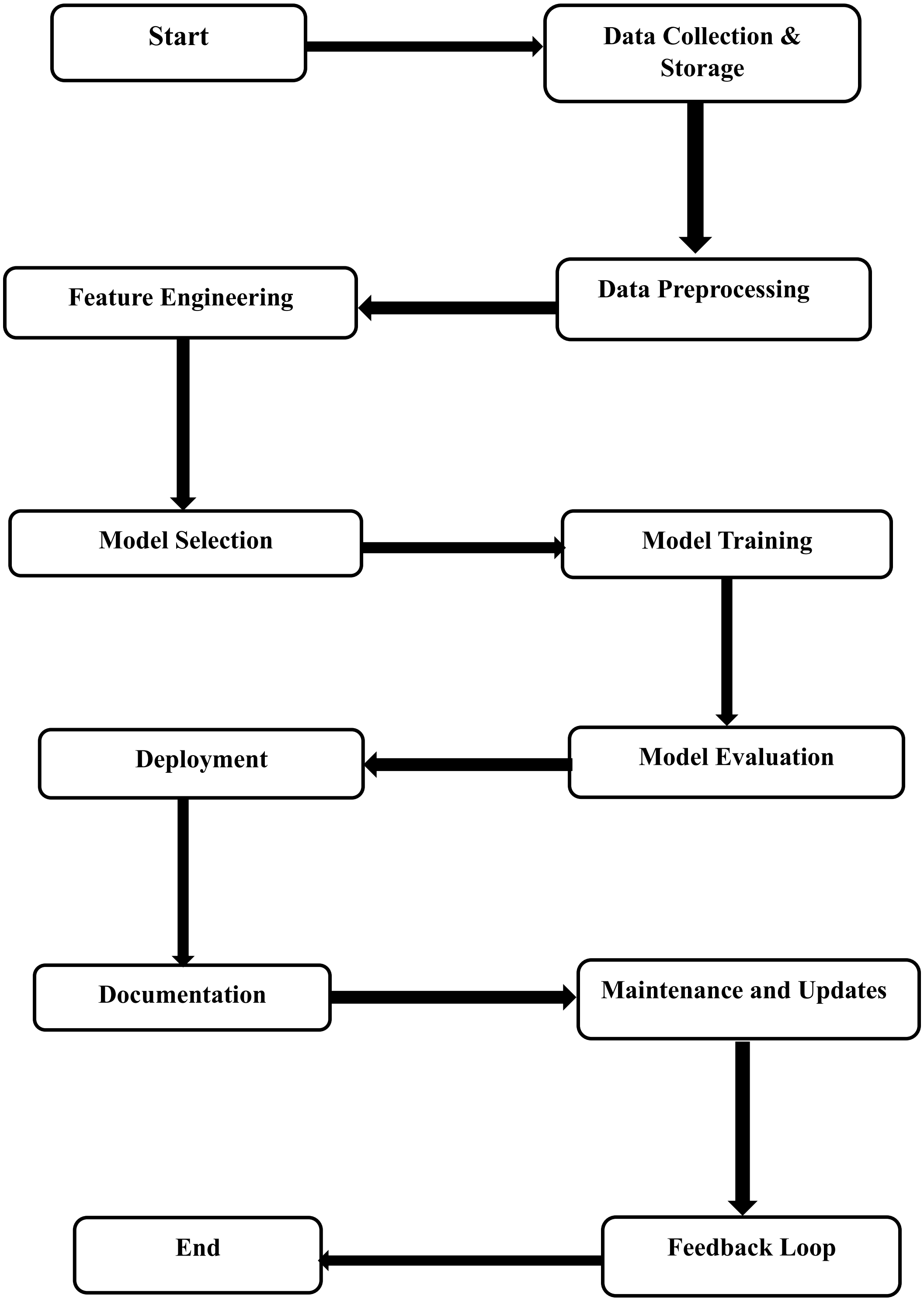
## CONTEXT:

IMDb (Internet Movie Database) is a popular platform where users can find information about movies, including user and critic ratings. Our project aims to leverage this data to help users discover highly rated films that align with their preferences. It is one of the most popular and comprehensive sources of information related to the entertainment industry

## POTENTIAL CONSTRAINTS AND LIMITATIONS:

1) Data Availability
2) Data Bias
3) Model Complexity
4) Evaluation Metrics
5) User Preferences
6) Legal Constraints
7) User feedback

**<u>FLOW CHART:</u>** **Predicting IMDb Scores**



**STEPS TO IMPLEMENT DESIGN:**

**Step 1: Empathy**

Imagine ourself as a movie enthusiast who loves discovering new films to watch. Users often rely on IMDb ratings to choose what to watch, but it can be time-consuming to browse through all the options, so users want a tool that can make personalized movie recommendations based on your taste

**Step 2: Data Collection & Storage**

- Obtain a dataset containing movie information, including features such as genre, premiere date, runtime, language, and IMDb scores.

- Store the dataset in a structured format, such as a relational database or a data warehouse, to facilitate data access and manipulation.

**Step 3: Data Preprocessing**

- **Data Cleaning:** This involves removing any duplicate entries, correcting errors, and ensuring data consistency. It's essential to have clean and reliable data for accurate predictions.

- **Handling Missing Values:** Decide on an appropriate strategy for handling missing data. Common methods include imputation (replacing missing values with reasonable estimates) or removing rows/columns with too many missing values.

- **Categorical to Numerical Conversion**: Many machine learning algorithms require numerical inputs. Use techniques like one-hot encoding (for categorical variables with no inherent order) or label encoding (for ordinal categorical variables) to convert categorical features into numerical ones.

**Step 4: Feature Engineering**

- **Feature Selection**: Analyze the importance of each feature in relation to predicting IMDb scores. Feature selection techniques like correlation analysis or feature importance from tree-based models can help in choosing the most relevant features.

- **Creating New Features**: If necessary, generate new features that can capture valuable information. For example, you could create a "release season" feature based on the premiere date or calculate the average IMDb score for each genre.

**Step 5: Model Selection**

- **Algorithm Selection:** Experiment with different regression algorithms suitable for predicting continuous values like IMDb scores. Common choices include Linear Regression (for simplicity), Random Forest Regressor (for non-linearity), and Gradient Boosting Regressor (for high accuracy).

- **Hyperparameter Tuning:** Fine-tune the selected models by adjusting hyperparameters. Techniques like grid search or randomized search can help find the optimal hyperparameters.

**Step 6: Model Training**

- Split your dataset into training and testing subsets (typically 70-80% for training and the rest for testing). This separation helps evaluate the model's generalization performance.

- Train each selected model on the training data, using techniques like cross-validation to prevent overfitting.

**Step 7: Model Evaluation**

- Assess the performance of each model using appropriate regression evaluation metrics:

- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual IMDb scores.

- **Mean Squared Error (MSE):** Measures the average squared difference between predicted and actual scores, giving more weight to large errors.

- **R-squared ($R^2$):** Evaluates how well the model fits the data. A higher $R^2$ indicates a better fit.

**Step 8: Deployment**

Once you've chosen the best-performing model, integrate it into a website, application, or system where users can input movie details and receive IMDb score predictions.

**Step 9: Documentation**

-Maintain thorough documentation for the entire pipeline, including data sources, preprocessing steps, model selection criteria, and deployment instructions.

- Implement logging to capture important events, errors, and model performance metrics for auditing and debugging.

**Step 10: Maintenance and update**

Keep the model up-to-date with new movie data to ensure its accuracy.

**Step 11: Feedback Loop**

- Incorporate user feedback into the model's improvement cycle, allowing for adjustments to the feature set, model selection, and hyperparameters.

- Maintain a feedback loop to gather user ratings and incorporate them into the dataset for continuous model improvement.

## **CONCLUSION**:

By addressing the above components, this project aims to provide a valuable tool for movie enthusiasts, enabling them to discover and enjoy highly-rated films based on their preferences and interests.