**🎬 Solution Explanation: IMDb Rating Prediction Using Machine Learning**

In this project, I worked with the **IMDb Top 250 Shows dataset** to build a machine-learning model that can predict IMDb ratings based on features such as **genre**, **year**, **votes**, and other metadata.

**1. Data Understanding & Cleaning**

* Loaded the dataset using **pandas** and explored the structure and types of data
* Handled **missing values** and removed irrelevant or redundant columns Renamed columns where necessary for better readability.
* Checked data distributions and cleaned anomalies like outliers in vote counts or unexpected rating values.
* One-hot encoded **categorical features** like Genre, which contained multiple comma-separated values.

**2. Exploratory Data Analysis (EDA)**

* Performed basic statistical analysis to understand:
  + Distribution of ratings and votes.
  + Most common genres.
  + Year-wise show production trends.
* Key questions explored:
  + Do more votes mean higher ratings?
  + Which genres typically receive higher ratings?
  + Is there a trend over the years in how shows are rated

**3. Visualizations**

* Used **Matplotlib** and **Seaborn** to generate multiple plots:
  + **Histogram** of IMDb ratings.
  + **Scatter plot** showing the relationship between votes and ratings.
  + **Correlation heat map** to identify relationships between numerical features.
  + **Bar plots** for genre-wise rating averages.
  + **Pie charts** for genre distribution.
  + **Box plots** to visualize rating distribution across different genres and decades.

**4. Feature Engineering**

* Created **decade-based categories** using the Year column.
* Transformed the Genre column into **multiple binary columns** using one-hot encoding (multi-label).
* Scaled numerical features like Votes using **StandardScaler** to normalize for better model performance.
* Removed features like Title that do not contribute predictive value.

**5. Model Building**

* Split the dataset into **training and testing** sets (80/20 split).
* Trained multiple regression models including:
  + **Linear Regression**
  + **Decision Tree Regressor**
  + **Random Forest Regressor**
  + **Cat Boost Regressor**

**6. Model Evaluation**

* Evaluated models using:
  + **Mean Absolute Error (MAE)**
  + **Mean Squared Error (MSE)**
  + **R-squared (R² Score)**
* Plotted **residuals** and **predicted vs actual** ratings to check performance visually.
* Compared results of all models to determine the most accurate one.

**7. Hyper parameter Tuning**

* Used **Grid SearchCV** and **RandomizedSearchCV** on Random Forest and Cat Boost models.
* Tuned parameters like:
  + n\_estimators
  + max\_depth
  + min\_samples\_split
  + learning\_rate
* Achieved significant performance improvement on test data.

**8. Final Predictions**

* Applied the **best-performing model** (e.g., Cat Boost Regressor) to predict ratings on the test dataset.
* Compared **predicted ratings** against actual ones using scatter plots and regression lines.

**9. Insights & Conclusions**

* **Votes** and **Genres** had the highest influence on IMDb rating predictions.
* The most successful models handled **nonlinear relationships** better (e.g., Cat Boost).
* Categorical data (multi-genre tags) added complexity but improved performance after encoding.
* Visualized and interpreted **feature importance**, helping to understand which variables drive rating changes.
* Suggested improvements like adding plot summaries, user reviews, or cast/crew popularity for future iterations.

**Result**

The final notebook provides a complete solution pipeline:

* From **data loading to prediction**
* Includes **visual storytelling**
* Demonstrates strong use of **feature engineering** and **model selection**
* Serves as a great reference for:
  + Beginners learning machine learning
  + Real-world regression use cases
  + Model interpretability practices