Rotten Tomatoes Movies Model

Sathish Madhiyalagan

Python Developer

Objective

The primary objective of this project is to develop a predictive model for accurately forecasting audience ratings for movies. By analyzing various influencing factors such as runtime, genres, director and cast, and critical metrics, the model aims to provide actionable insights that can enhance movie recommendations, improve audience satisfaction, and support better decision-making in the entertainment industry.

Software and Libraries

- Python: Programming language for implementing the machine learning model.

Libraries Used :

- Pandas: For data manipulation and analysis.
- NumPy: For numerical computations.
- Matplotlib and Seaborn: For data visualization.
- Scikit-learn: For building and evaluating regression models.

Step 1: Data Import and Preliminary Exploration

Importing and understanding the dataset is the first step in our analysis. We begin by importing the data into our Python environment and examining its structure and contents. This initial step allows us to familiarize ourselves with the data and prepare it for further analysis.

The dataset used in this project is `Rotten_Tomatoes_Movies3.xls`, which provides valuable information about movies, including:

- Runtime (in minutes).
- Genres, Directors, Cast, and Writers.
- Tomatometer Status (critic metrics).
- Audience Rating (target variable).

By exploring these features, we aim to uncover key insights that will guide the subsequent stages of data preprocessing, feature engineering, and model development.

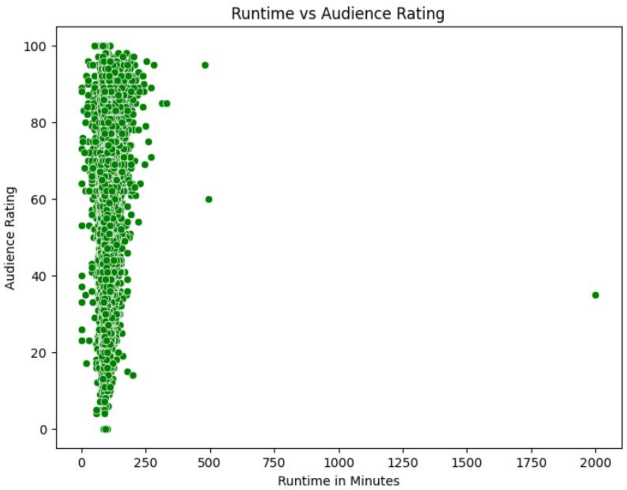
Step 2: Data Preprocessing

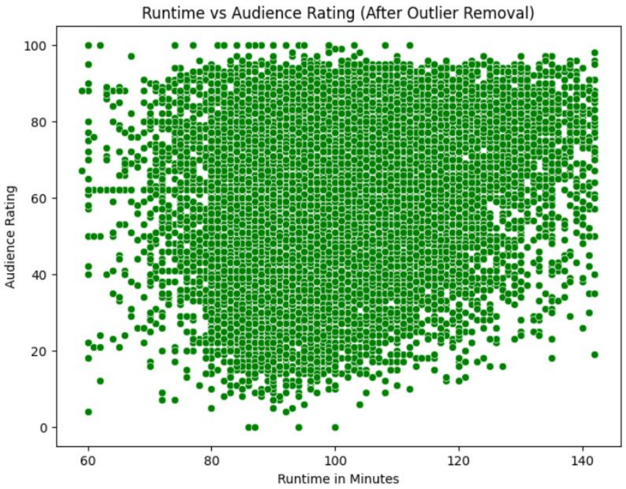
- **Handling Missing Values :**
  - Replaced missing values in numerical columns with the median to minimize the impact of extreme values.
  - Replaced missing values in categorical columns with the placeholder \"Unknown\" to maintain the dataset's integrity.
- **Encoding Features :**
  - Applied One-Hot Encoding to the `genres` column to convert categorical data into binary features.
  - Used MultiLabelBinarizer for columns with multiple values (e.g., `directors`, `writers`) to create binary indicator columns for each unique value.
- **Outlier Removal :**
  - Used IQR (Interquartile Range) method to identify and remove outliers in the `runtime_in_minutes` column, ensuring that the analysis and model training focus on realistic data points.

Step 3: Exploratory Data Analysis (EDA)

1. Scatter Plots :

Insight: The scatter plot indicates whether longer movies generally receive higher or lower ratings. This helps us understand the relationship between runtime and audience preferences.



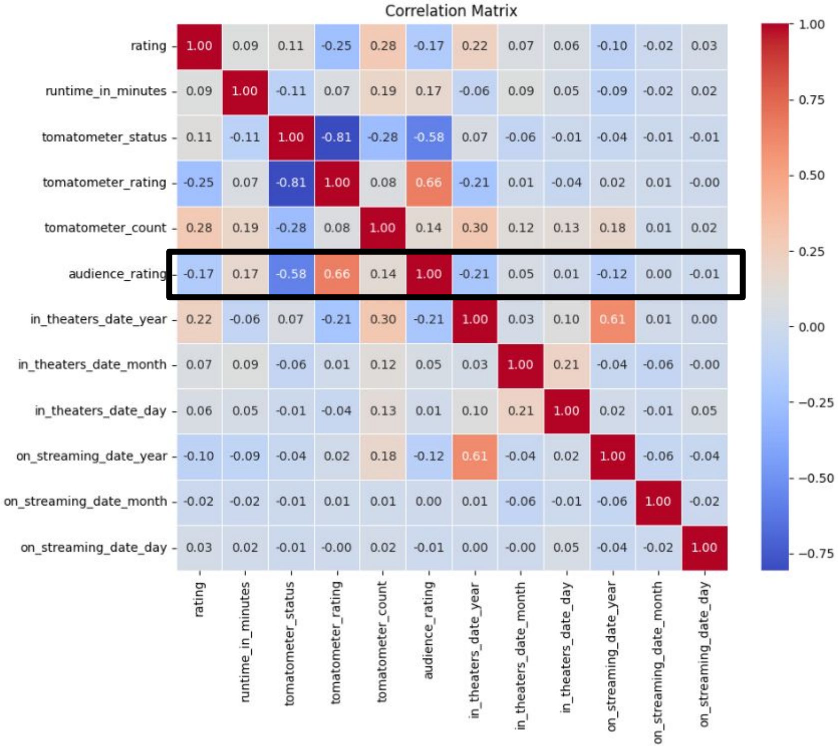


2. Correlation Matrix :

Insight: The heatmap reveals relationships between numerical variables in the dataset.

Features like `runtime_in_minutes` demonstrate a moderate correlation with `audience_rating`.

This insight helps prioritize features for model building, as those with stronger correlations are likely to be more predictive of the target variable.



Correlation Matrix

	rating	runtime_in_minutes	tomatometer_status	tomatometer_rating	tomatometer_count	audience_rating	in_theaters_date_year	in_theaters_date_month	in_theaters_date_day	on_streaming_date_year	on_streaming_date_month	on_streaming_date_day
rating	1.00	0.09	0.11	-0.25	0.28	-0.17	0.22	0.07	0.06	-0.10	-0.02	0.03
runtime_in_minutes	0.09	1.00	-0.11	0.07	0.19	0.17	-0.06	0.09	0.05	-0.09	-0.02	0.02
tomatometer_status	0.11	-0.11	1.00	-0.81	-0.28	-0.58	0.07	-0.06	-0.01	-0.04	-0.01	-0.01
tomatometer_rating	-0.25	0.07	-0.81	1.00	0.08	0.66	-0.21	0.01	-0.04	0.02	0.01	-0.00
tomatometer_count	0.28	0.19	-0.28	0.08	1.00	0.14	0.30	0.12	0.13	0.18	0.01	0.02
audience_rating	-0.17	0.17	-0.58	0.66	0.14	1.00	-0.21	0.05	0.01	-0.12	0.00	-0.01
in_theaters_date_year	0.22	-0.06	0.07	-0.21	0.30	-0.21	1.00	0.03	0.10	0.61	0.01	0.00
in_theaters_date_month	0.07	0.09	-0.06	0.01	0.12	0.05	0.03	1.00	0.21	-0.04	-0.06	-0.00
in_theaters_date_day	0.06	0.05	-0.01	-0.04	0.13	0.01	0.10	0.21	1.00	0.02	-0.01	0.05
on_streaming_date_year	-0.10	-0.09	-0.04	0.02	0.18	-0.12	0.61	-0.04	0.02	1.00	-0.06	-0.04
on_streaming_date_month	-0.02	-0.02	-0.01	0.01	0.01	0.00	0.01	-0.06	-0.01	-0.06	1.00	-0.02
on_streaming_date_day	0.03	0.02	-0.01	-0.00	0.02	-0.01	0.00	-0.00	0.05	-0.04	-0.02	1.00

Feature Correlation:

Numerical features such as `runtime_in_minutes` and `tomatometer_rating` show varying levels of correlation with the target variable, `audience_rating`.

Data Visualization:

Positive correlations (closer to 1) indicate features that may have a direct relationship with audience ratings.

Negative correlations (closer to -1) indicate inverse relationships, which might highlight features negatively impacting ratings.

Impact on Feature Engineering:

Correlation results can guide dimensionality reduction or interaction terms during feature engineering.

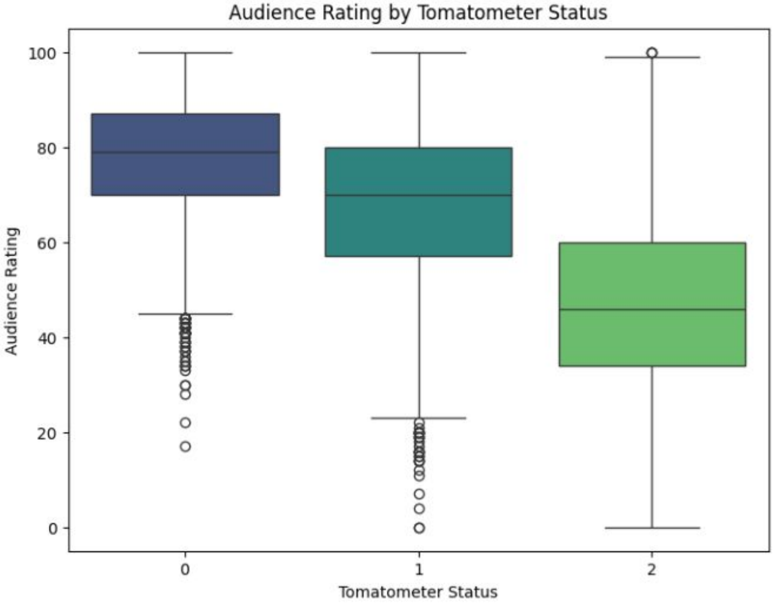
3. Box Plots :

Insight: The box plot demonstrates the distribution of audience ratings for different Tomatometer Status categories.

FCategory 0 has a higher median audience rating compared to others, indicating that movies with this tomatometer status tend to perform better with audiences.

Outliers, represented as individual points, highlight extreme ratings that deviate significantly from the general trend.

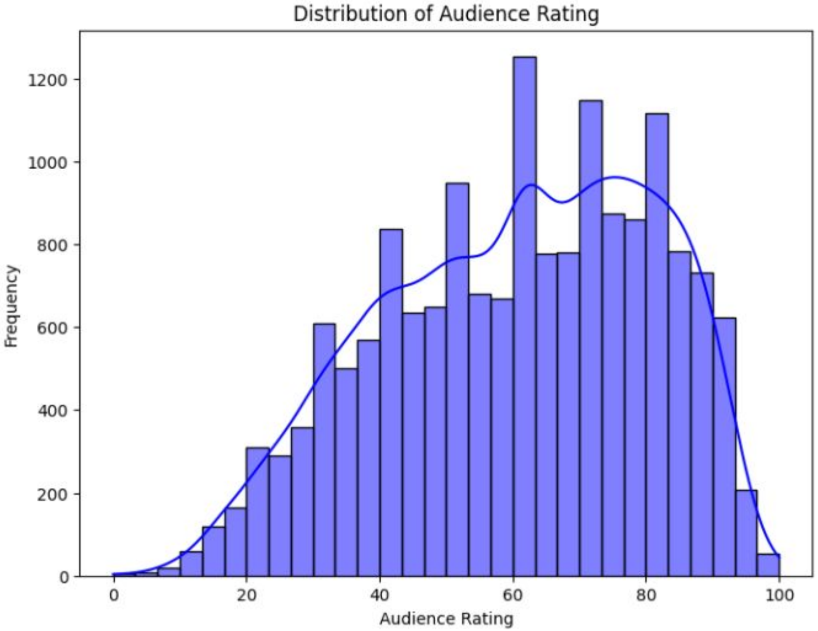
This analysis helps identify the potential influence of critic scores on audience preferences.



4. Distribution Analysis :

Insight: The histogram illustrates the spread and skewness of audience ratings across the dataset.

Observation: Most ratings are concentrated around the mid-range values, between 40 to 80, indicating that a majority of movies received average to above-average ratings from the audience.



The dataset shows a slight right-skew, which implies that fewer movies received extremely high ratings compared to the mid-range.

This balanced distribution is ideal for modeling since it reduces the risk of bias toward extreme values.

Step 4: Feature Engineering

1. Date Features:

- Extracted Year, Month, and Day from the `in_theaters_date` column.
- These features help capture the temporal aspect of movie releases, which may influence audience ratings.

2. Processing Cast:

- Identified the top 100 actors based on their frequency in the dataset.
- Encoded these actors as binary columns, where each column indicates whether a particular actor was in a movie (1) or not (0).
- This transformation captures the impact of prominent actors on audience ratings.

3.One-Hot Encoding for Genres:

- Applied One-Hot Encoding to the `genres` column to create dummy variables.
- Each genre now has its own binary column, indicating whether a movie belongs to a specific genre.
- This ensures the categorical variable is represented numerically for the machine learning models.

Step 5: Model Preparation

1. Splitting the Dataset:

- The dataset was divided into 80% training data and 20% testing data using the `train_test_split` method.
- This ensures that the model is trained on a majority of the data while reserving a portion for evaluation.

Why Split the Data?

- Training data helps the model learn patterns.

- Testing data evaluates the model's ability to generalize to unseen data.

2. Conversion to Sparse Matrices:

- Features were converted into sparse matrices using `csr_matrix`.
- Sparse matrices optimize memory usage by storing only non-zero values, which is especially useful for high-dimensional data (e.g., one-hot-encoded genres).

3. Feature Normalization:

- Applied MaxAbsScaler to scale feature values.
- This scaling technique normalizes data by dividing each feature by its maximum absolute value, ensuring all values lie between -1 and 1.
- Particularly beneficial for sparse data as it preserves sparsity and maintains feature distribution.

Step 6: Models Built

Why These Models Were Used

1. Linear Regression:

- Reason: Acts as a simple baseline model to understand linear relationships between features and the target variable.
- Advantage: Easy to interpret and computationally efficient for initial model comparison.

2. Ridge Regression:

- Reason: Adds L2 regularization to penalize large coefficients, reducing overfitting.
- Advantage: Handles multicollinearity effectively by shrinking coefficients, making the model more robust.

3. Lasso Regression:

- Reason: Uses L1 regularization to enforce sparsity in the model, effectively performing feature selection.
- Advantage: Helps identify and eliminate irrelevant features, improving interpretability and efficiency.

4. Random Forest Regressor:

- Reason: Captures non-linear relationships and interactions between features using an ensemble of decision trees.
- Advantage: Robust against overfitting and works well with high-dimensional data.

5. Voting Regressor (Ensemble Approach):

- Reason: Combines the predictions of multiple models (Linear Regression, Ridge, and Random Forest) to improve overall performance.
- Advantage: Balances the strengths of individual models to achieve better predictions and reduce variance.

Step 7: Model Evaluation

1. R² Score:

- Definition: Indicates the proportion of variance in the target variable that is explained by the model.
- Interpretation:
  - A value closer to 1 implies a better fit.
  - Helps assess how well the model captures the overall trend in the data.

2. Mean Absolute Error (MAE):

- Definition: The average absolute difference between the predicted and actual values.
- Interpretation:
  - Provides an intuitive measure of error.
  - Less sensitive to outliers compared to RMSE.

3. Root Mean Squared Error (RMSE):

- Definition: The square root of the average squared differences between the predicted and actual values.
- Interpretation:
  - Penalizes larger errors more than MAE.
  - Useful for scenarios where large deviations in predictions are undesirable.

Step 8: Evaluation Result

Model	R² Score	MAE	RMSE
Linear Regression	0.0116	15.2686	20.2129
Ridge Regression	0.4981	11.4770	14.4041
Lasso Regression	0.4805	11.6998	14.6534
Random Forest	0.4671	11.5361	14.8418
Voting Regressor	0.4604	11.7268	14.9345

Step 9: Cross-Validation

1. **Linear Regression:**

- **Cross-Validation Scores:**

  [0.1353, 0.0408, -0.0063, -0.0322, 0.0466]

  Explanation:

  - The variability in scores indicates that Linear Regression is sensitive to the data distribution in the folds.

  - Negative scores show that in some splits, the model performed worse than a baseline predictor (mean prediction).

- **Average R² Score:**

  0.0368

  Interpretation:

  - Linear Regression explains only 3.68% of the variance in the target variable on average, indicating poor generalizability.

2.Ridge Regression:

- **Best Hyperparameters:**

  Alpha: 10.

  Explanation:

  - The alpha parameter controls the strength of the regularization.

  - A value of 10 provides the optimal trade-off between bias and variance for this dataset.

- **Average R² Score:**

  0.4981

  Interpretation:

  - Ridge Regression, with its optimized alpha value, performs significantly better, indicating its robustness in capturing meaningful patterns while minimizing overfitting.

Step 10: Residual Analysis

1. Residual Plot:

Description:

- The residual plot shows the difference between the actual and predicted audience ratings.
- Residuals are scattered around the zero line, indicating that the model captures the trend reasonably well.

Insights:

- Presence of patterns in residuals may suggest that the model is missing some key non-linear relationships in the data.
- Opportunities for further optimization or additional feature engineering exist.

2. Actual vs Predicted Plot:

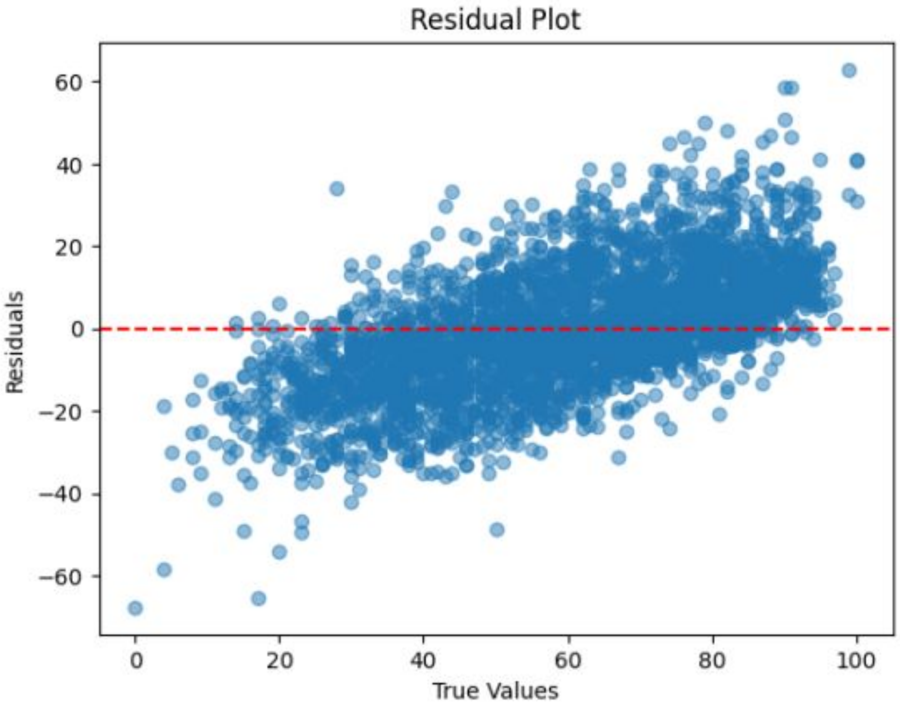
Description:

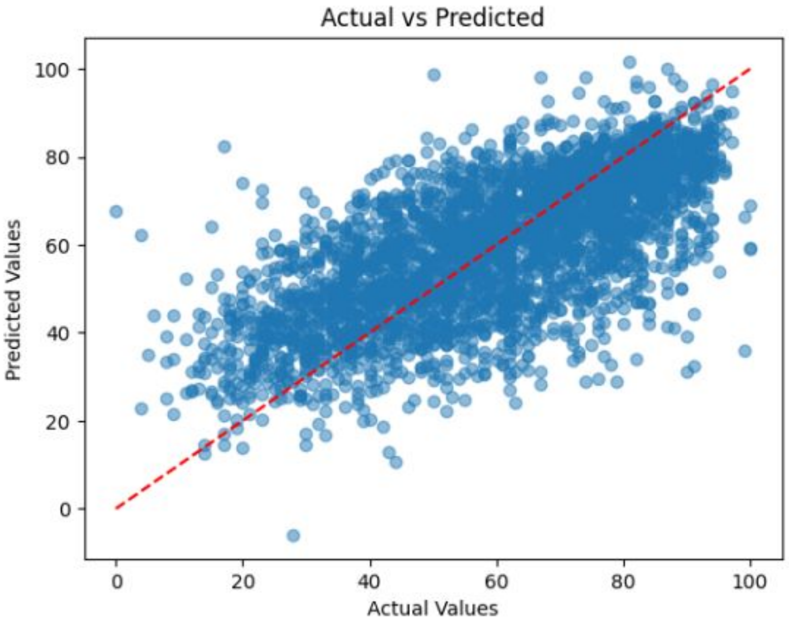
- This plot compares actual audience ratings with the predicted values.
- Points closer to the diagonal line signify better predictions.

Insights:

- The model is reasonably accurate, but deviations from the line highlight areas where predictions can be improved.
- Outliers or extreme residuals could indicate noise in the dataset or limitations in model generalization.

Visualizations





Step 11: Conclusions

1. Best Model:

- Ridge Regression emerged as the best-performing model.

- Achieved the highest R² score and lowest error metrics among all models evaluated.

- Its regularization technique effectively handled multicollinearity and improved generalization.

2. Feature Engineering Impact:

- Transformations like One-Hot Encoding, MultiLabelBinarizer, and outlier removal significantly improved model performance.

- Extracting date features and encoding the top 100 actors contributed to better capturing the underlying patterns in the data.

3. Baseline Comparison:

- All models, including Linear Regression, outperformed the baseline mean prediction.

- Demonstrates the effectiveness of the machine learning pipeline in improving prediction accuracy over simple statistical approaches.

Step 12: Future Scope:

1. Optimization:

- Explore advanced models like Gradient Boosting or Neural Networks for better results.
- Hyperparameter tuning for further improvements in ensemble models.

2. Feature Selection:

- Incorporate additional features, such as user reviews or external popularity metrics.

3. Scalability:

- Test the pipeline on larger datasets and extend the model to predict ratings for unseen movies.

Thank you for the opportunity to share and discuss this project.

I look forward to your feedback and further collaboration!