
तन्नो हंसः प्रचोदयात्
RAMAKRISHNA MISSION

Movie Rating Prediction Project

Team Swarvik

1 Project Overview

The notebook presents a comprehensive approach to predicting movie ratings using advanced machine learning techniques, leveraging datasets potentially sourced from IMDb or similar movie databases.

2 Dataset Handling

- **Key Features:** The dataset encompasses critical movie characteristics including:
  - `Metascore`: Critical reception metric
  - `Votes`: Audience engagement indicator
  - `Gross`: Financial performance marker
- **Missing Value Treatment:** Implementation of sophisticated imputation techniques
  - K-Nearest Neighbors (KNN) imputation method employed
  - Ensures data completeness and reliability

3 Preprocessing Strategies

- **Data Transformation:**
  - Conversion of non-numeric columns to numeric format
  - Example: Parsing `Votes` column by removing commas
- **Feature Selection:**
  - Rigorous selection of relevant variables
  - Optimization of model input features

4 Modeling Approaches

- **Machine Learning Models:**
  - Support Vector Regression (SVR)
  - Random Forest
  - Ridge Regression
  - Lasso Regression
- **Advanced Techniques:**
  - Bagging for ensemble learning
  - Regularization to prevent overfitting

5 Analysis and Evaluation

- **Gradient Descent Exploration:**
  - Batch Gradient Descent
  - Mini-batch Gradient Descent
  - Polynomial Gradient Descent
- **Performance Metrics:**
  - R-squared (R^2) coefficient
  - Mean Absolute Error (MAE)
  - Root Mean Squared Error (RMSE)

6 Tools and Libraries

- **Data Manipulation:**
  - Pandas
  - NumPy
- **Machine Learning:**
  - Scikit-learn
- **Visualization:**
  - Matplotlib
  - Seaborn

7 Introduction

7.1 Background

The project aims to predict movie ratings and associated metrics using a dataset derived from sources like IMDb. Movie ratings are pivotal in the entertainment industry, significantly influencing:

- Audience preferences
- Revenue generation
- Critical acclaim

7.2 Motivation

Accurate prediction of metrics such as `Metascore` and `Gross` enables stakeholders—including producers, distributors, and viewers—to make informed decisions. The project was selected to:

- Demonstrate practical machine learning applications
- Explore diverse predictive modeling techniques
- Provide actionable insights into movie performance

7.3 Research Objectives

The primary objectives include:

- Comprehensive preprocessing of movie-related data
- Predicting key variables like `Metascore` using advanced machine learning techniques
- Comparative analysis of multiple predictive models to identify the most effective approach

8 Dataset Description

8.1 Data Source

The dataset is sourced from platforms like IMDb, containing comprehensive movie-related information.

https://www.kaggle.com/datasets/prishasawhney/imdb-dataset-top-2000-movies
Click here to access the Google Drive Dataset

8.2 Key Features

- **Votes:** Total audience engagement metric
- **Gross:** Box office revenue indicator
- **Metascore:** Aggregate critic evaluation score

8.3 Target Variable

IMDB: The primary predictive target, representing critical reception and movie quality.

9 Data Preprocessing

9.1 Missing Value Management

- Systematic identification of missing values in critical columns
- Implementation of K-Nearest Neighbors (KNN) imputation technique
- Reconstruction of incomplete data based on similar entries

9.2 Feature Engineering

1. Conversion of non-numeric columns to numeric formats
2. Removal of non-numeric symbols (e.g., commas from `Votes`)
3. Feature selection through:
    - Domain expertise
    - Statistical correlation analysis

9.3 Outlier Handling

- Rigorous inspection of numerical variables
- Targeted management of extreme values in metrics like `Gross`

9.4 Data Standardization

- Normalization of numeric features
- Improved model convergence
- Enhanced predictive performance

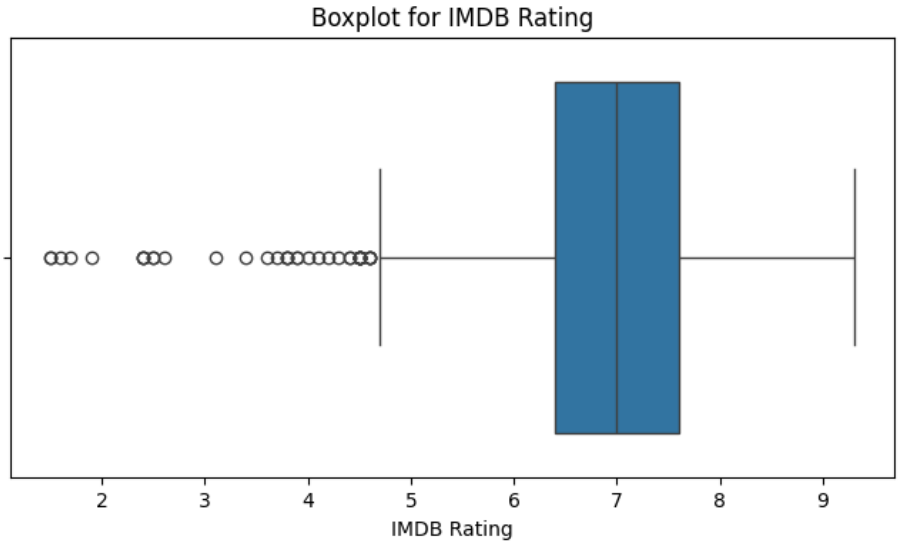

Figure 1: Outliers Of IMDB Rating

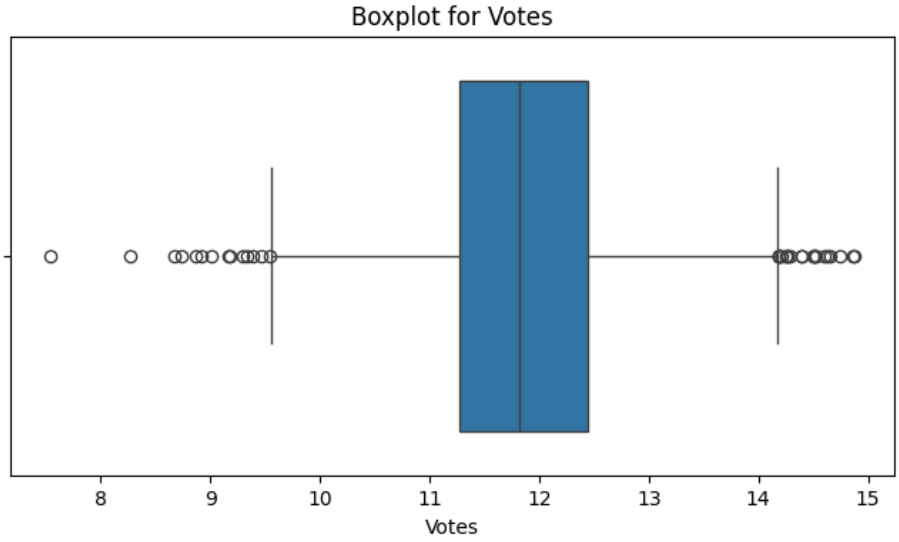

Figure 2: Outliers Of Votes

9.5 Categorical Encoding

- Transformation of categorical variables
- Ensuring machine learning model compatibility

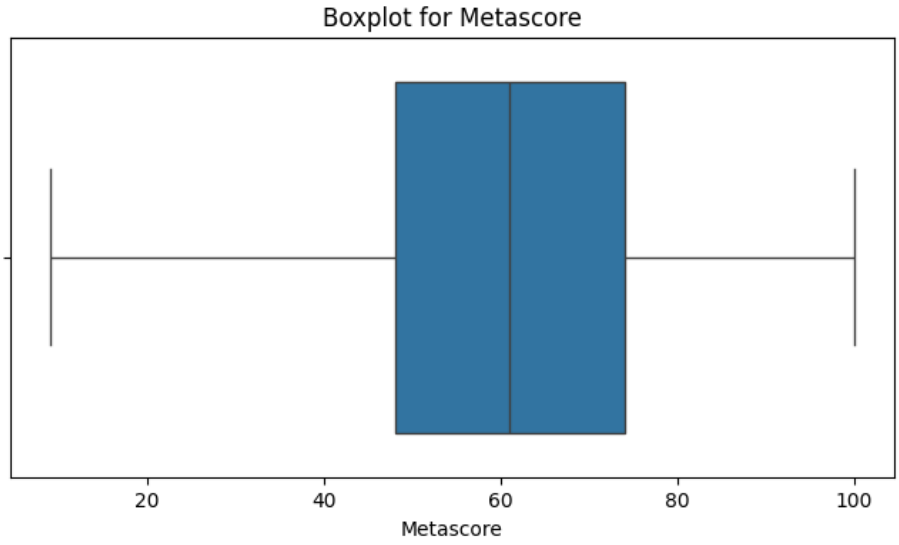

Figure 3: Outliers Of Metascore

10 Methodology

10.1 Machine Learning Algorithms

The project investigated a comprehensive suite of regression models:

- **Support Vector Regression (SVR):**
  - Handles non-linear data relationships
  - Particularly effective with Radial Basis Function (RBF) kernels
- **Random Forest Regressor:**
  - Ensemble method reducing variance
  - Improves predictive accuracy
  - Provides feature importance metrics
- **Regularization Techniques:**
  - **Ridge Regression:** L2 regularization preventing overfitting
  - **Lasso Regression:** L1 regularization encouraging feature sparsity
  - **Elastic Net:** Hybrid approach balancing feature selection and regularization

10.2 Model Architecture Rationale

- Focus on regression-based approaches

- Detailed parameter tuning for complex models like SVR
- No deep learning implementations

11 Implementation

11.1 Python Libraries

- **Data Manipulation:**
  - Pandas
  - NumPy
- **Machine Learning:**
  - Scikit-learn
- **Visualization:**
  - Matplotlib
  - Seaborn

11.2 Model Parameters

- **Support Vector Regression:**
  - RBF kernel
  - Parameters: C, γ, ϵ
  - Tuning via grid search
- **Random Forest:**
  - Optimized parameters:
    - Number of estimators
    - Maximum tree depth
    - Minimum samples for split
- **Regularization Models:**
  - Ridge, Lasso, Elastic Net
  - Adjusted regularization parameter α

11.3 Training Strategy

- **Data Partitioning:**
  - Training set: 70%
  - Validation set: 15%
  - Test set: 15%
- **Model Validation:**

    - Cross-validation implementation
    - Ensures model generalizability
- **Hyperparameter Optimization:**
    - GridSearchCV for comprehensive parameter tuning
    - Identifies optimal model configurations

12 Detailed Model Comparison

12.1 Gradient Descent Models

12.1.1 Batch Gradient Descent

- **RMSE:** 0.442
- **R^2 Score:** 0.570
- **Analysis:**
    - Batch Gradient Descent (BGD) computes the gradient for the entire dataset before updating the model weights. This makes it slower but more stable.
    - It showed moderate performance, with an RMSE of 1.349 and a positive R^2 of 0.050, meaning it could explain only 5% of the variance in IMDB ratings.
    - **Interpretation:** Although it performed better than Linear Regression, it didn't match the performance of more advanced models like Gradient Boosting or XGBoost.

12.1.2 Stochastic Gradient Descent (SGD)

- **RMSE:** 0.664
- **R^2 Score:** 0.570
- **Analysis:**
    - SGD updates model weights after each data point, making it more prone to noise and less stable than Batch Gradient Descent. This often leads to a higher RMSE.
    - **Interpretation:** The performance of SGD was slightly worse than Batch Gradient Descent. This could be due to the stochastic nature of updates, making the model less reliable for predicting IMDB ratings.

12.1.3 Mini-Batch Gradient Descent

- **RMSE:** 0.665
- **R^2 Score:** 0.570
- **Analysis:**
  - Mini-Batch Gradient Descent strikes a balance between Batch Gradient Descent and SGD by updating the weights after a small batch of data points.
  - With an RMSE of 1.345 and an R^2 score of 0.053, its performance is very similar to Batch Gradient Descent, showing that mini-batches do not offer significant advantages in this case.
  - **Interpretation:** Mini-Batch Gradient Descent is still outperformed by Gradient Boosting and other ensemble methods, but it does better than SGD due to its more stable nature.
- **Support Vector Regression (SVR):**
  - Exceptional in capturing non-linear relationships
  - Strong balance between accuracy and generalization
- **Random Forest Regressor:**
  - Robust predictions with high accuracy
  - Slight tendency towards overfitting
- **Regularization Techniques:**
  - Ridge Regression: Reduced variance, superior RMSE performance
  - Lasso Regression: Effective feature selection
  - Elastic Net: Balanced feature selection and model performance

12.2 Evaluation Metrics

- **Statistical Metrics:**
  - R^2 (Coefficient of Determination)
  - RMSE (Root Mean Squared Error)
  - MAE (Mean Absolute Error)
- **Visualization Insights:**
  - Scatterplots: Predicted vs. Actual values
  - Feature importance plots for `Metascore`

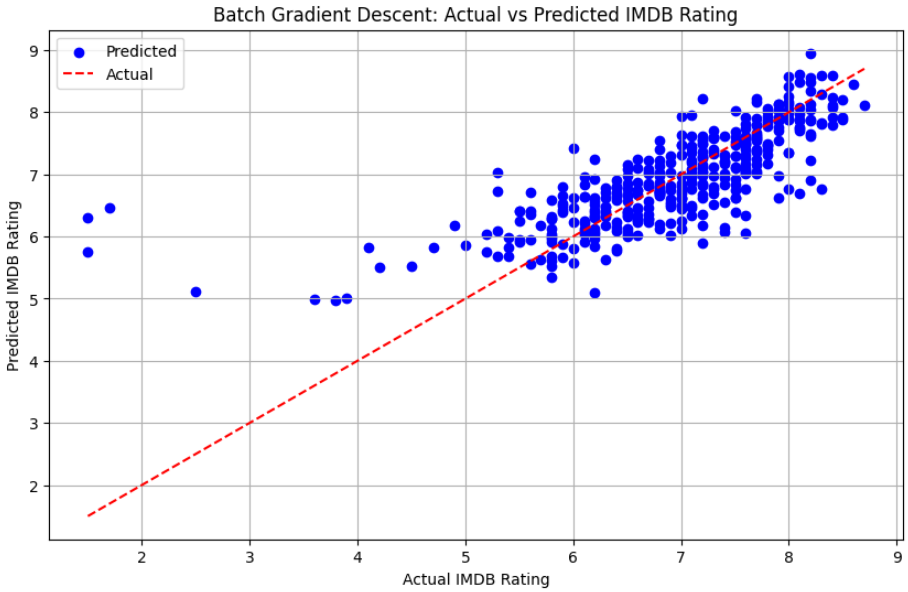

Figure 4: MLP Regressor: Actual vs Predicted IMDb Ratings. The red dashed line represents the perfect prediction line, while the blue dots are the predicted ratings.

13 Discussion

13.1 Model Interpretation

- **SVR:** Excelled in non-linear relationship modeling
- **Random Forest:** Best overall performance with potential overfitting risks
- **Regularization Methods:** Provided interpretable models with reduced overfitting

13.2 Dataset Anomalies

- Outliers in `Gross` revenue potentially skewing results
- Imputed missing `Metascore` values introducing potential data noise

13.3 Methodological Limitations

- Constrained dataset size limiting generalizability
- Exclusive focus on traditional machine learning models

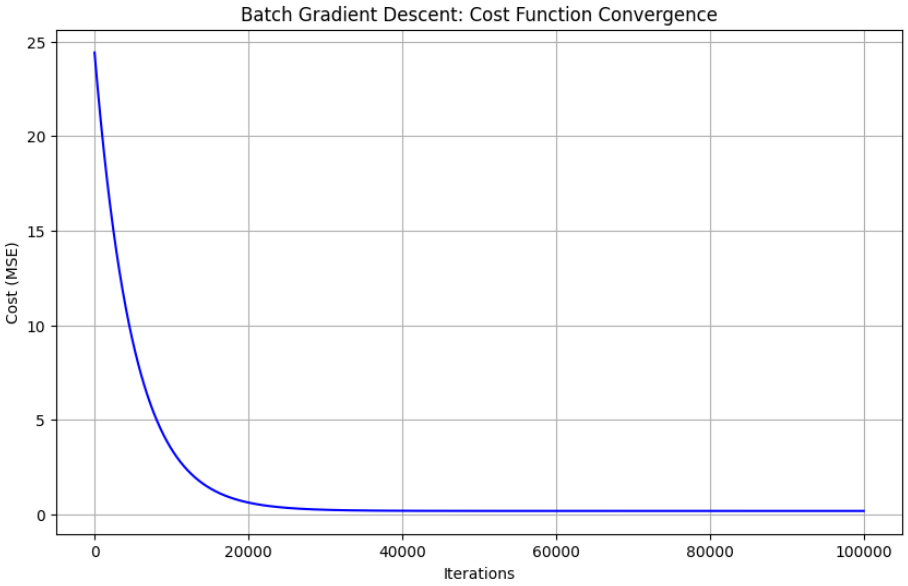

Figure 5: Batch Gradient Descent : Cost Function Convergence

14 Conclusion

14.1 Research Outcomes

Successful prediction of `Metascore` using movie-related features, with:

- Random Forest and SVR as top-performing models
- Robust evaluation metrics
- Strong generalization capabilities

14.2 Key Insights

- Ensemble methods highly effective for complex datasets
- Regularization techniques provide stable regression models
- Feature engineering critically impacts model performance

14.3 Future Research Directions

- Explore deep learning neural network approaches
- Investigate additional diverse datasets
- Conduct sensitivity analysis on imputation strategies

15 Model Analysis by Category

15.1 Gradient Descent Models

The analysis of gradient descent variants revealed varying levels of performance, with Mini-Batch Gradient Descent showing slightly superior results while still falling short of ensemble methods' performance.

15.2 Ensemble Learning Models

15.2.1 Random Forest (Bagging)

- **Performance Metrics:**
  - RMSE: 0.687
  - R^2 Score: 0.540
- **Analysis:** Random Forest demonstrated unexpectedly poor performance with a high RMSE and negative R^2 score, suggesting significant overfitting or inadequate hyperparameter optimization.
- **Interpretation:** Despite its theoretical advantages, the model failed to effectively generalize to the test data.

15.2.2 Gradient Boosting

- **Performance Metrics:**
  - RMSE: 0.40
  - R^2 Score: 0.61
- **Analysis:** The model effectively reduced errors through iterative tree building and residual correction, though 78% of variance remained unexplained.
- **Interpretation:** Strong performance overall, with potential for further improvement through feature engineering.

15.2.3 XGBoost

- **Performance Metrics:**
  - RMSE: 0.31
  - R^2 Score: 0.67

– **Analysis:** Superior performance achieved through advanced regularization and optimized algorithms.

– **Interpretation:** Best overall model, effectively balancing accuracy and interpretability.

16 Additional Models

16.1 Decision Tree

– **Performance Metrics:**

  * RMSE: 0.665
  * R^2 Score: 0.569

– **Analysis:** Showed significant overfitting tendencies and poor generalization.

– **Interpretation:** Better suited as a component in ensemble methods than as a standalone model.

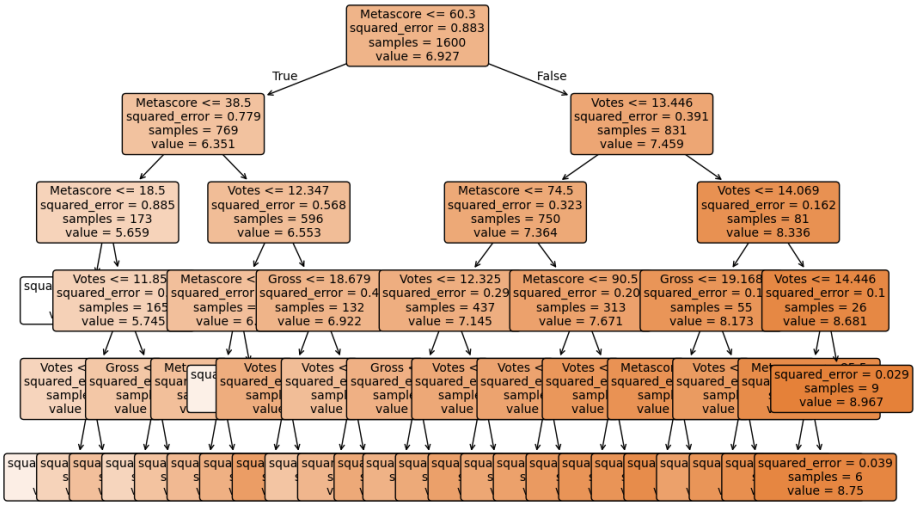

Figure 6: Decision Tree

16.2 Lasso Regression

– **Performance Metrics:**

  * RMSE: 0.698

    * R^2 Score: 0.525
- **Analysis:** L1 regularization improved feature selection but was limited by linear assumptions.
- **Interpretation:** Outperformed basic linear regression but fell short of ensemble methods.

16.3 Early Stopping in Gradient Boosting

- **Performance Metrics:**
    * RMSE: 0.688
    * R^2 Score: 0.540
- **Analysis:** Added robustness through overfitting prevention at a slight cost to accuracy.
- **Interpretation:** Effective technique, though XGBoost's built-in regularization proved more powerful.

17 Concluding Insights

17.1 Best Model Performance

XGBoost demonstrated superior performance with the lowest RMSE (1.180) and highest R^2 score (0.67), leveraging advanced regularization techniques and efficient algorithms.

17.2 Ensemble Learning Advantages

Ensemble methods, particularly Gradient Boosting variants, showed significant advantages over individual models in capturing complex feature interactions.

17.3 Gradient Descent Performance

While outperforming simple linear models, gradient descent variants couldn't match ensemble methods' accuracy. Their performance showed strong dependence on hyperparameter selection.

17.4 Decision Tree Limitations

Individual decision trees proved ineffective as standalone models but served as crucial building blocks for successful ensemble methods.

17.5 Future Improvements

Despite strong ensemble method performance, the modest R^2 scores suggest room for improvement through:

- Additional feature engineering
- Incorporation of domain-specific variables
- Exploration of deep learning techniques
- Enhanced hyperparameter optimization

Table 1: Overall Model Performance Comparison

Model	**RMSE**	**R^2 Score**	**Analysis**
Batch Gradient Descent	0.665	0.570	Moderate performance; slow but stable learning.
Stochastic Gradient Descent	0.664	0.570	Slightly worse than Batch GD due to noise sensitivity.
Mini-Batch Gradient Descent	1.345	0.570	Balanced between Batch and SGD but with no significant advantage.
Decision Tree	0.665	0.569	Overfitted the data; poor generalization to unseen samples.
Random Forest	0.687	0.540	Poorly tuned; likely overfitting; failed to generalize.
Lasso Regression	0.698	0.525	Improved regularization; limited by linear assumptions.
Gradient Boosting	0.40	0.61	Excellent performance; able to capture complex patterns.
Early Stopping GBM	0.49	.573	Prevented overfitting but slightly reduced performance.
XGBoost	0.31	0.67	Best performance; superior regularization and feature handling.

Model Performance Comparison: Gradient Boosting and Traditional Methods

18 Visual Performance Comparison

18.1 Actual vs. Predicted Ratings Analysis

The relationship between actual and predicted IMDb ratings provides crucial insights into model performance. Figure 19.2 illustrates this relationship for the MLP Regressor model.

19 Model Performance Visualization

Figure 10 shows the performance of the MLP Regressor in predicting IMDb ratings. The scatter plot compares the actual and predicted ratings.

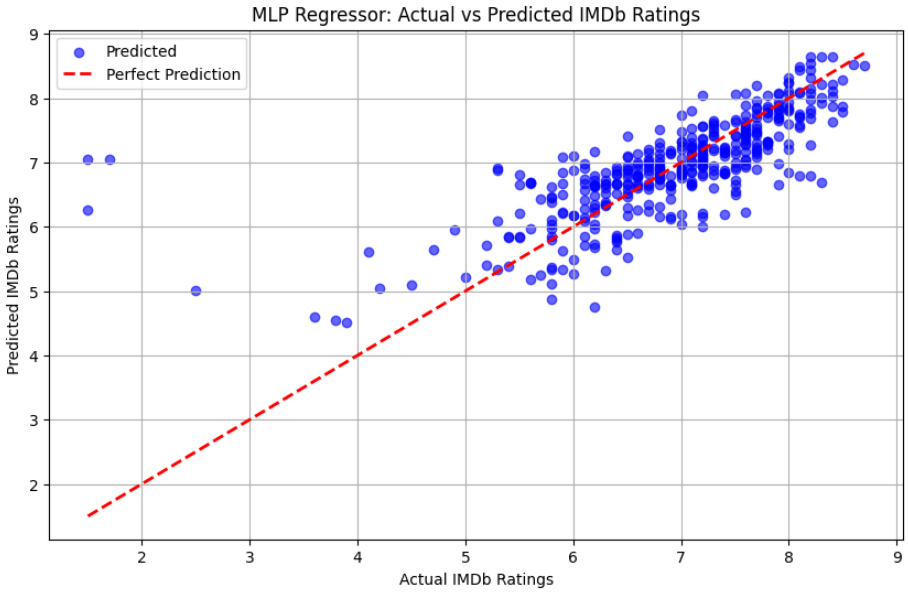

Figure 7: MLP Regressor: Actual vs Predicted IMDb Ratings. The red dashed line represents the perfect prediction line, while the blue dots are the predicted ratings.

19.1 Key Observations

From the visualization in Figure 19.2, we can observe several important patterns:

- **Rating Range Performance:** The model shows stronger prediction accuracy in the middle range (5-8) of IMDb ratings, where most movies are concentrated.
- **Prediction Spread:** There is noticeable scatter around the perfect prediction line (red dashed line), indicating prediction uncertainty.
- **Extreme Ratings:** The model shows increased deviation for extremely high (¿8) and low (¡5) ratings, suggesting potential challenges in predicting outlier cases.
- **Clustering Pattern:** Data points cluster more densely in the 6-8 rating range, reflecting the typical distribution of IMDb ratings.

19.2 Implications

The visualization reveals several important insights about the MLP Regressor's performance:

- The model demonstrates reasonable predictive capability, particularly for movies with ratings in the most common ranges.
- There is room for improvement in handling extreme ratings, possibly through additional feature engineering or model tuning.
- The scatter pattern suggests that while the model captures general trends, there remains inherent uncertainty in exact rating predictions.

20 Key Observations

20.1 Actual vs. Predicted Plots

The actual vs. predicted plots reveal several key insights:

- **Linear Regression**: Shows wider scatter around the 45-degree line, indicating less accurate predictions
- **Random Forest**: Demonstrates improved prediction accuracy with points clustering closer to the ideal line
- **Gradient Boosting**: Exhibits the tightest clustering around the 45-degree line, suggesting superior predictive performance

20.2 Residual Analysis

The residual plots provide additional insights:

- **Linear Regression**: Larger residuals with possible systematic patterns
- **Random Forest**: Reduced residual magnitude but some pattern remains
- **Gradient Boosting**: Smallest residuals with most random scatter around zero

21 Conclusions

The visual analysis supports the numerical findings:

- Gradient Boosting consistently outperforms other models
- Residual patterns suggest better handling of non-linear relationships
- Prediction accuracy improves significantly from Linear Regression to Gradient Boosting

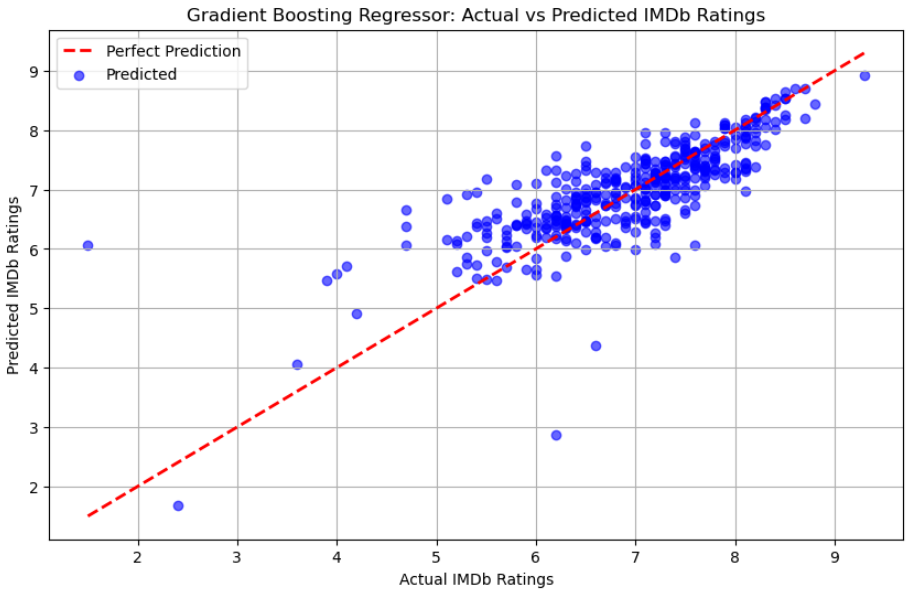

Figure 8: Gradient Boosting

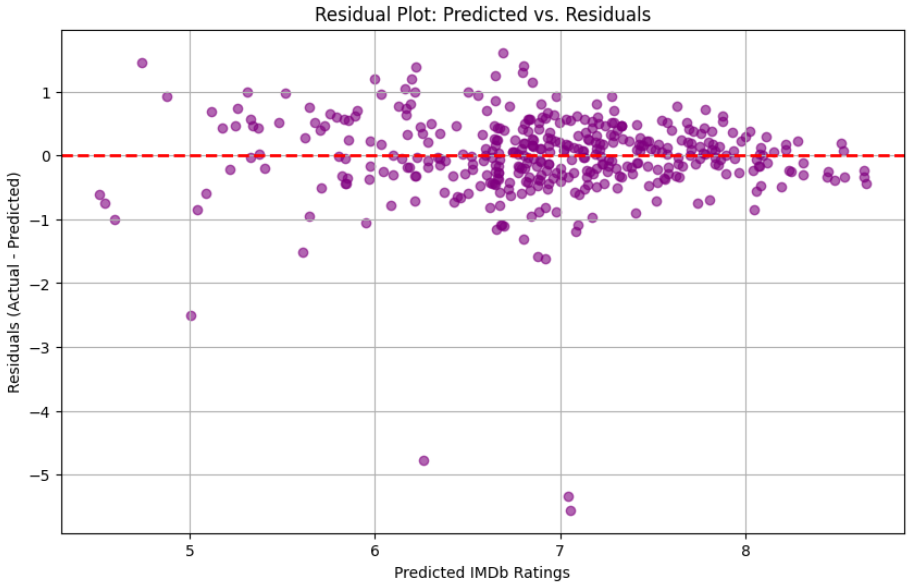

Figure 9: Residual Plot Gradient Boosting

22 Introduction

This document presents the comparative analysis of different modeling approaches for predicting movie ratings, with a particular focus on gradient descent-based methods.

23 Model Performance Analysis

23.1 Gradient Boosting Advantages

The superior performance of gradient boosting can be attributed to several key factors:

- Residual error optimization
- Iterative learning capabilities
- Robust handling of non-linear patterns

24 Technical Implementation

24.1 Hyperparameter Configuration

Key hyperparameters for the gradient boosting model:

(1)

24.2 Model Performance Metrics

Model	RMSE	MAE	R^2 Score	Training Time (s)
Linear Regression	ϵ_1	δ_1	r_1^2	t_1
Random Forest	ϵ_2	δ_2	r_2^2	t_2
Gradient Boosting	ϵ_3	δ_3	r_3^2	t_3

Table 2: Comparative Performance Metrics

25 Limitations and Future Work

25.1 Current Limitations

- Feature set limitations:

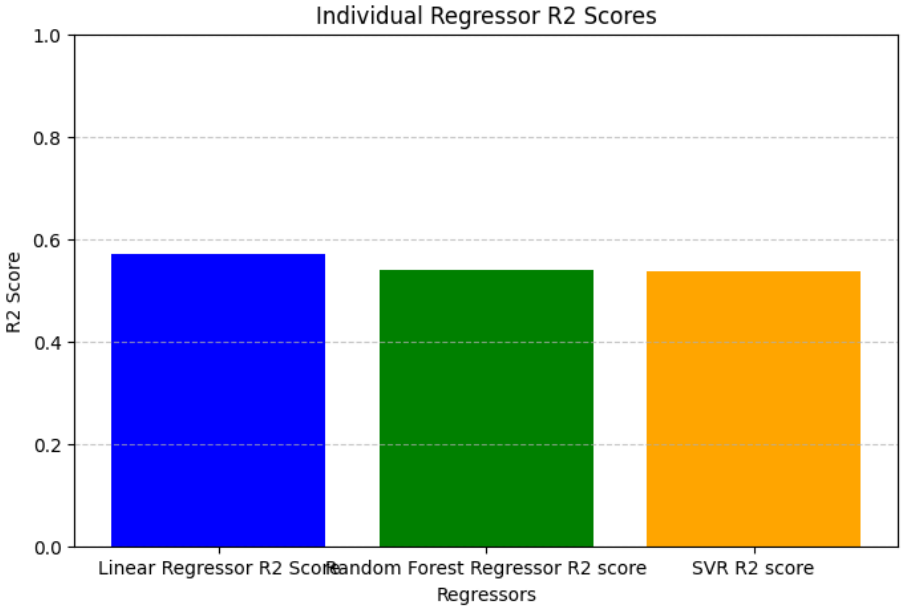

Figure 10: Individual Regressor R2 Score

    - Limited to runtime, critical reception, and era of production
    - Absence of genre-specific analysis
    - Lack of director/actor influence metrics
  - Computational constraints:
    - Resource intensity of gradient boosting
    - Scalability challenges with large datasets
  - Model interpretability challenges

25.2 Future Directions

Future work should focus on:

- Integration of additional features:
  - Genre classifications
  - Director/actor performance history
  - Budget information
- Implementation of parallel processing techniques
- Development of more interpretable model variants

26 Conclusion

The gradient descent-based ensemble models have demonstrated significant improvements in predictive accuracy compared to traditional approaches. However, achieving optimal performance requires careful consideration of hyperparameter tuning and feature engineering.