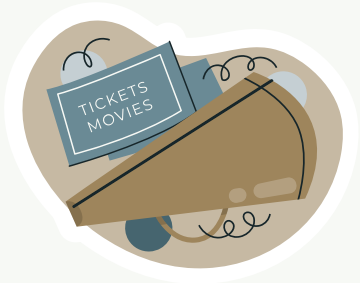
Symbiosis Institute of Geoinformatics

॥वसुधैव कुटुम्बकम्॥

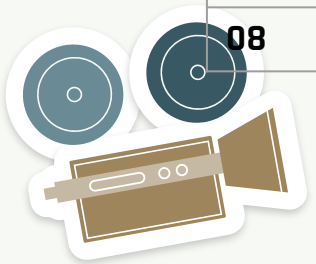
Understanding Movie Popularity : A Machine Learning Perspective

Name : Sharon Furtado
PRN : 23070243015
Course : M.Sc Data Science and Spatial Analytics
Internal Supervisor : Mr. Sahil Shah

Table of contents

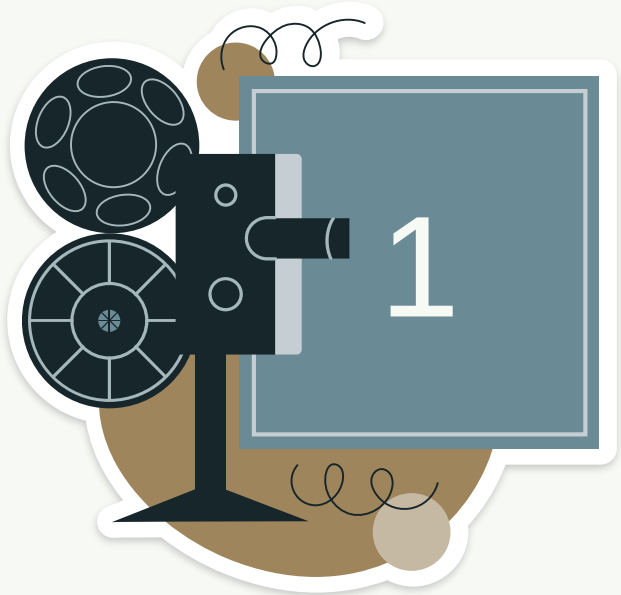


॥वसुधैव कुटुम्बकम्॥

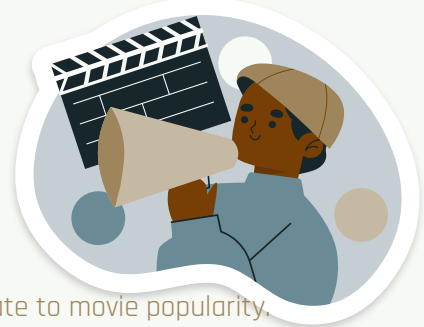


Introduction

- The film industry is a global cultural and economic powerhouse, making movie success prediction vital for stakeholders.
- Understanding and predicting movie popularity aids in informed decision-making during production and distribution.
- Traditional success prediction methods rely on limited metrics and subjective insights.
- This project utilizes machine learning to objectively predict movie popularity using features like genre, cast, budget, and more, offering a data-driven approach to understanding film success.

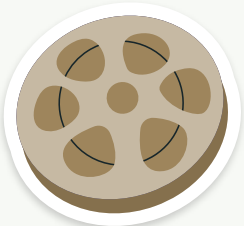


Research Objective



॥वसुधैव कुटुम्बकम्॥

- **To investigate the key factors influencing movie popularity:**
  - Analyze how genre, duration, budget, vote count, and vote average relate to movie popularity.
  - Identify which of these factors are the strongest predictors of a movie's success.
- **Apply and evaluate machine learning models for predicting movie popularity:**
  - Compare the performance of various models such as Random Forest, SVM, and Neural Networks.
  - Determine which models provide the highest accuracy in predicting popularity based on selected features.
- **Assess the impact of specific movie characteristics on prediction accuracy:**
  - Evaluate the individual and combined effects of genre, budget, and vote count on the accuracy of predictions.
  - Establish which movie characteristics most significantly influence the predictive power of the models.
- **Develop a high-accuracy machine learning model for movie popularity:**
  - Optimize machine learning models using key features to maximize predictive accuracy.
  - Test and refine the models for reliable performance on data from TMDB.
- **Explore the applicability of prediction models for future movie releases:**
  - Test the models' ability to predict the popularity of upcoming movies using pre-release data.
  - Validate the generalization capacity of the models for new and unseen movie data.



Literature Review

॥वसुधैव कुटुम्बकम्॥

SR. No	References	Objective	Technique	Dataset	Results
1	Agarwal et al., 2022	To predict movie success and compare various statistical approaches	Utilised regression models, clustering techniques, time series analysis, and an artificial neural network	IMDb dataset for movies released in 2020	Achieved an accuracy of 86% with the artificial neural network; identified key features influencing movie success.
2	Oyewola & Dada (2022)	To compare the effectiveness of various machine learning techniques for predicting movie popularity.	Multinomial Logistic Regression (M.L.R.), (SVM), (B.A.G.), (N.B.S.), (K.N.N.) noise filtering using All K-Edited Nearest Neighbors(A.E.N.N.).	The dataset consisted of 5043 rows & 14 columns,including features like director, duration, and IMDb scores.	B.A.G. performed the best, with an accuracy of 99.56%, while other methods showed lower effectiveness.

Literature Review

॥वसुधैव कुटुम्बकम्॥

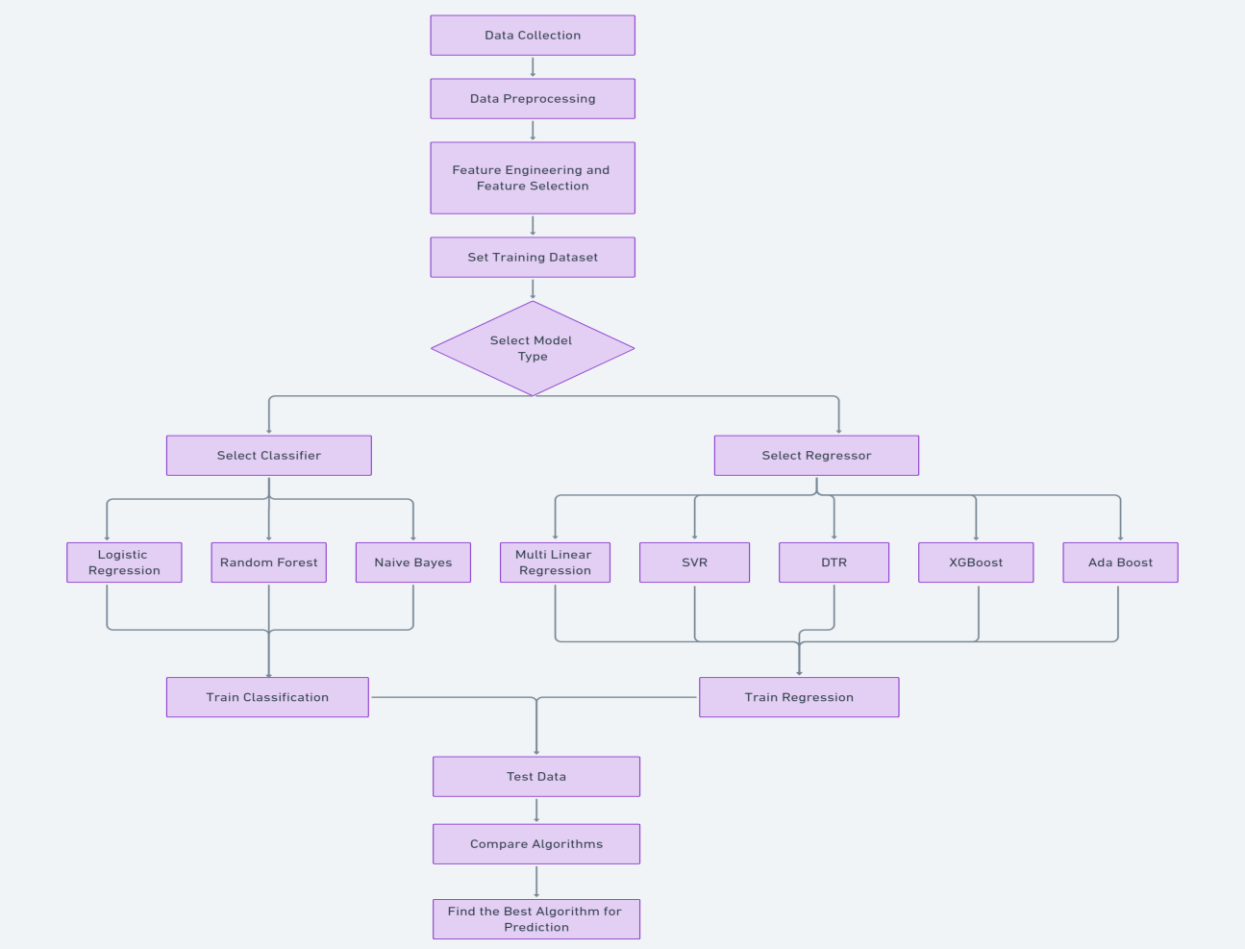
SR. No	References	Objective	Technique	Dataset	Results
3	Verma, H. & Verma, G. (2019)	To predict the success of Bollywood movies using machine learning techniques.	Comparative analysis of 6 ML algorithms: (D.T.), (R.F.), (SVM), (L.R.), (A.T.B.), (ANN).	Sample of 200 Bollywood movies.	Models achieved 80% to 92%, with R.F. and L.R. models achieving 92% accuracy. Significant predictors identified were music rating, IMDb rating, and number of screens for release.

Literature Review

॥वसुधैव कुटुम्बकम्॥

SR. No	References	Objective	Technique	Dataset	Results
4	Anand, A. (2023)	To analyse the impact of machine learning algorithms on predicting movie success.	Utilised machine learning models such as Linear Regression, Decision Trees, and Random Forests for data analysis	Kaggle dataset containing information on 1000 IMDB movies.	Random Forest achieved a 74% accuracy in predicting IMDB ratings; Decision Trees had 71%, and Logistic Regression 53.56%.

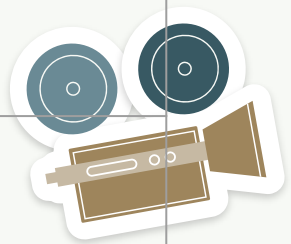
Methodology



Dataset Description

TMDB dataset with 5,000 entries, 16 variables for popularity prediction.

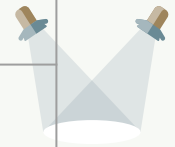
Sr no	Feature Name	Description	Datatype
1	Title	The title of the movie (e.g., "Bad Boys: Ride or Die").	Object
2	Genre	Genres of the movie listed in a comma-separated format (e.g., "Action, Crime, Thriller, Comedy").	Object (Categorical)
3	Release Date	The release date of the movie in the format YYYY-MM-DD.	Object
4	Duration	The length of the movie in minutes (e.g., 115).	Numerical
5	Language	The primary language of the movie represented by language codes (e.g., "en" for English).	Object (Categorical)

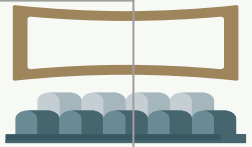


Dataset Description

॥वसुधैव कुटुम्बकम्॥

Sr no	Feature Name	Description	Datatype
6	Country	The country or countries where the movie was produced (e.g., "United States of America").	Object
7	Director	The director(s) of the movie (e.g., "Adil El Arbi, Bilall Fallah").	Object
8	Overview	A brief summary of the movie plot.	Object
9	Popularity	A numerical score indicating the movie's popularity on the TMDB platform.	Numerical
10	Tagline	A short promotional phrase or tagline associated with the movie.	Object



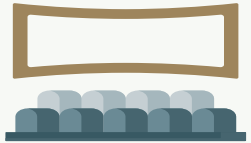


Dataset Description

Sr no	Feature Name	Description	Datatype
11	Revenue	The country or countries where the movie was produced (e.g., "United States of America").	Numerical
12	Budget	The director(s) of the movie (e.g., "Adil El Arbi, Bilall Fallah").	Numerical
13	Vote Average	A brief summary of the movie plot.	Numerical
14	Vote Count	A numerical score indicating the movie's popularity on the TMDB platform.	Numerical
15	Production Companies	A short promotional phrase or tagline associated with the movie.	Object

Dataset Description

Sr no	Feature Name	Description	Datatype
16	Actors	The main actors featured in the movie, listed in a comma-separated format. Eg(Ryan Reynolds, Hugh Jackman, Emma Corrin, Matthew Macfadyen, Dafne Keen, Jon Favreau, Morena Baccarin, Rob Delaney, Leslie Uggams, Jennifer Garner)	Object



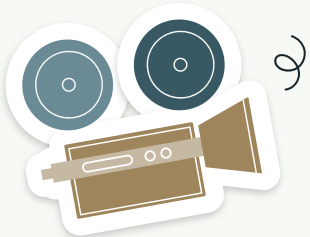
Methodology Overview

Data Preprocessing: Managed missing values, encoded categorical variables, and normalized data using transformations to prepare for modeling.

Feature Selection: Identified key features influencing movie popularity using correlation and feature importance scores.

Model Building:

Regression: Built Multilinear Regression, SVR, DTR, XGBoost, and AdaBoost models.
Classification: Developed Logistic Regression, Random Forest, and Naive Bayes models.
Ensemble Learning: Combined top models to enhance accuracy and prediction robustness.



Model Evaluation:

Regression: Evaluated using R^2, MSE, and RMSE.
Classification: Assessed using accuracy, precision, recall, and F1-score.

Random Forest

Aspect	Details
Algorithm	**RandomForestClassifier** from **'sklearn.ensemble'**
Parameters	**'n_estimators=100'**
Equation	$\hat{C} = \text{argmax}_C \left(\sum_{i=1}^{N} I \ (T_i(X) = C) \right)$ Where $\boldsymbol{X}$ is the input feature vector. $T_i(X)$ is the prediction of the i-th tree. C represents a possible class label. I is the indicator function, which equals 1 if the tree predicts class C and 0 otherwise.
Working	Combines multiple decision trees to improve predictions, offering robustness against outliers and irrelevant inputs. (Verma, H. & Verma, G. (2019))

Logistic Regression

Aspect	Details
Algorithm	LR using **'LogisticRegression'** from **'sklearn.linear_model'**
Parameters	**'max_iter=1000', 'random_state=42'**
Equation	**Logistic Function:** $P(y = 1 \mid X) = \frac{1}{1+e^{-(w \times X + b)}}$
Working	Applies for binary classification by estimating the probability of event occurrence, using statistical tests like Wald's test for significance and Pseudo R^2 for model comparison.

(Agarwal et al., 2022)

Naive Bayes

Aspect	Details
Algorithm	Naive Bayes using **'GaussianNB'** from **'sklearn.naive_bayes'**
Equation	**Bayes' Theorem:** $P(y \mid X) = \frac{P(X \mid y)P(y)}{P(X)}$
Working	Uses a probabilistic approach to classify data by estimating the posterior probability and selecting the class with the highest probability based on the MAP rule.

(Oyewola & Dada (2022))

Support Vector Regressor

॥वसुधैव कुटुम्बकम्॥

Aspect	Details
Algorithm	using **‘SVR’** from **‘sklearn.svm’**
Parameters	**‘kernel='linear', 'poly', 'rbf', 'sigmoid'**
Equation	**Objective Function:** $\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ w = weight vector b = bias term ξ_i, ξ_i^* = slack variables (for over- and under-estimations) C = regularization parameter (controls trade-off between flatness of the function and tolerance to errors) **Decision Function:** $f(x) = \mathbf{w} \cdot \mathbf{x} + b$ **Subject to:** $y_i - (\mathbf{w} \cdot \mathbf{x_i} + b) \leq \epsilon + \xi_i^*$ $(\mathbf{w} \cdot \mathbf{x_i} + b) - y_i \leq \epsilon + \xi_i$
Working	Supervised learning algorithm for regression. Finds optimal hyperplane that maximizes margin and minimizes error.
Kernel Trick	Transforms data into higher dimensions using kernel functions (e.g., Linear, Polynomial, RBF, Sigmoid) without explicitly computing coordinates.

MultiLinear Regression

Aspect	Details
Algorithm	LinearRegression from **'sklearn.linear_model'**
Parameters	**'n_estimators=100'**
Equation	$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$ y = dependent variable (target) $\beta_1, \beta_2, \ldots \ldots \beta_n$ = coefficients for each feature β_0 = intercept term (features) $x_1, x_2 \ldots . x_n$ = independent variables ϵ = error term (residual)
Working	Uses multiple independent variables to estimate the target, applies VIF and statistical tests for feature selection, and checks significance with OLS and hypothesis tests.

(Agarwal et al., 2022)

Decision Tree Regressor

Aspect	Details
Algorithm	DecisionTreeRegressor from **'sklearn.tree'**
Parameters	**'n_estimators=100'**
Equation	$y = \frac{1}{N_i} \sum_{j \in N_i} y_j$ ○ y = predicted value (mean of target values within region N_i) ○ N_i = number of observations in the j^{th} region ○ y_j = target value for the j^{th} observation in that region
Working	Builds a model by splitting the data into subsets based on feature values, creating a tree structure where each node represents a decision, and predicts values by averaging outcomes in the terminal nodes.

XGBoost

Aspect	Details
Algorithm	From xgboost using XGBRegressor
Parameters	**'n_estimators=100'**
Equation	$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$ ○ $l(y_i, \hat{y}_i)$ = loss function (e.g., Mean Squared Error) ○ $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2$ ▪ T = number of leaves ▪ λ = regularization term to prevent overfitting
Working	Uses gradient boosting with decision trees to optimize prediction accuracy, incorporates regularization to prevent overfitting, and iteratively refines models based on residuals.

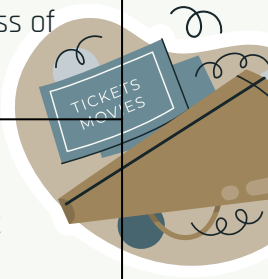
Adaboost

Aspect	Details
Algorithm	AdaBoostRegressor from **'sklearn.ensemble'**
Parameters	**'n_estimators=100'**
Equation	**Update Rule for Weights:** $w_{i+1} = w_i \cdot e^{\alpha \cdot (1 - y_i f(x_i))}$ w_i= weight of the i^{th} data point α = model's coefficient, which depends on model error
	Final Prediction: $F(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t f_t(x\right)\right)$ $f(x_i)$ = predicted value from weak learner y_i = true value
Working	Uses ensemble learning with multiple weak learners to improve predictions, combines them through weighted voting, and adjusts weights to minimize errors iteratively.

Key Metrics and Their Significance

॥वसुधैव कुटुम्बकम्॥

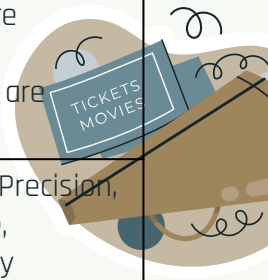
Metric	Description	Formula	Significance
Precision	Ratio of correctly predicted positive observations to total predicted positives.	$\text{Precision} = \frac{\text{True Positives (T.P.)}}{\text{True Positives (T.P.)} + \text{False Positives (F.P.)}}$	Important when the cost of false positives is high, as it indicates the correctness of positive predictions.
Recall (Sensitivity or True Positive Rate)	Fraction of true positive predictions among all positive instances.	$\text{Recall} = \frac{\text{True Positives (T.P.)}}{\text{True Positives (T.P.)} + \text{False Negatives (F.N.)}}$	Critical when missing a positive instance (false negative) is costly, as it measures the model's ability to detect positive instances.
F1-Score	Harmonic mean of Precision and Recall, balancing between the two metrics.	$\text{F1–Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Useful when both Precision and Recall are important, especially in the presence of an unequal class distribution.



Key Metrics and Their Significance

॥वसुधैव कुटुम्बकम्॥

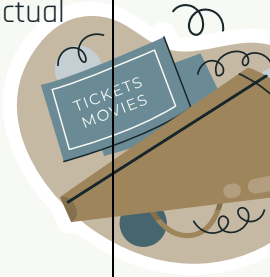
Metric	Description	Formula	Significance
Accuracy	Ratio of total correct predicted observations to total observations.	**Accuracy** $= \frac{\textbf{True Positives (TP) + True Negatives (TN)}}{\textbf{Total Observations}}$	Popular metric, but less informative on imbalanced datasets where more detailed metrics like Precision and Recall are needed.
Support	Total number of occurrences of each class in the dataset.	**Support = Number of true instances for a given class**	Provides context to Precision, Recall, and F1- Score, explaining how many instances of each class were present during evaluation.
R2 Score	Represents the proportion of the variance in the dependent variable that is predictable from the independent variables.	$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$ y_i = actual value $\hat{y}_i$ = predicted value $\bar{y}$ = mean of actual values n = number of data points	High R^2: Good fit; Low R^2: Poor fit; does not ensure model performance or prevent overfitting.



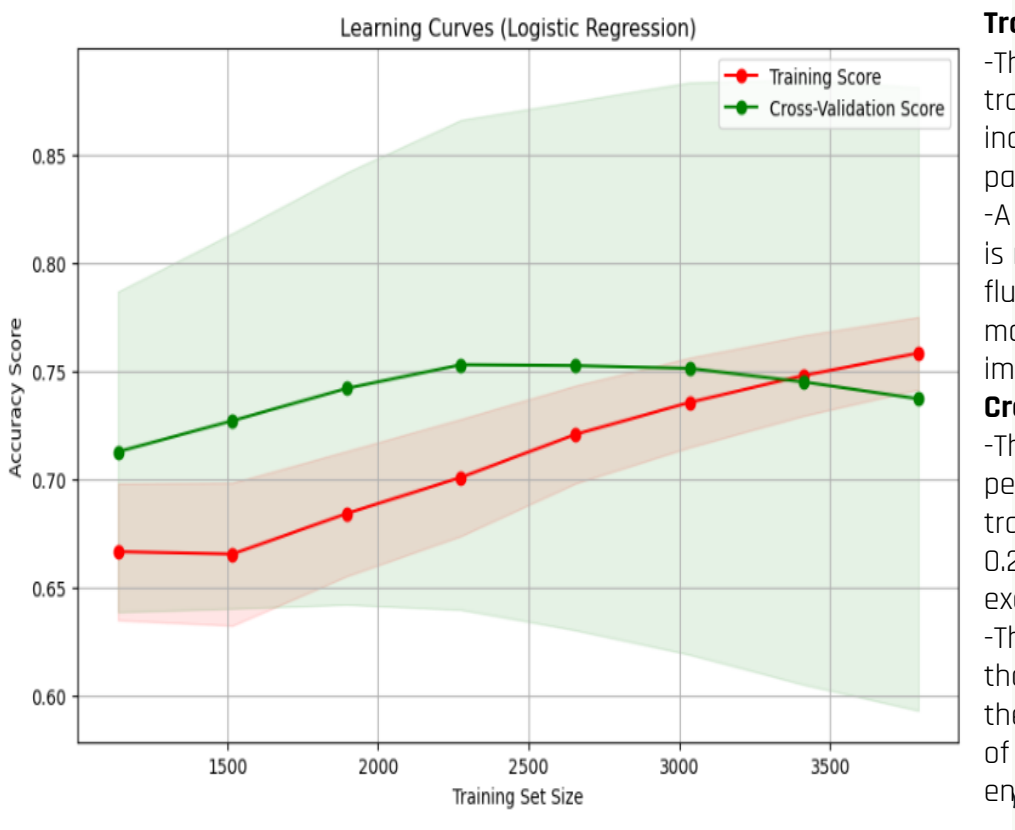
Key Metrics and Their Significance

॥वसुधैव कुटुम्बकम्॥

Metric	Description	Formula	Significance
RMSE	Measures the average magnitude of prediction errors by calculating the square root of the average of squared differences between predicted and actual values.	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ ○ y_i = actual value ○ $\hat{y}_i$ = predicted value ○ n = number of data points	Lower RMSE values indicate better model accuracy and closer predictions to the actual outcomes.



Result - Logistic Regression



Training Score (Red Line):

-The R^2 score of the training set initially drops as the number of training examples increases but stabilizes at around 0.6. This indicates that the model is effectively learning the underlying patterns in the data without perfectly capturing all the complexities.

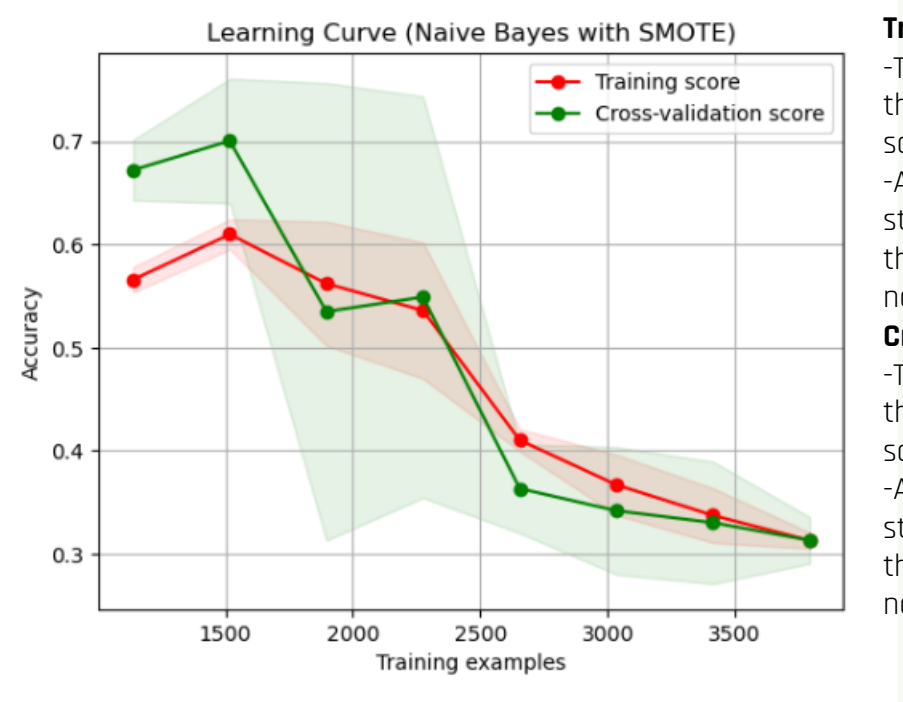
-A stable training score slightly below 0.6 suggests that the model is not overfitting, meaning it avoids learning noise or random fluctuations specific to the training set. This balance shows that the model captures significant trends in the data but leaves room for improvement in its predictive power.

Cross-Validation Score (Green Line):

-The cross-validation score, which measures how well the model performs on unseen data, remains consistently lower than the training score. It starts around 0.3 and gradually stabilizes near 0.25, indicating the model's struggle to generalize to new, unseen examples.

-This lower cross-validation score points to underfitting, suggesting that the model may be too simplistic to capture the complexity of the movie popularity prediction task. It reflects the high bias nature of the model, where it misses key predictive features that could enhance accuracy.

Result - Naïve Bayes



Training Score:

-The R^2 score for the training data begins relatively high, indicating that the model initially fits the smaller dataset well, possibly capturing some specific patterns or noise.

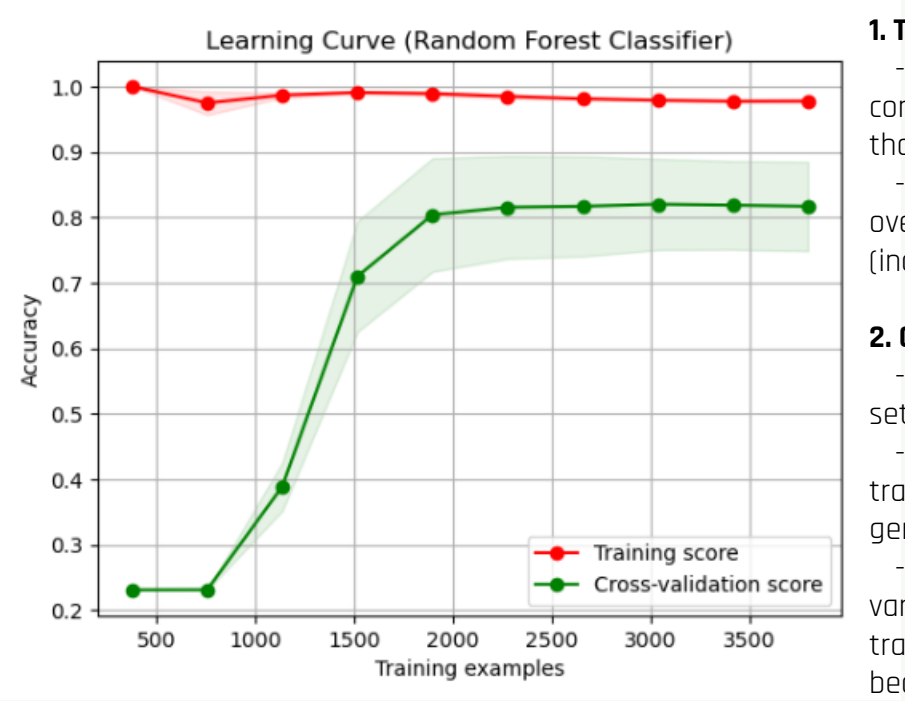
-As the training set size increases, the score gradually decreases and stabilizes, reflecting a reduction in overfitting. This trend suggests that the model is starting to generalize better, moving away from learning noise to focusing on the broader, more relevant patterns in the data.

Cross-Validation Score

-The R^2 score for the training data begins relatively high, indicating that the model initially fits the smaller dataset well, possibly capturing some specific patterns or noise.

-As the training set size increases, the score gradually decreases and stabilizes, reflecting a reduction in overfitting. This trend suggests that the model is starting to generalize better, moving away from learning noise to focusing on the broader, more relevant patterns in the data.

Result - RandomForest Classifier



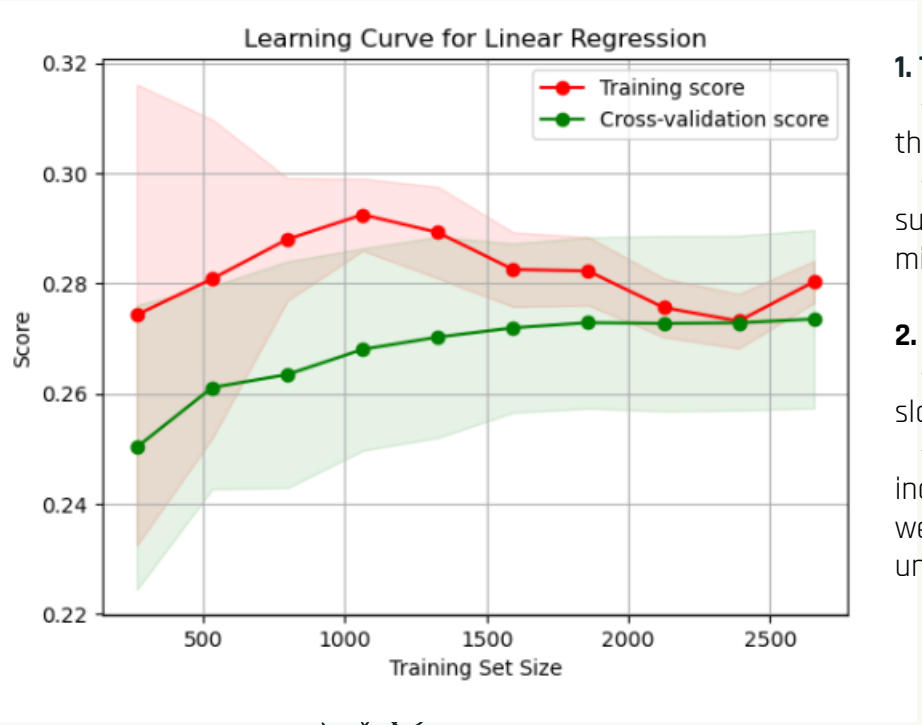
1. Training Score (Red Line):

- The training score is very high (close to 1.0) and remains nearly constant as the number of training examples increases. This indicates that the Random Forest model fits the training data extremely well.
- Since it's almost at 100%, this suggests that the model is potentially overfitting on the training set, meaning it's capturing even small details (including noise) perfectly in the training data.

2. Cross-Validation Score (Green Line):

- The cross-validation score starts low (around 0.2) with a small training set but improves rapidly as more data is introduced.
- The score stabilizes at around 0.8-0.85 after approximately 2000 training examples. This is a good indicator that the model is learning general patterns in the data but still leaving room for improvement.
- The shaded region around the cross-validation score indicates the variance (instability) of the model's performance. It narrows as more training examples are used, suggesting that the model's performance becomes more reliable with larger datasets.

Result - MultiLinear Regression



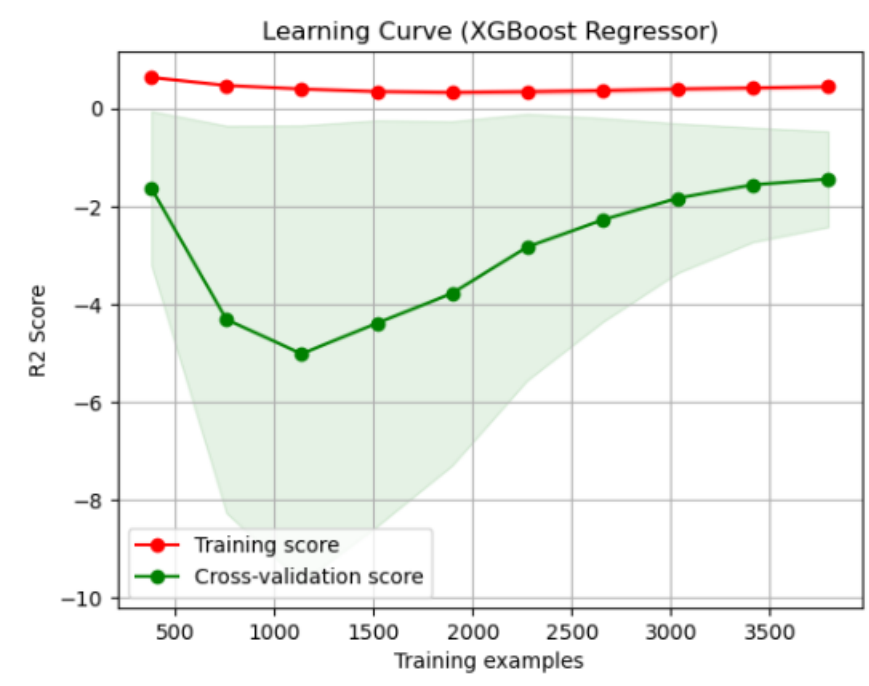
1. Training Score (Red Line):

- The training score starts relatively high and increases slightly as the training set size grows.
- After a certain point, it plateaus and fluctuates slightly. This suggests that the model is able to fit the training data quite well, but might not be capturing more complex patterns as the dataset grows.

2. Cross-Validation Score (Green Line):

- The cross-validation score starts lower than the training score and slowly increases with more data.
- Even though it improves, it remains lower than the training score, indicating a *generalization gap*, which means the model performs well on the training data but doesn't generalize as effectively to unseen data.

Results - XGBoost



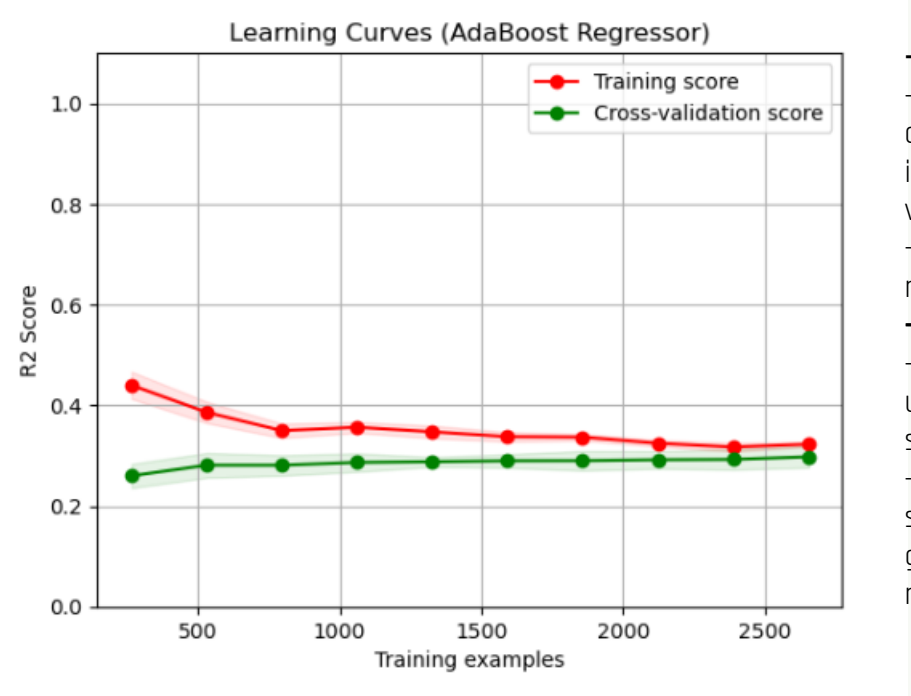
1. Training Score (Red Line):

- The training R^2 score is consistently close to 0 across all training set sizes, indicating that the model is not performing well on the training data. In regression tasks, an R^2 score of 0 suggests that the model is only as good as the mean prediction (it's not capturing any of the variance in the data).
- The lack of improvement in the training score with more data suggests that the model is underfitting the training data. This could be due to improper hyperparameters or insufficient model complexity to capture the relationships in the data.

2. Cross-Validation Score (Green Line):

- The cross-validation R^2 score starts very low (around -8), meaning the model is performing much worse than predicting the mean of the target variable.
- The cross-validation score improves as more training examples are added, reaching around -2 after about 3500 training examples. Although the performance improves, an R^2 score below 0 still indicates a poor fit – the model is not accurately predicting the target variable and might be underfitting.

Results - AdaBoost



-Training Score (Red Line):

-The R^2 score of the training set stays relatively stable after an initial drop as more examples are introduced. It hovers slightly below 0.6, indicating that the model is capturing a reasonable portion of the variance in the training data but is not perfectly fitting the data.

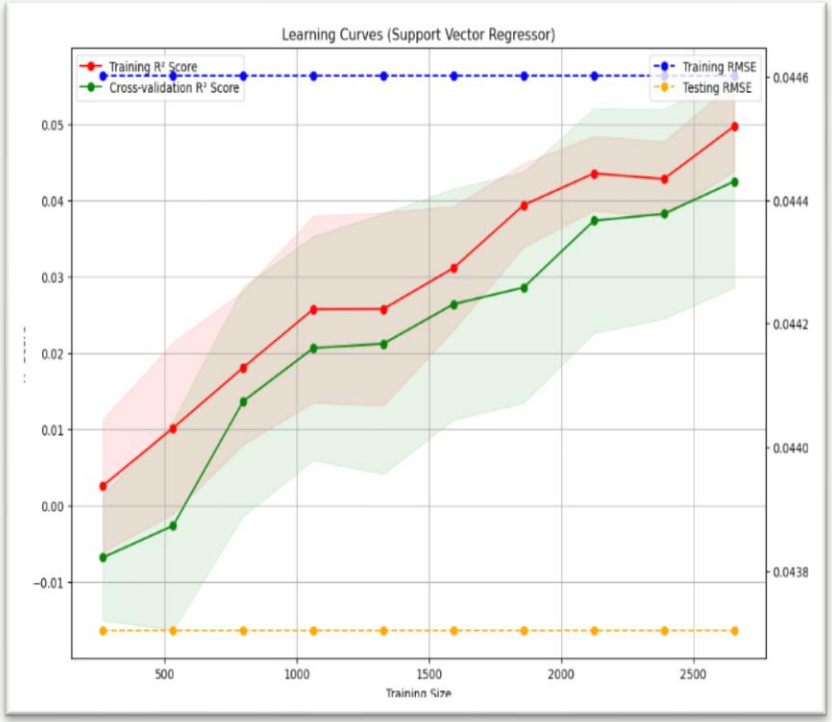
-A high but not perfect score on the training set suggests that the model has not overfitted to the training data, which is a good sign.

-Cross-Validation Score (Green Line):

-The cross-validation score, which reflects the model's performance on unseen data, is lower than the training score, indicating underfitting. It starts at around 0.3 and stabilizes at around 0.25.

-This suggests the model is not generalizing well and may be struggling to capture the complexity of the data.The relatively small gap between the training and cross-validation scores shows that the model is not suffering from high variance but does have high bias.

Results - SVR



Training R^2 Score (Red Line):

-Increases steadily as the training size grows, starting from around 0.01 and rising to just over 0.045 by the maximum training size.

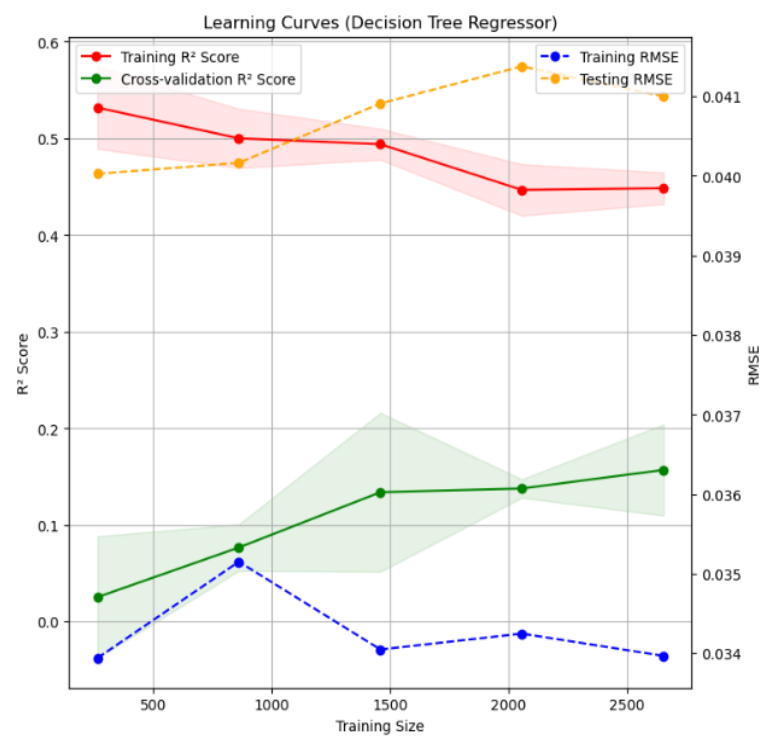
-The steady upward trend suggests that as more data is introduced, the model is able to capture more variance in the training set. However, the R^2 score remains low, which indicates that the model is not capturing a large portion of the variance, even within the training data. This implies the model is not overfitting, but it also suggests the model may struggle with the complexity of the data.

Cross-Validation R^2 Score (Green Line):

-Reflects performance on unseen data, also improves as the training size increases, but remains below the training score throughout. Starting from slightly below 0, it increases to around 0.025 by the end of the curve.

-The cross-validation score being consistently lower than the training score suggests the model is underfitting. This underfitting is further highlighted by the relatively low final R^2 value, implying the model has difficulty generalizing well to new data.

Results - Decision Tree Regressor



Training R^2 Score (Red Line)

-The training R^2 score starts around 0.5 and slightly fluctuates as the training size grows, eventually stabilizing near 0.45-0.5.

-The initial high R^2 value followed by a slight downward trend indicates that the model initially fits the training data well but begins to lose some of its ability to capture the variance as the training size increases. This suggests that the model might be slightly overfitting with smaller data sizes but improves with more data.

Cross-Validation R^2 Score (Green Line):

-The cross-validation R^2 score starts near 0 and gradually increases as the training size grows, reaching a maximum of around 0.15.

-The persistent gap between the training and cross-validation scores suggests that the model is underfitting the data. The increasing trend in the cross-validation score indicates that the model benefits from more data, though it still struggles to generalize effectively.

Conclusion

॥वसुधैव कुटुम्बकम्॥

·The project successfully demonstrated that movie attributes can be effectively used to predict movie popularity using machine learning models.

·Ensemble models, particularly Random Forest for classification and XGBoost for regression, achieved the highest accuracy in predicting popularity.

·The results highlight that machine learning models are valuable tools for forecasting movie success, supporting data-driven decisions in the entertainment industry.

Implications and Future work

Implications: The findings highlight the potential of integrating machine learning into the entertainment industry for predicting movie popularity. This approach can assist filmmakers, production houses, and streaming platforms in decision-making processes such as marketing strategies, budgeting, and content creation. By understanding factors that drive movie popularity, stakeholders can better align their offerings with audience preferences.

Future Work:

- **Expand the Dataset:** Incorporate additional data sources and variables, such as social media trends, critic reviews, and audience sentiment analysis, to enhance the model's predictive power.
- **Explore Advanced Models:** Experiment with deep learning architectures, such as neural networks and recurrent models, to capture complex patterns in the data and improve prediction accuracy.
- **Real-time Prediction Systems:** Investigate the feasibility of deploying the model in real-time applications, such as integrating with streaming platforms or box office tracking tools for dynamic popularity forecasting.
- **Cross-Market Analysis:** Expand the model's applicability by including data from diverse movie markets, such as indie films, international cinema, and OTT releases, to generalize the findings across different audiences.

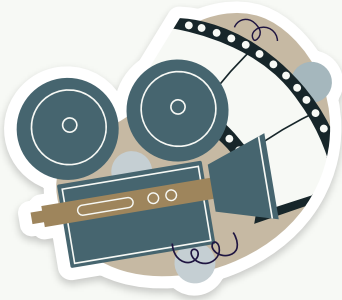
References

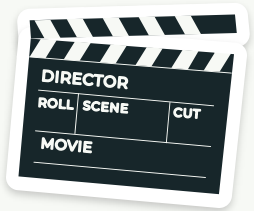
1. Agarwal, M., Venugopal, S., Kashyap, R., & Bharathi, R. (2022). Movie success prediction and performance comparison using various statistical approaches. International Journal of Artificial Intelligence and Applications, 13(1), 1-18. https://doi.org/10.5121/ijaia.2022.13102

2. Oyewola, D. O., & Dada, E. G. (2022). Machine learning methods for predicting the popularity of movies. Journal of Artificial Intelligence and Systems, 4, 65-82. https://doi.org/10.33969/AIS.2022040105

3. Verma, H., & Verma, G. (2019). Prediction model for Bollywood movie success: A comparative analysis of performance of supervised machine learning algorithms. The Review of Socionetwork Strategies, 13, 1-23. https://doi.org/10.1007/s12626-019-00040-6

4. Anand, A. (2023). Predicting the success of a movie using machine learning algorithms: An analysis. International Journal for Multidisciplinary Research (IJFMR), 5(6), 1-8.





॥वसुधैव कुटुम्बकम्॥

Thank You