Data Analysis Using ML Presentation

Predicting the Success of Netflix Shows Using Supervised Learning

Shijin Ramesh
Data Analyst

Business Problem

Objective:

Can we predict whether a newly launched Netflix show will be successful using historical show attributes?

Why it Matters:

- *Helps in early identification of high-potential content.*
- *Optimizes marketing spend.*
- *Improves content acquisition/production decisions.*

Dataset Overview

- **Objective:** *Netflix Titles Dataset (2025 version).*
- **Key Features:** *Title, Genre, Language, Country, Popularity, Vote Count, Rating, etc.*
- **Total Records:** *16,000 shows.*
- **Target Variable:** *success (engineered using rating ≥ 7.5).*

Target Engineering

Success Defined: *Based on rating only (not vote average).*

Why Vote Average Was Excluded:

- *It led to data leakage in model training.*
- *Inflated accuracy without true predictive power.*
- *Business goal is to predict success before the rating becomes available.*

EDA & Key Insights

Language Trends: *Chinese & Japanese shows dominated the dataset.*

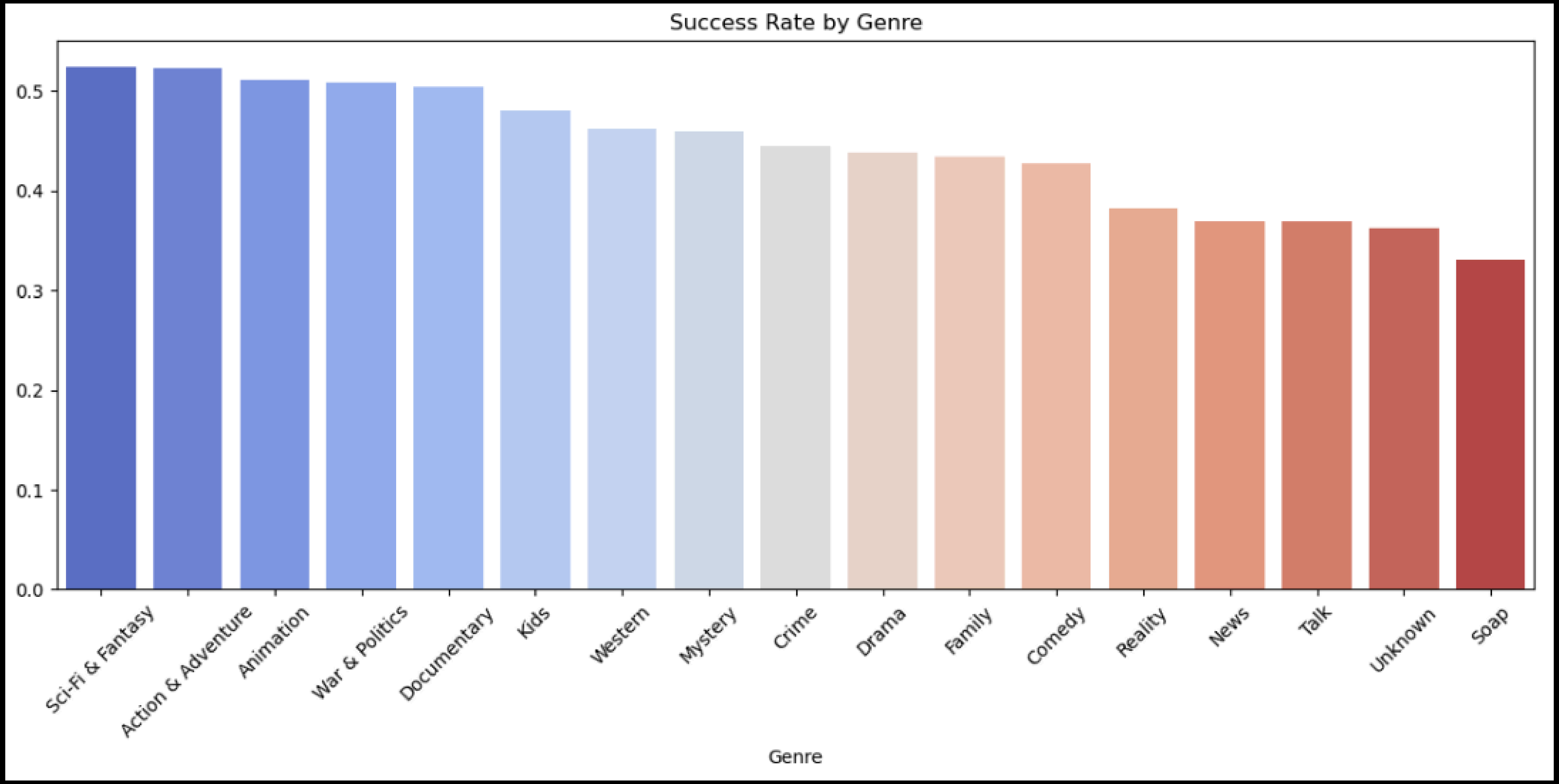
Genres: *Sci-Fi & Fantasy and Action & Adventure shows had higher success rates.*

Vote Average: *Strong correlation with success.*

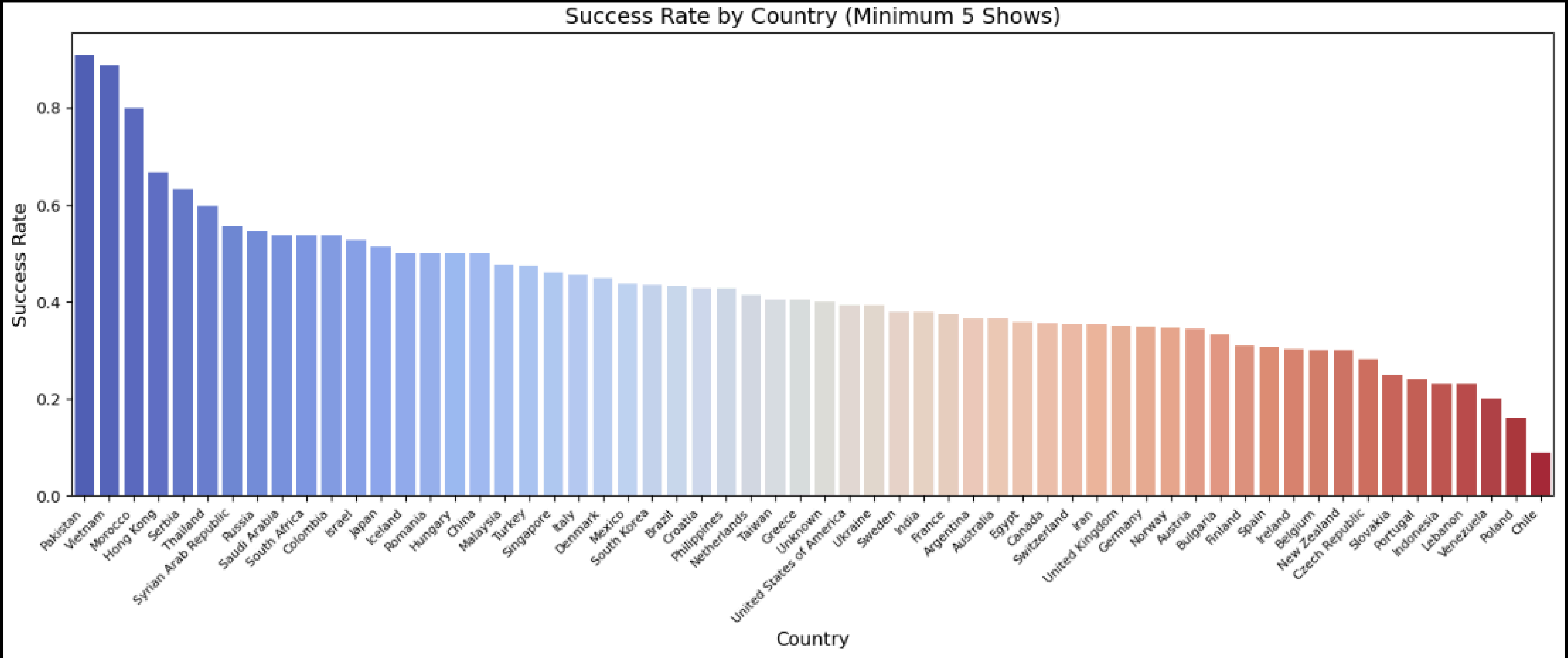
Country: *Pakistan & Vietnam had high success rate with minimum five shows.*

Outliers:

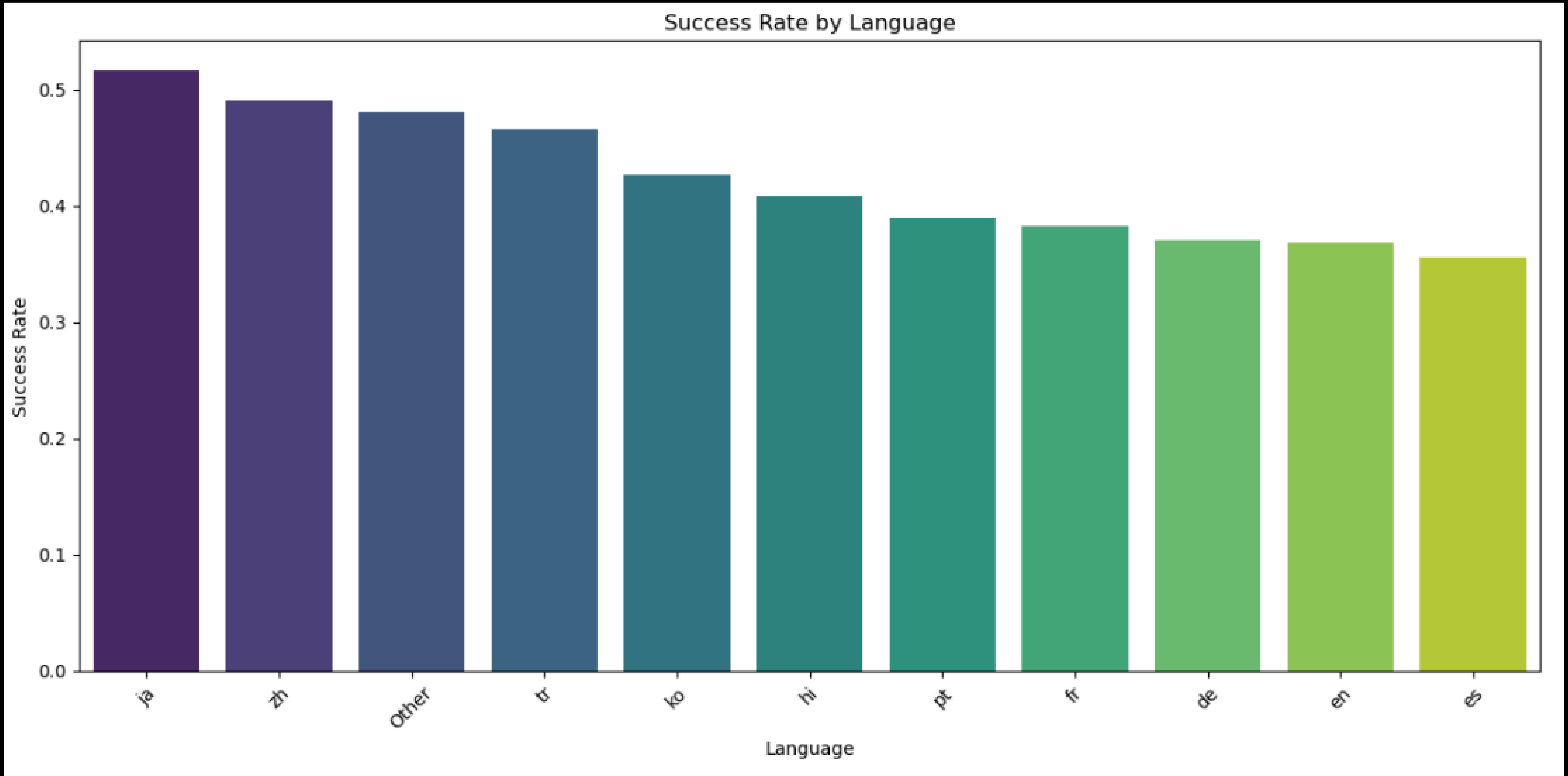
- *1,269 outliers in popularity*
- *1,872 outliers in vote_count*
- *Not removed, as Random Forest is robust to outliers.*



EDA & Key Insights



EDA & Key Insights



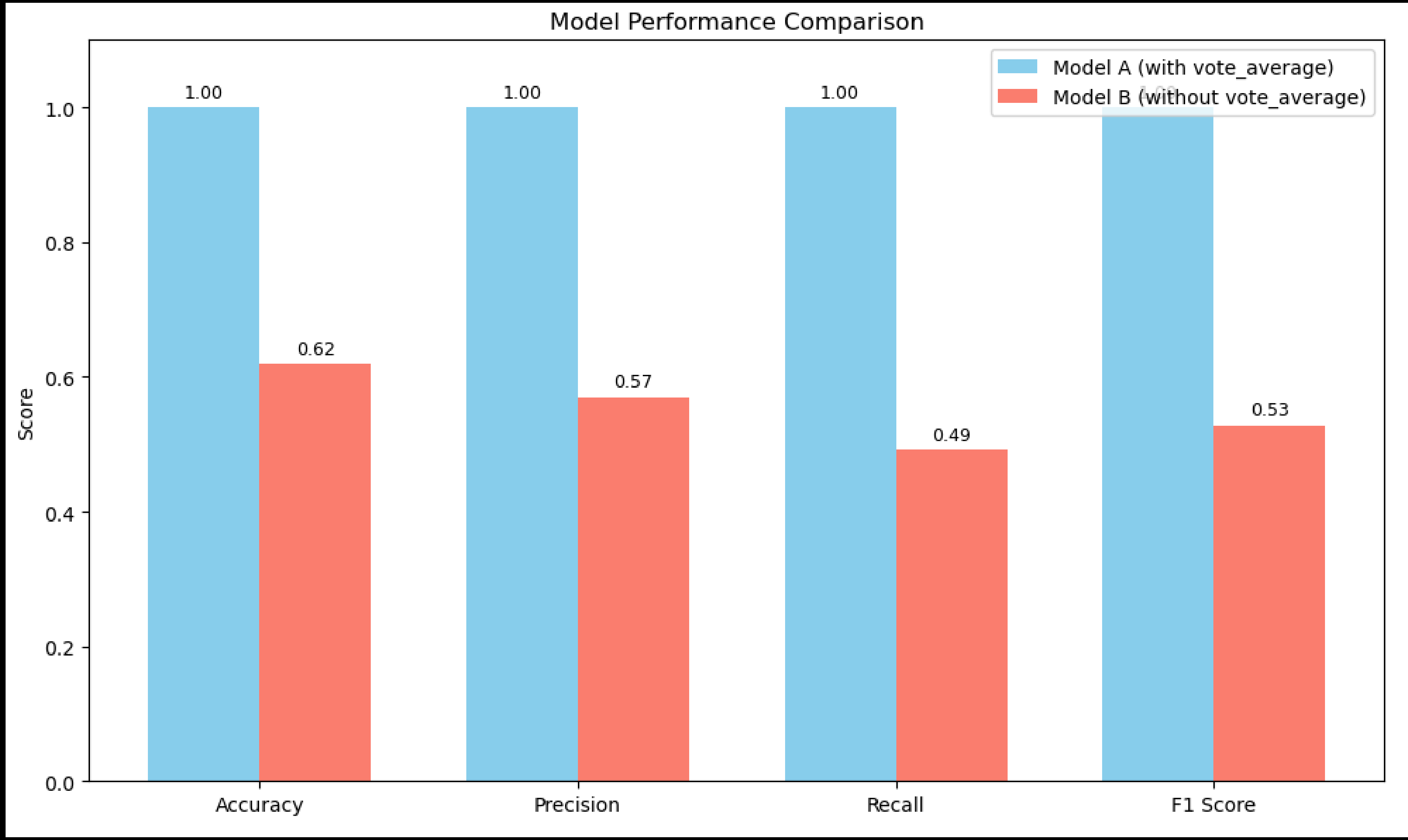
Feature Engineering

- **One-hot encoding on categorical columns:** *Genre, Language, Country.*
- *Removed high-cardinality features with low information gain.*
- *Engineered relevant binary flags for key genres.*

Modeling Strategy

Initial Comparison:

- *Model A (with vote_average) had perfect metrics due to data leakage.*
- *Model B (without vote_average) gave realistic results.*



Model Training

Algorithm Used: *Random Forest Classifier*

Why Random Forest:

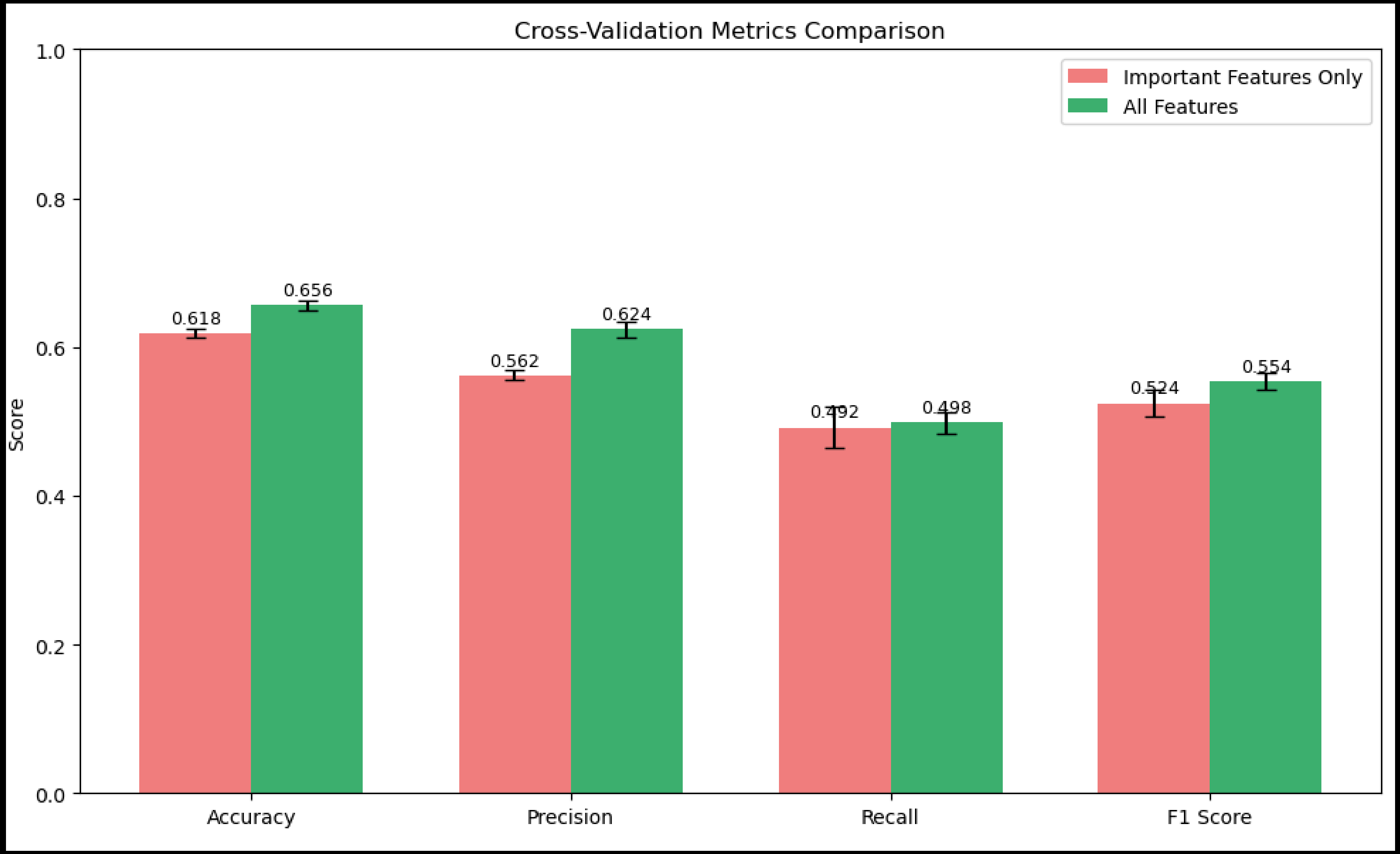
- *Handles mixed data types well*
- *Model B (without vote_average) gave realistic results.*
- *Robust to outliers and non-linearity*
- *Good baseline for feature importance*

Cross-Validation & Tuning

GridSearchCV + RandomizedSearchCV used

Compared two versions:

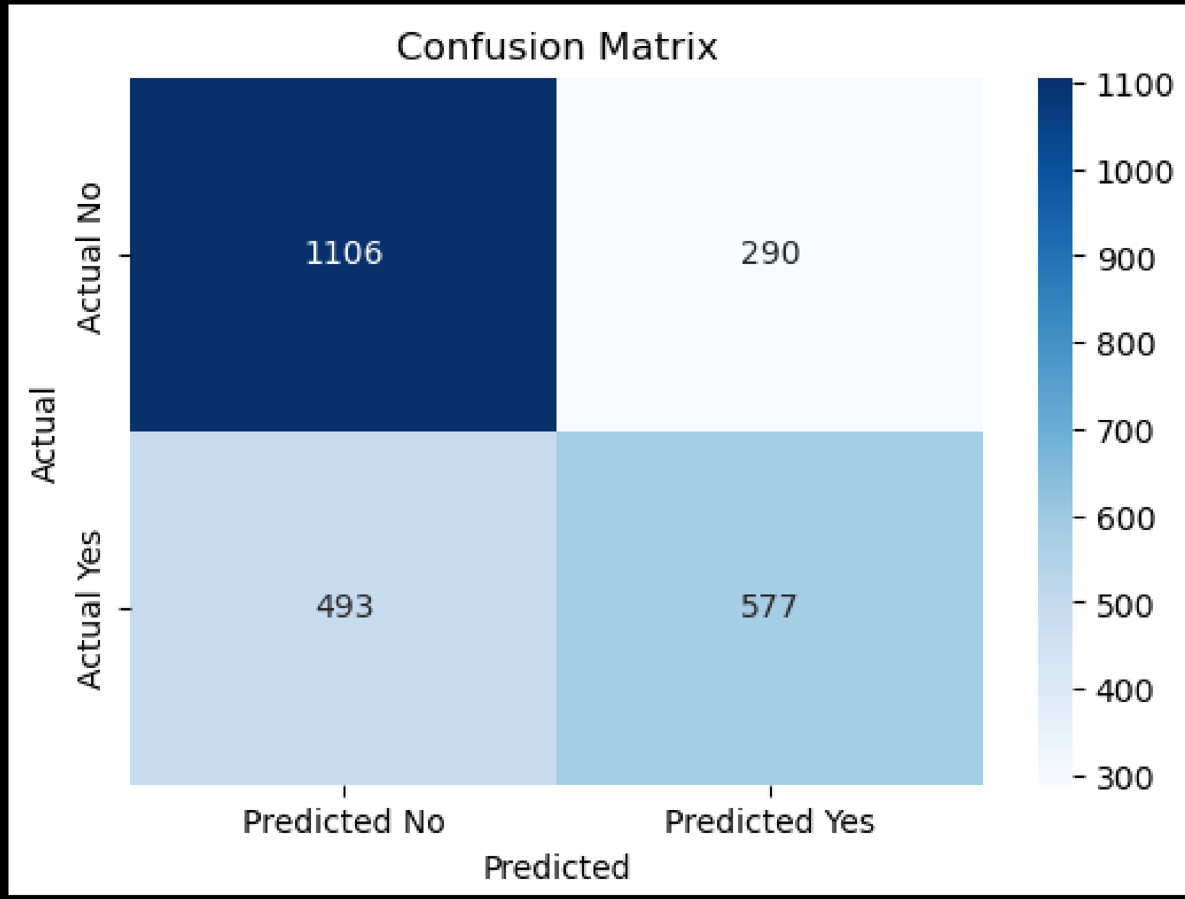
- *Using top important features (based on feature importance)*
- *Using all features*
- *Result: All features gave better CV scores overall.*

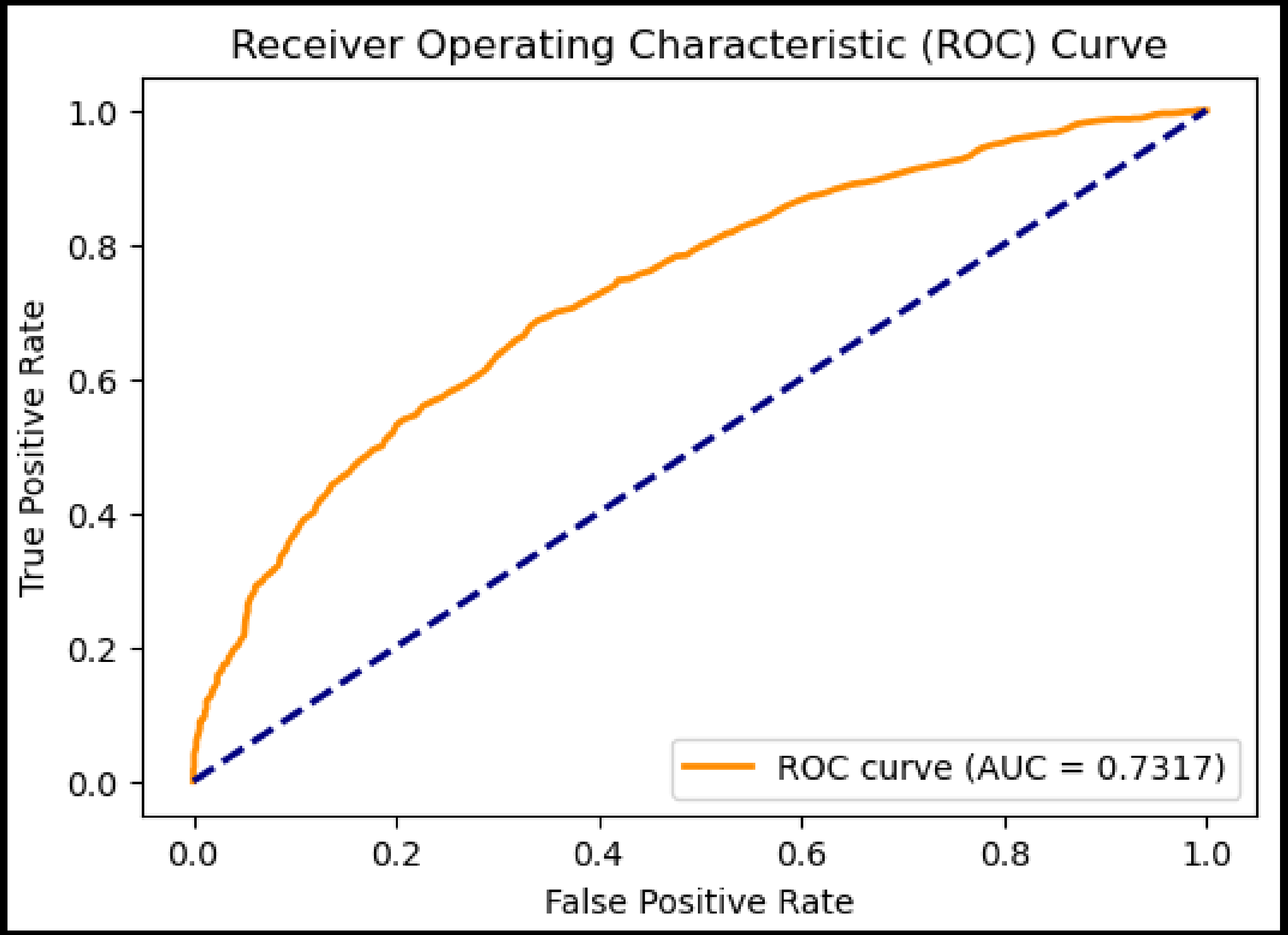


Final Model Evaluation

Test Set Results:

- *Accuracy: 0.6825*
- *Precision: 0.6655*
- *Recall: 0.5393*
- *F1 Score: 0.5958*
- *AUC Score: 0.7317*

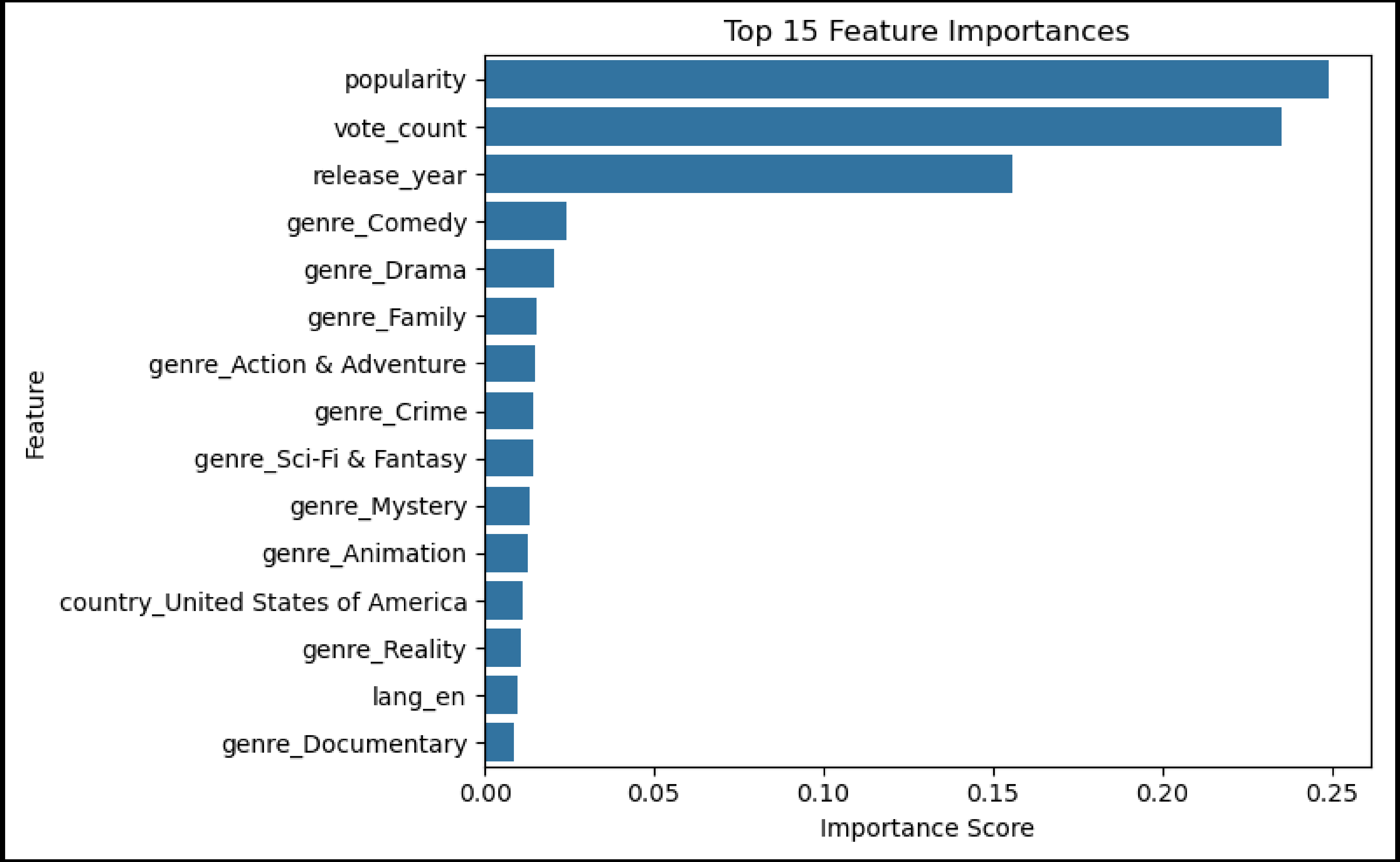




Feature Importance

Top Predictive Features:

- *Popularity*
- *Vote Count*
- *Release Year*
- *genre_Comedy*
- *genre_Drama*



Recommendations to Stakeholders

- Use the model in pre-production to identify shows with high success potential.
- Focus marketing spend on shows predicted to succeed but with low popularity.
- Explore feature interactions (e.g., language + genre) for future commissioning.
- Retrain regularly with recent data to capture changing trends.

Limitations & Next Steps

- *Does not include text data like synopsis or cast yet.*
- *Cultural influence, production budget not factored in.*
- *Plan to extend with NLP and ensemble learning techniques.*

Appendix

- *Code Link: https://github.com/shijin/NetflixSuccessfulShowPredictiveModeling*