Final Project Report: Milestone 3

Movie Success Prediction: Analysis and Interactive Dashboard

Author: Pranil Ingle

Course: CAP5771 SP25

Date: April 23, 2025

Table of Contents

1. Project Summary

This project aimed to develop an interactive data science tool to analyze movie datasets and understand the key factors influencing critical movie success, specifically the Rotten Tomatoes tomatometer_status ('Rotten', 'Fresh', 'Certified-Fresh'). The project utilized datasets from Rotten Tomatoes (Movies and Reviews), IMDb, Netflix, and TMDB, sourced primarily from Kaggle. The technology stack included Python, Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, and Streamlit for the final dashboard. The project followed three milestones: Data Collection/Preprocessing/EDA, Feature Engineering/Selection/Modeling, and Evaluation/Interpretation/Tool Development. The ultimate goal was to create a tool allowing users to explore movie data and interact with a predictive model.

2. Milestone 1 Recap: Data Preprocessing & EDA

Milestone 1 focused on preparing the data for analysis. Key steps included:

- **Data Loading & Initial Exploration:** Datasets were loaded, and initial properties (info, head, missing values) were examined.
- **Handling Missing Data:** Strategies such as imputation (using median for numerical, mode for categorical) or dropping rows/columns were applied based on the extent of missingness.
- **Duplicate Removal:** Duplicate records were identified and removed.
- **Outlier Treatment:** Statistical methods (Z-score, IQR) were used to manage extreme outliers in numerical columns like review counts and runtime.
- **Exploratory Data Analysis (EDA):** Descriptive statistics were calculated. Visualizations like histograms, box plots, correlation matrices, and pairplots were generated to understand distributions, relationships, and identify potential issues like skewness or high correlation. Key findings included the right-skewed nature of review counts and the moderate positive correlation between critic and audience ratings.

(Refer to Milestone 1 Report for full details)

3. Milestone 2 Recap: Feature Engineering, Selection, & Modeling

Building upon the cleaned data, Milestone 2 focused on preparing features and building predictive models:

- **Feature Engineering:** New features were created to potentially improve model performance. These included time-based features (release_year, movie_age, etc.), interaction/ratio features (runtime_rating_interaction, audience_tomatometer_ratio), and processing of text/categorical data (num_genres, production_company_encoded). The target variable tomatometer_status was label encoded.
- **Feature Selection:** Random Forest Feature Importance was used to identify the most predictive features for tomatometer_status. The top 10 selected features were: tomatometer_rating, runtime_rating_interaction, tomatometer_fresh_critics_count, tomatometer_count, tomatometer_rotten_critics_count, audience_rating, tomatometer_top_critics_count, runtime, movie_age, audience_count.
- **Modeling:** The task was defined as multiclass classification. Data was split (60/20/20 Train/Validation/Test), and features were scaled using StandardScaler. Three models were trained: Logistic Regression, Decision Tree, and Random

Forest. The Random Forest model showed superior performance on the validation set.

(Refer to Milestone 2 Report for full details)

4. Milestone 3: Evaluation and Interpretation

This milestone involved finalizing the model evaluation and interpreting the results.

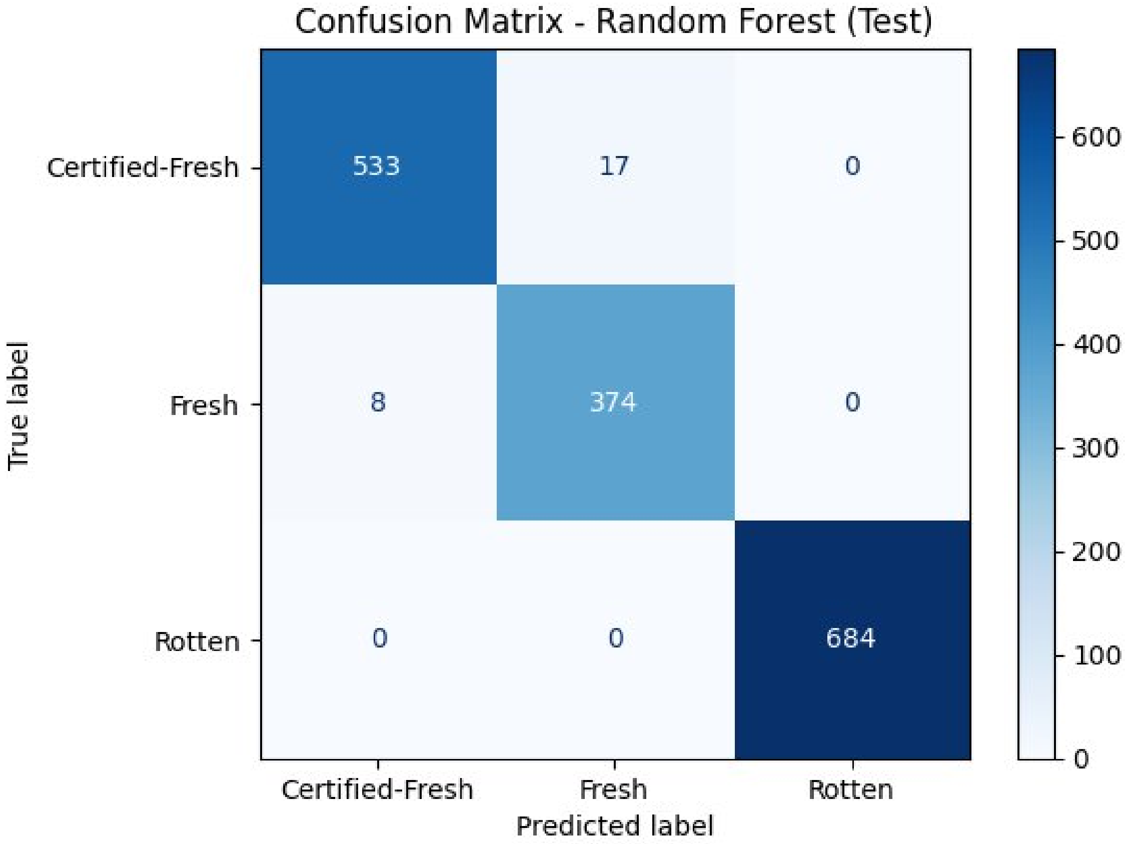
4.1. Final Model Evaluation (Test Set)

The best model from Milestone 2 (Random Forest Classifier) was evaluated on the held-out test set. The performance was excellent and consistent with validation results:

- **Accuracy:** 0.9845
- **Precision (Weighted):** 0.9847
- **Recall (Weighted):** 0.9845
- **F1-Score (Weighted):** 0.9846
- **ROC AUC (Weighted OvR):** 0.9987

The detailed classification report confirmed high performance across all classes:

	Precision	Recall	F1-Score	Support
Certified-Fresh	0.99	0.97	0.98	550
Fresh	0.96	0.98	0.97	382
Rotten	1.00	1.00	1.00	684
Accuracy			0.98	1616
Macro-Avg	0.98	0.98	0.98	1616
Weighted-Avg	0.98	0.98	0.98	1616



The confusion matrix visually confirmed the model's ability to correctly classify the vast majority of test cases:

4.2. Model Interpretation

The Random Forest model achieved high accuracy primarily because the tomatometer_rating feature, which largely defines the tomatometer_status, was included as input. The model effectively learned the thresholds (<60% for Rotten, >=60% for Fresh).

However, the model also learned the nuances required to distinguish 'Fresh' from 'Certified-Fresh'. Feature importance showed that critic counts (tomatometer_fresh_critics_count, tomatometer_count, tomatometer_rotten_critics_count, tomatometer_top_critics_count) were also important, indicating the model used this information, alongside the rating, to identify movies meeting the stricter criteria for 'Certified-Fresh' status (e.g., minimum number of total and top critic reviews). The engineered feature runtime_rating_interaction was also highly important, suggesting the combined effect of runtime and rating influenced the prediction.

4.3. Actionable Insights

- The model confirms the strong, definitional link between the numerical critic rating percentage and the final status label ('Rotten', 'Fresh', 'Certified-Fresh').
- Achieving 'Certified-Fresh' status is dependent not only on a high rating (>=75%) but significantly on the *volume* and *type* (Top Critic) of reviews received, as reflected by the importance of count features.
- Audience metrics (audience_rating, audience_count) had lower importance for predicting the *critic-defined* status compared to critic-based metrics.

4.4. Limitations and Potential Bias

- **Data Bias:** The model is trained on data from Rotten Tomatoes. Its predictions reflect the patterns within that specific ecosystem. The critics included, the movies reviewed, and the rating system itself may contain inherent biases that the model learns. The model's applicability might be limited outside this context.
- **Feature Scope:** The model relies heavily on rating and review count features. Other factors potentially influencing success (e.g., budget, marketing spend, specific cast/director combinations beyond encoding, script quality) were not included in this feature set.

5. Milestone 3: Tool Development (Interactive Dashboard)

An interactive dashboard was developed as the primary tool deliverable for this project.

5.1. Tool Choice and Objective

Streamlit was chosen as the dashboarding library due to its ease of use and rapid development capabilities for data-centric applications. The objective of the dashboard is to provide an interactive interface for exploring the movie data, visualizing feature importance, and allowing users to get predictions from the trained Random Forest model.

5.2. Dashboard Features

The Streamlit application (dashboard.py) includes the following sections:

- **Title and Introduction:** Welcomes the user and explains the dashboard's purpose.
- **Data Exploration Tab:** Displays a sample of the cleaned rotten_tomatoes_movies.csv data using st.dataframe. (Further filtering could be added).
- **Feature Importance Tab:** Lists the top 10 features used by the Random Forest model, as determined in Milestone 2.

- **Model Performance Tab:** Summarizes the final test set performance metrics (Accuracy, ROC AUC, F1-Score) and displays the saved test set confusion matrix image.
- **Interactive Prediction Tool (Sidebar):** Allows users to input values for the 10 required features using sliders and number inputs. An interaction term (runtime_rating_interaction) is calculated automatically. A "Predict Status" button triggers the model.
- **Prediction Output:** Displays the predicted tomatometer_status ('Rotten', 'Fresh', or 'Certified-Fresh') and the model's confidence probabilities for each class.

5.3. Technology Used

Python, Streamlit, Pandas, Joblib (for model/scaler persistence), Scikit-learn, Matplotlib/Seaborn (for plot generation saved prior).

5.4. Running the Dashboard

1. Ensure Python and required libraries (streamlit, pandas, scikit-learn, joblib) are installed.
2. Place dashboard.py, random_forest_model.joblib, scaler.joblib, cleaned_rotten_tomatoes.csv, and confusion_matrix_test_set.png in the same directory.
3. Open a terminal or command prompt, navigate to that directory.
4. Run the command: streamlit run dashboard.py
5. The dashboard will open in the default web browser.

6. Conclusion

This project successfully navigated the data science workflow from data collection and cleaning through to model development and interactive tool creation. A highly accurate Random Forest model (Test ROC AUC: 0.9987) was developed to predict the Rotten Tomatoes status category, leveraging engineered features and critic review metrics. While the model's accuracy is high partly due to the nature of the prediction task (status from rating), it effectively learned the nuances differentiating the status categories, particularly 'Certified-Fresh'. The developed Streamlit dashboard provides an accessible way to explore the data, understand feature importance, and interact with the predictive model, fulfilling the project's core objective.

7. Repository Information

- **GitHub Repository:** https://github.com/CodeRanger1998/cap5771sp25-project

- **Collaborators Invited:** TA Jimmy Rao [@JimmyRaoUF], Grader Daniyal Abbasi [@abbasidaniyal], Dr. Cruz [@lcruzcas], Dr. Grant [@cegme].
- **Submission:** Repository linked via Gradescope.
- **Demo Video:** A demonstration video showcasing the Streamlit tool is included in Videos folder