Capstone Project 2: Movie Voting Average Prediction

Objective

The goal of this project is to predict the vote average of movies using metadata available prior to release. By building a machine learning model on features like genres, cast, crew, popularity, and others, studios can gain early insight into a movie's reception, aiding in marketing, investment, and distribution strategies.

Dataset

The dataset was sourced from The Movie Database (TMDB) and included approximately 10,000 movies. Key features included title, release date, genres, actors, directors, popularity, vote count, and vote average (target).

Data Wrangling

Initial steps involved dropping irrelevant columns such as 'Unnamed: 0', 'poster_path', and 'backdrop_path'. The 'release_date' column was converted to datetime format to extract 'release_year'. Several columns like 'genres', 'actors', 'keywords', and 'director' were stringified lists, so they were converted to actual Python lists using `ast.literal_eval`.

Exploratory Data Analysis (EDA)

EDA revealed the distribution of numerical features and correlations between variables. While features like 'runtime', 'popularity', and 'vote_count' were positively skewed, log transformation helped in normalizing them. Correlations between vote average and individual features were weak, suggesting nonlinear patterns.

Top genres included Drama, Action, and Comedy, while Nicolas Cage, Jackie Chan, and Tom Hanks emerged as most frequent actors.

Preprocessing

Preprocessing included:
- Log transformation of skewed variables: vote_count, popularity, and runtime.
- Trimming outliers using a standard deviation-based filter.
- One-hot encoding of categorical features.
- Handling of missing values using SimpleImputer.
- Feature engineering: extracted primary genre and actor, and counted total genres/actors.

Modeling

Multiple models were trained and evaluated:
- Linear Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- XGBoost (with and without hyperparameter tuning)

The final model used RandomizedSearchCV to tune XGBoost parameters.

Results

The tuned XGBoost model achieved the best performance:
- R^2 Score: 0.46
- Mean Absolute Error (MAE): 0.46
- Root Mean Squared Error (RMSE): 0.65

This indicates the model is able to explain 46% of the variance in vote averages using the input features.

Conclusion

Despite limited correlation between individual features and the target, tree-based models captured non-linear relationships effectively. The inclusion of feature engineering and preprocessing strategies notably improved model performance.

Future work could explore incorporating additional metadata such as budget, revenue, or user reviews to further enhance model accuracy.