Movie Rating Prediction

What movie features can influence the audience’s preference?

Project Objectives and Data Source

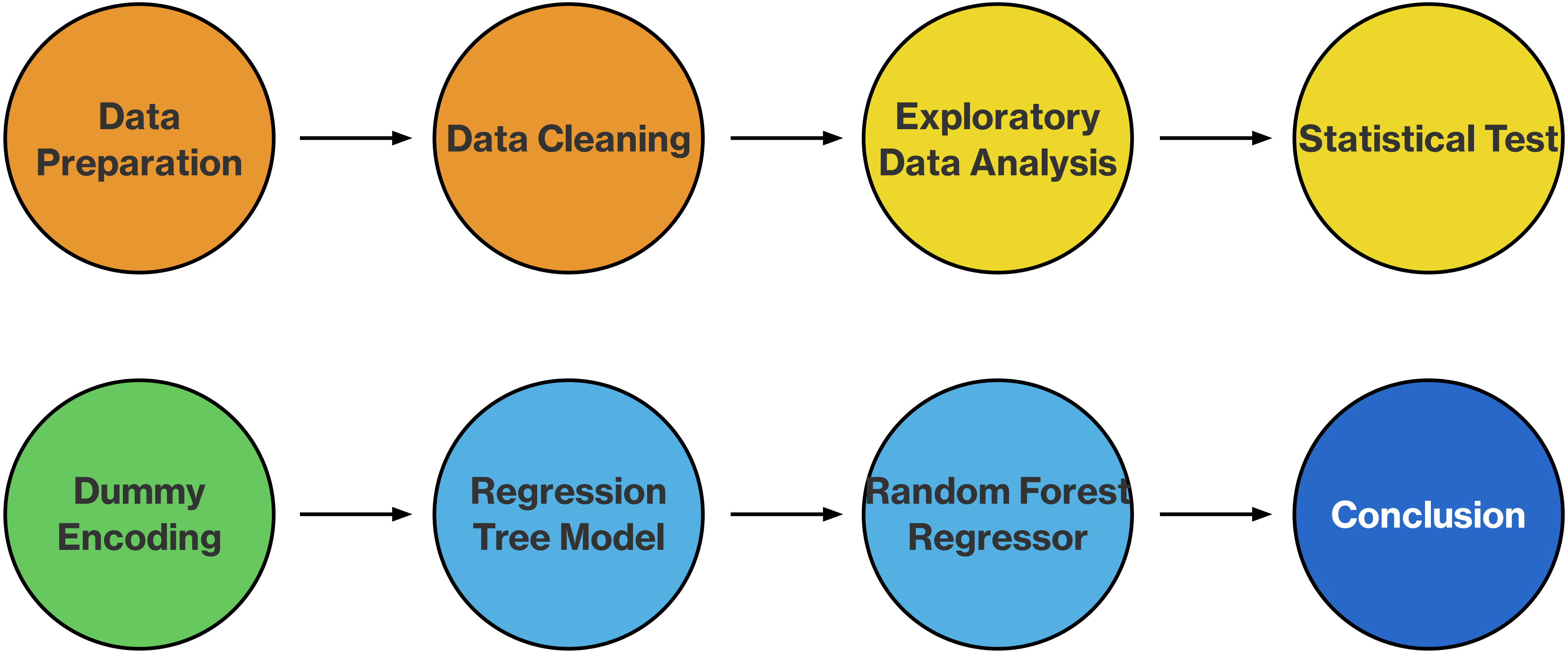
Objectives:

The project aims to examine features that affect people’s movie preference, which is, how people score a movie, and build regression models to predict the rating score for a movie with important features.

Data Source:

The datasets are from TMDB and GroupLens. It covers 45,000 movies, released before or on 2017, and contain 26 million ratings from 270,000 users. It comes with two parts: one about movie features and another about user ratings. Target variable in this project is the TMDB rating scores which have been weighted.

Flow Chart



Data Preparation

Data Cleaning

Exploratory Data Analysis

Statistical Test

Dummy Encoding

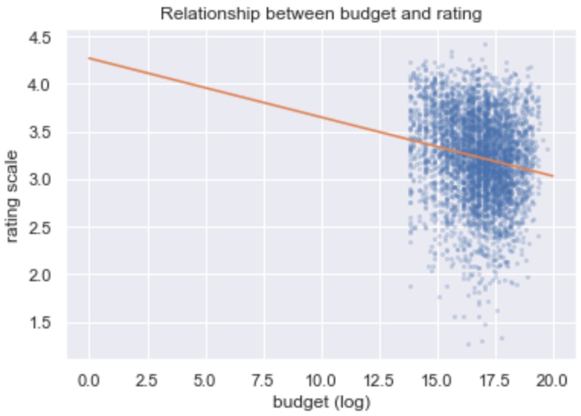
Regression Tree Model

Random Forest Regressor

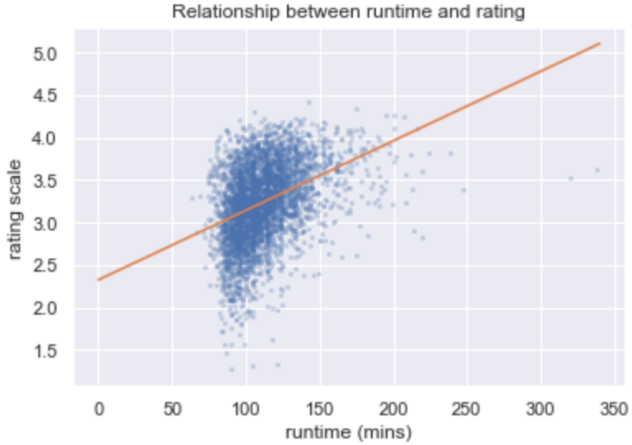
Conclusion

Features Examination with EDA

- The scatter plots illustrating the relationship between feature and target variables shows the influence of budget and runtime.

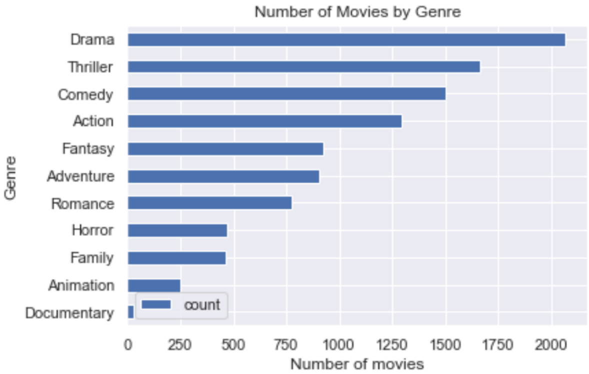


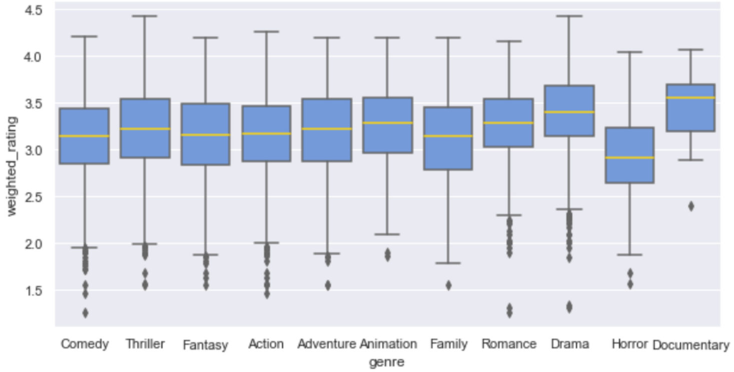
```
correlation:-0.167, p-value0.000
```



```
correlation:0.379, p-value0.000
```

- Drama as the most popular film types in terms of high production numbers also have higher average rating score than others.

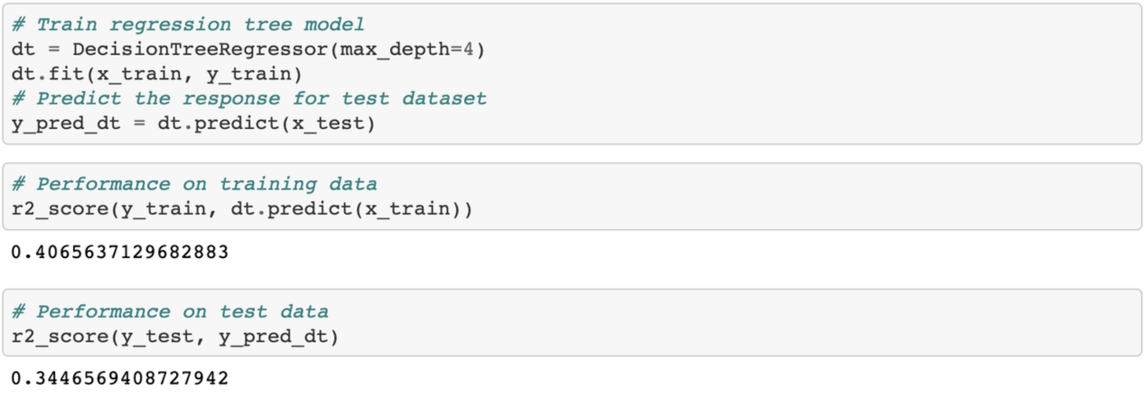




Model Performance

Decision Tree Regressor

The baseline model is regression tree. Since its parameters have been optimized, there are “overfitting” problem observed. Its prediction score is 0.34.



```
# Train regression tree model
dt = DecisionTreeRegressor(max_depth=4)
dt.fit(x_train, y_train)
# Predict the response for test dataset
y_pred_dt = dt.predict(x_test)
```

```
# Performance on training data
r2_score(y_train, dt.predict(x_train))
```

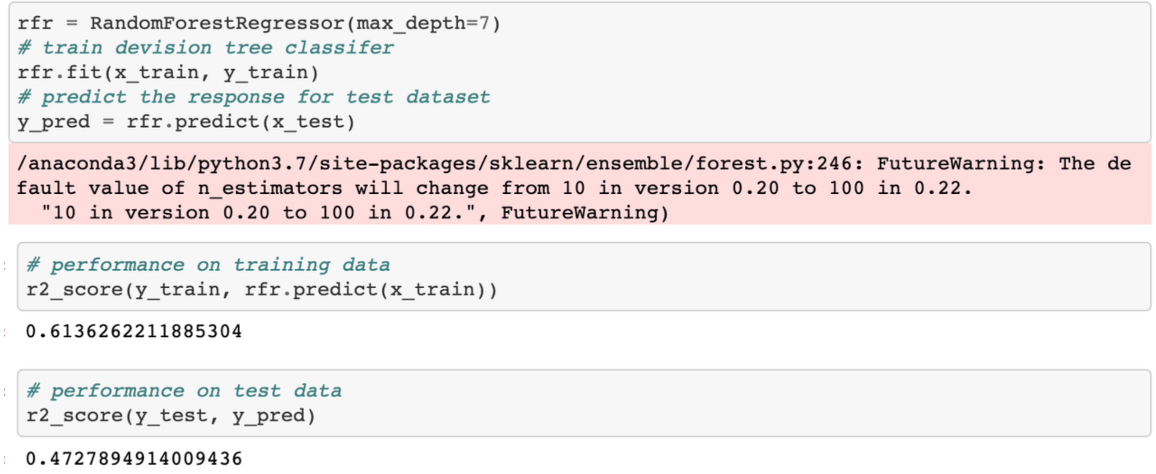
```
0.4065637129682883
```

```
# Performance on test data
r2_score(y_test, y_pred_dt)
```

```
0.3446569408727942
```

Random Forest Regressor

Random forest is good at avoiding overfitting and also reducing variance. As a result, the prediction score improves to 0.47.



```
rfr = RandomForestRegressor(max_depth=7)
# train devision tree classifer
rfr.fit(x_train, y_train)
# predict the response for test dataset
y_pred = rfr.predict(x_test)
```

```
/anaconda3/lib/python3.7/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The de
fault value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
# performance on training data
r2_score(y_train, rfr.predict(x_train))
```

```
0.6136262211885304
```

```
# performance on test data
r2_score(y_test, y_pred)
```

```
0.4727894914009436
```

Conclusions and Limitations

Conclusion

- Decision tree regressor and random forest regressor are used to predict the weighted values of movie ratings. Since the target variable is a continuous values with infinite possible outcomes, two models' performance evaluated with R-squared metric are not high as expected.
- For both models, budget, runtime, how many votes a move has, and also whether it belongs to drama category show are important features for the model. Among them, how many votes a move has is most crucial. It's reasonable to think that popular movies tend to attract more people to watch.

Limitations: What can be improved

- There are too many categories within a feature, causing some categories only have little data points. It would be better if we can aggregate some features based on their similar relationship with the target variables, in order to simplify the category number.
- Besides using the current columns which have come with the dataset as the features, it would better to extract others, especially the dataset contain text data. For this project, I only keyword-matched solutions to extract new features form the text data, however, their accuracy should be double verified with other sources to be more complete.