IMDB Box Office Prediction

Student: Kengtian Lu
ID: a1840363

Introduction
Methodology
Implementation
Results and Findings

Introduction

Movie Revenue Prediction
Dataset: from Kaggle website
Goal: find the best model

Regression problem

Methodology

Exploratory data analysis

Data pre-processing

Feature Engineering

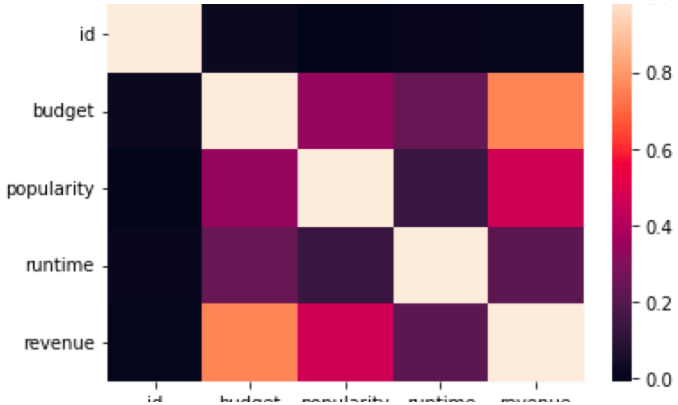
Research models

- Random Forest Regression
- XGBoost Regression
- Ridge Regression
- Lasso Regression

```
train.describe()
```

started 02:44:42 2022-10-17, finished in 22ms

	id	budget	popularity	runtime	revenue
count	3000.000000	3.000000e+03	3000.000000	2998.000000	3.000000e+03
mean	1500.500000	2.253133e+07	8.463274	107.856571	6.672585e+07
std	866.169729	3.702609e+07	12.104000	22.086434	1.375323e+08
min	1.000000	0.000000e+00	0.000001	0.000000	1.000000e+00
25%	750.750000	0.000000e+00	4.018053	94.000000	2.379808e+06
50%	1500.500000	8.000000e+06	7.374861	104.000000	1.680707e+07
75%	2250.250000	2.900000e+07	10.890983	118.000000	6.891920e+07
max	3000.000000	3.800000e+08	294.337037	338.000000	1.519558e+09



Implementation:

Pre-process:

Null and Missing Values

- Remove the features which have too much empty value

‘homepage’ ‘tagline’ ‘keywords’

- Replace null data with mean

‘runtime’ ‘budget’

```
Missing value in train set:
belongs_to_collection    2396
homepage                 2054
tagline                   597
Keywords                  276
production_companies      156
production_countries       55
spoken_languages           20
crew                       16
cast                       13
overview                    8
genres                      7
runtime                     2
poster_path                 1
title                       0
status                      0
id                          0
release_date                0
popularity                  0
original_title              0
original_language           0
imdb_id                     0
budget                      0
revenue                     0
dtype: int64
```

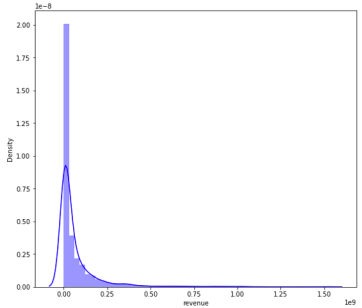
Implementation:

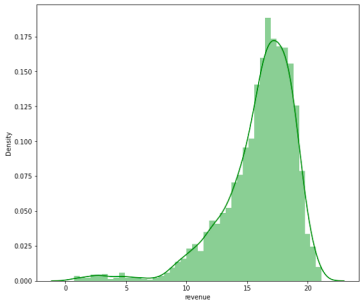
Target Variable

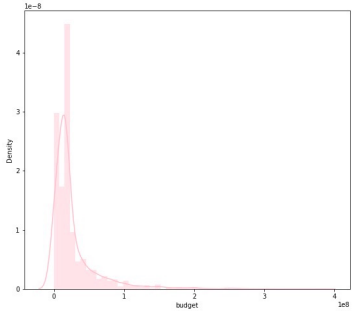
- Logarithm Transformation

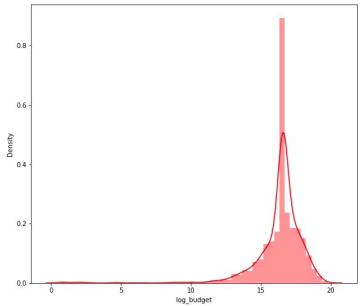
‘revenue’ ‘budget’

Json → Numerical
Nominal







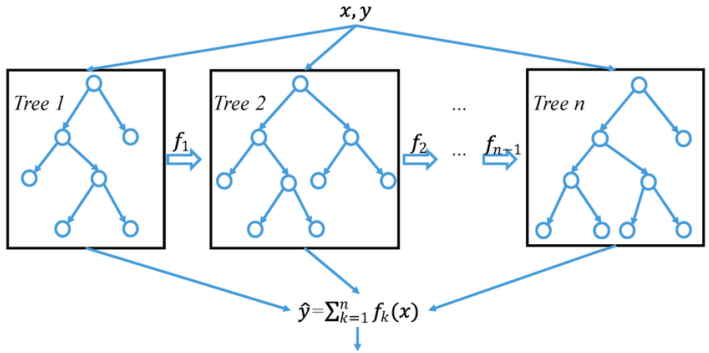


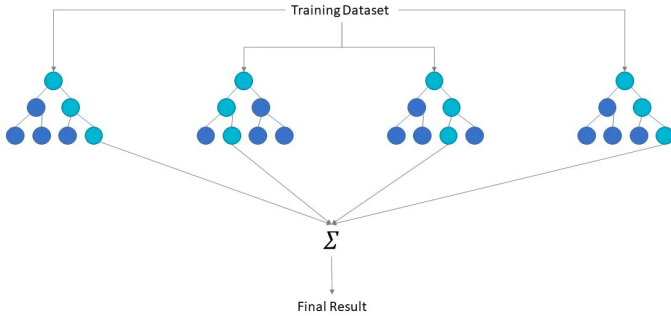
Implementation:

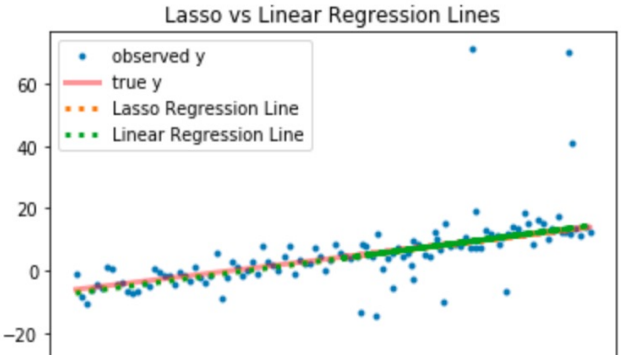
Training data and Testing data:

- Randomly pick 20% data from train as testing data use left 80% data as training data

Random Forest Regression
XGBoost Regression
Ridge Regression
Lasso Regression







Results and Findings

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

```
# Defining the Random Forest Model
rf_model = RandomForestRegressor(random_state=39)
# Fitting the model
rf_model.fit(X_train_full, y_train)
# Prediction
y_pred_rf = rf_model.predict(X_valid_full)

# Calculate RMSE
rf_rmse= np.sqrt(mean_squared_error(y_valid, y_pred_rf))
rf_rmse
```

executed in 1.03s, finished 02:44:55 2022-10-17

```
2.0910228704589255
```

```
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
ls_model = Lasso()
ls_model.fit(X_train_full, y_train)

#prediction
y_pred_ls = ls_model.predict(X_valid_full)

ls_rmse= np.sqrt(mean_squared_error(y_valid, y_pred_ls))
ls_rmse
```

executed in 9ms, finished 02:44:56 2022-10-17

```
2.535626548290038
```

```
# Define the XGBoostmodel
xgb_model = XGBRegressor()

# Fit the model
xgb_model.fit(X_train_full, y_train)
# Prediction
y_pred_xgb = xgb_model.predict(X_valid_full)

xgb_rmse= np.sqrt(mean_squared_error(y_valid, y_pred_xgb))
xgb_rmse
```

executed in 186ms, finished 02:44:56 2022-10-17

```
2.2268147279365027
```

```
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
rg_model = Ridge()
rg_model.fit(X_train_full, y_train)
#prediction
y_pred_rg = rg_model.predict(X_valid_full)

rg_rmse= np.sqrt(mean_squared_error(y_valid, y_pred_rg))
rg_rmse
```

executed in 9ms, finished 02:44:56 2022-10-17

```
2.52395018348213
```

Random Forest Regression is the best

Reference:

Movie posters in page 1&2:

https://www.imdb.com/title/tt0848228/?ref_=fn_al_tt_0

https://www.imdb.com/title/tt5273488/?ref_=fn_al_tt_1

https://www.imdb.com/title/tt2395427/?ref_=fn_al_tt_9

https://www.imdb.com/title/tt1399103/?ref_=tt_mv_close

https://www.imdb.com/title/tt2771200/?ref_=fn_al_tt_0

Regression pictures in page 7:

https://www.ibm.com/cloud/learn/random-forest

https://www.researchgate.net/figure/A-general-architecture-of-XGBoost_fig3_335483097

https://machinelearningjourney.com/index.php/2020/02/13/ridge-regression/

RMSE formula in page 8:

https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e