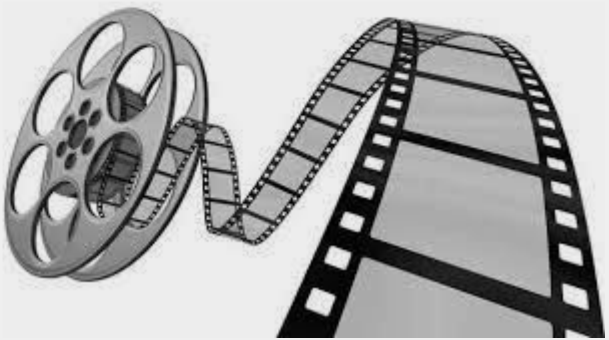
Capstone Project 1: Film revenue prediction



Kunture Junuspayeva

Problem

In a world where movies made an estimated $41.7 billion in 2018, the film industry is more popular than ever.

- But what movies make the most money at the box office?
- How much does a director matter? Or the budget?
- Can we build models, which will be able to accurately predict film revenue?

Goal and Data

- **Goal**
  Using Machine Learning models to predict a film revenue.

- **Data**
  Data comes from the public dataset uploaded to Kaggle.com

Data Wrangling

- **The train dataset consists of 3000 rows or films and 23 columns.**

- **The target variable is “revenue”.**

- **This dataset contains lists with dictionaries(JSON style). Some lists contain a single dictionary, some have several. We extract data from these columns and create dummy variables.**

```
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 23 columns):
id                       3000 non-null int64
belongs_to_collection    604 non-null object
budget                   3000 non-null int64
genres                   2993 non-null object
homepage                 946 non-null object
imdb_id                  3000 non-null object
original_language        3000 non-null object
original_title           3000 non-null object
overview                 2992 non-null object
popularity               3000 non-null float64
poster_path              2999 non-null object
production_companies     2844 non-null object
production_countries     2945 non-null object
release_date             3000 non-null object
runtime                  2998 non-null float64
spoken_languages         2980 non-null object
status                   3000 non-null object
tagline                  2403 non-null object
title                    3000 non-null object
Keywords                 2724 non-null object
cast                     2987 non-null object
crew                     2984 non-null object
revenue                  3000 non-null int64
dtypes: float64(2), int64(3), object(18)
```

Data Cleaning

- **“collection_name” and “has_collection” are extracted information from column “belongs_to_collection”**
- **Similar steps applied to other columns**

belongs_to_collection

```
for i, e in enumerate(master['belongs_to_collection'][:5]):
    print(i, e)
```

```
0 [{'id': 313576, 'name': 'Hot Tub Time Machine Collection', 'poster_path': '/iEhb00TGPucF0b4joM1ieyY026U.jpg', 'backdrop_p
ath': '/noeTVcgpBiD48fDjFVic1Vz7ope.jpg'}]
1 [{'id': 107674, 'name': 'The Princess Diaries Collection', 'poster_path': '/wt5AMbxPTS4Kfjx7Fgm149qPfZl.jpg', 'backdrop_p
ath': '/zSEtYD77pKRJlUPx34BJgUG9v1c.jpg'}]
2 nan
3 nan
4 nan
```

Lets create function text_to_dict to convert columns to dictionary.

```
dict_columns = ['belongs_to_collection', 'genres', 'production_companies',
                'production_countries', 'spoken_languages', 'Keywords', 'cast', 'crew']
#access the dictionaries
def text_to_dict(df):
    for column in dict_columns:
        df[column] = df[column].apply(lambda x: {} if pd.isna(x) else ast.literal_eval(x) )
    return df

dfx = text_to_dict(master)
for col in dict_columns:
       master[col]=dfx[col]
```

```
master['belongs_to_collection'].apply(lambda x:len(x) if x!= {} else 0).value_counts()
```

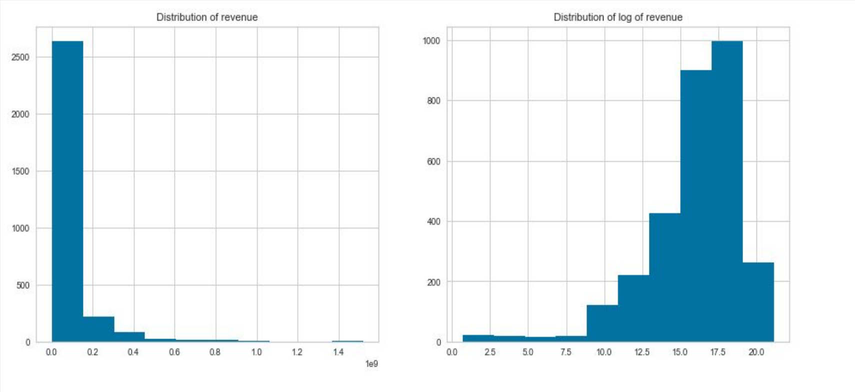
```
0    5917
1    1481
Name: belongs_to_collection, dtype: int64
```

We create two new columns from column "belongs_to collection", first one is collection name and second one has collection or not. We assume that other information from this column we cant use for futher prediction.

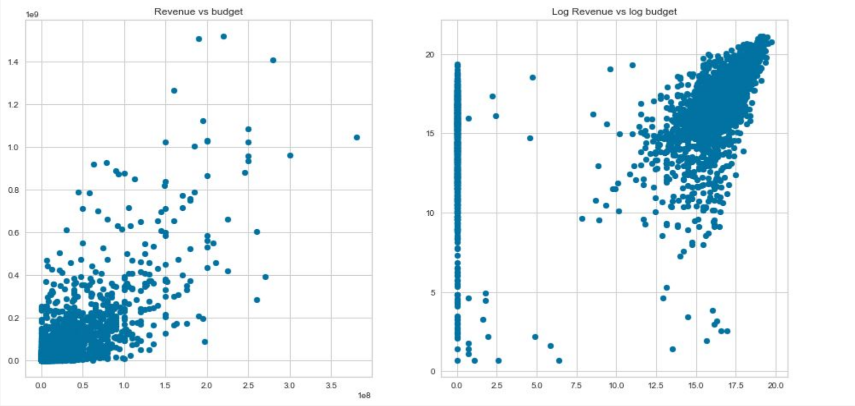
```
master['collection_name'] = master['belongs_to_collection'].apply(lambda x: x[0]['name'] if x != {} else 0)
master['has_collection'] = master['belongs_to_collection'].apply(lambda x: len(x) if x != {} else 0)

master = master.drop(['belongs_to_collection'], axis=1)
```

Data Exploration

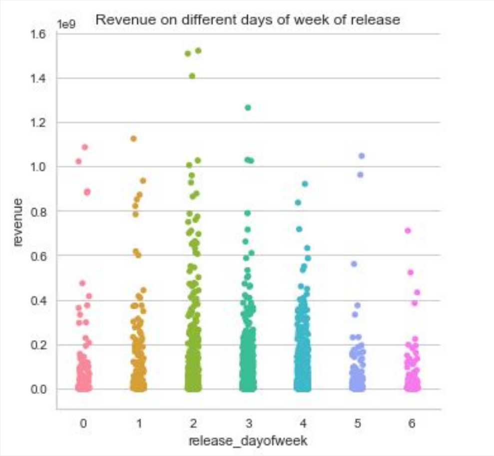
- **Revenue distribution has a high skewness, so we use logarithm transformation of revenue.**



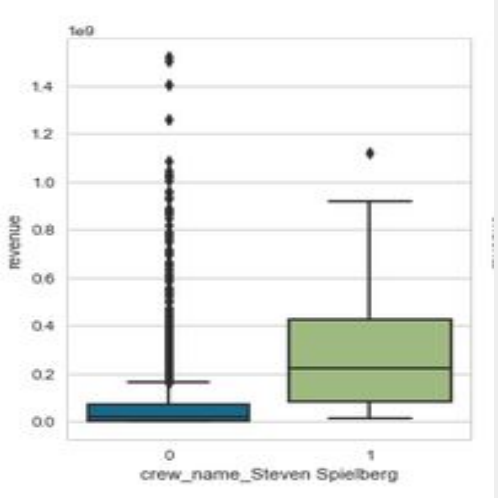
- **We can see some clear trends that an increase in budget tend to lead to higher revenue.**



- **Films released on Wednesdays and on Thursdays tend to have a higher revenue.**



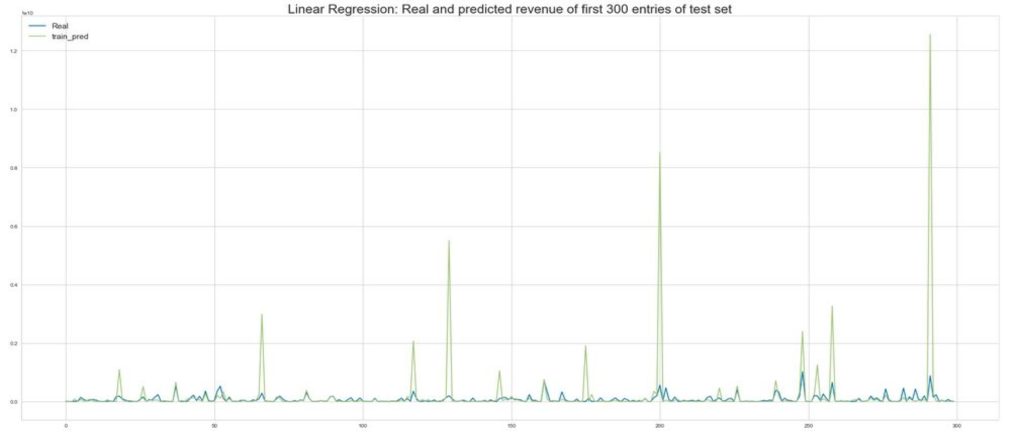
- **Films with Steven Spielberg tend to have higher revenue.**

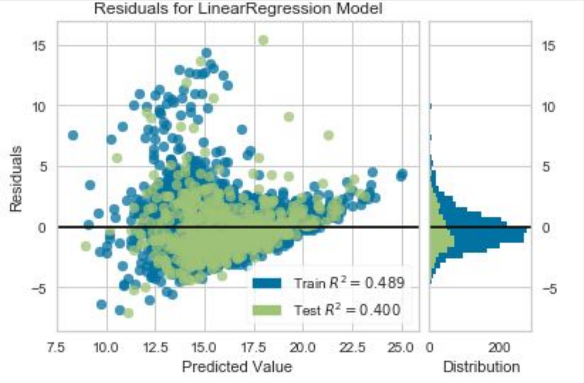


Machine Learning

Linear Regression

- **231 features**
- **80% training set, 20% test set**
- **RMSE - 2.33**
- **Test R^2 - 40%**

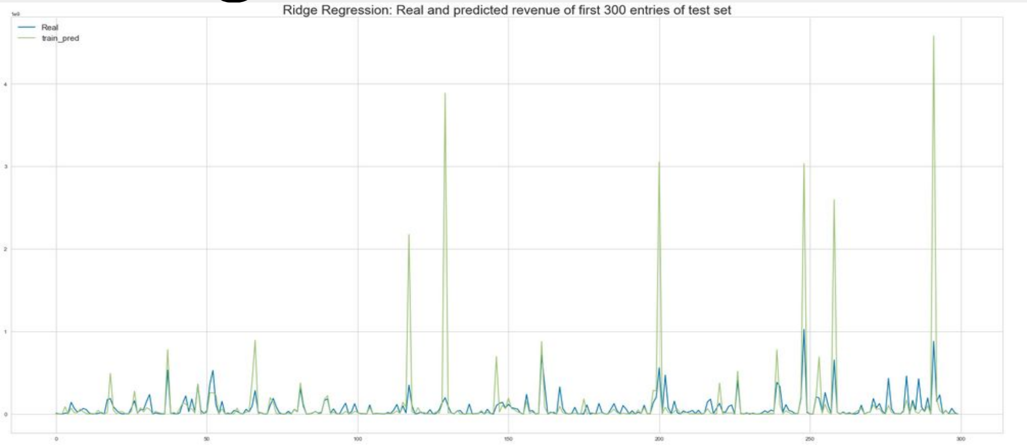


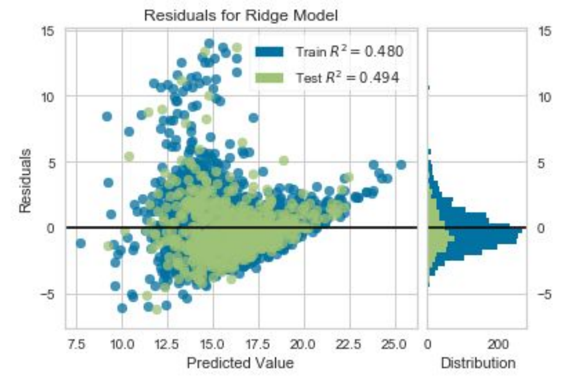


Machine Learning

Ridge Regression

- 231 features
- 80% training set, 20% test set
- RMSE - 2.14
- Test R^2 - 48%





Xgboost

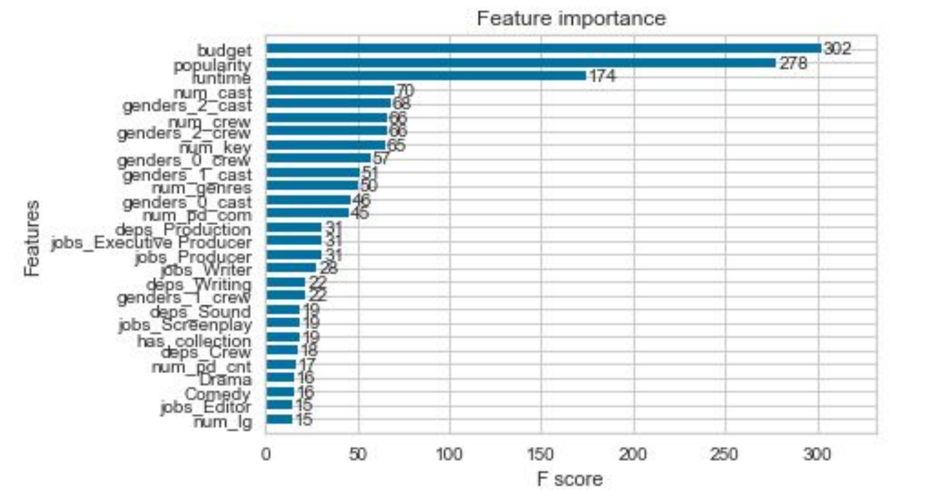
Fine tune parameters:

- **'max_depth' - the maximum depth of a tree**
- **'learning_rate'- makes the model more robust by shrinking the weights on each step**
- **'min_child_weight' - defines the minimum sum of weights of all observations required in a child**

RMSE 0.43

```
param_grid={'booster': 'gbtree', 'colsample_bytree': 0.9, 'lambda': 1.0,
            'learning_rate': 0.3, 'max_depth': 6, 'min_child_weight': 1,
            'nthread': -1, 'objective': 'reg:linear', 'silent': 1,
            'subsample': 0.9}
watchlist = [(dtrain, 'train'), (dtest,'test')]
eval_set = [(X_test, y_test)]
model_2 = xgb.train(param_grid, dtrain, 500, watchlist, early_stopping_rounds=50,
                    maximize=False, verbose_eval=0)
```

```
_=xgb.plot_importance(model_2, max_num_features=28, height=0.7)
```



Conclusion

- Film companies can use this model to predict their revenue
- Production companies can use high impact features to revenue on the planning stage
- Budget, popularity and runtime are top 3 important features