AA306 Final Project: Phase 3 - Progress Update

Group 1 - Varun Arvind, Sarah Freeman, Stephen Grivers

Overview

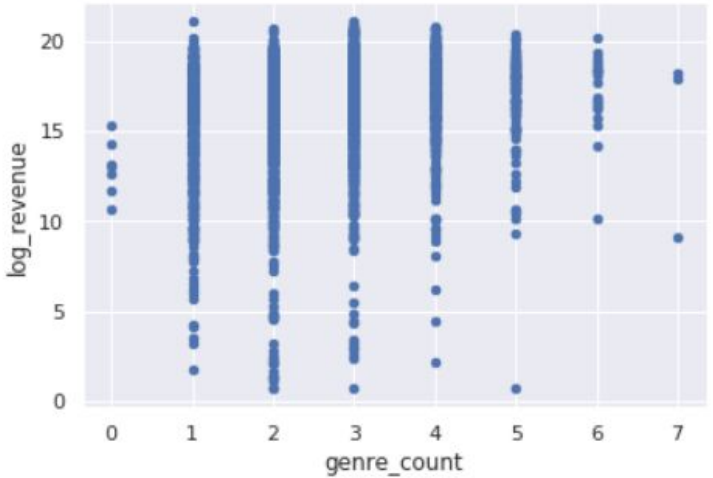
- Project Description
- EDA
- Feature Engineering and Pipelines
- Results
- Conclusion

Project Description

- Goal: to predict the overall worldwide box office revenue for each movie
- 7,398 movies (3,000 for training and 4,398 for testing)
- Input features
  - cast, crew, plot keywords, budget, posters, release dates, languages, production companies, and countries
- Phase 3
  - Feature engineering on production_countries, production_companies, spoken_languages, Keywords, cast, and crew
  - Kaggle test

Sample EDA


Revenue by Month
1e10
revenue
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7
8
9
10
11
12
release_month


log_revenue
20
15
10
5
0
0
1
2
3
4
5
6
7
genre_count

Phase 3 Feature Engineering

- Production Countries, Production Companies, Spoken languages, Keywords
  - Engineered the same way as Genres in Phase 2
    - De-JSONed, turned into CSVs, extracted names

- Cast, Crew
  - We were unable to run the full model once we added these in the same way, it would crash the kernel and cause a `TerminatedWorkerError`.
  - Survival plan: We just added a count of the number of Cast and Crew members each crew had to these two features

- LogTransformer
  - We added a LogTransformer class to replace our one-liners that took the log of some numerical features in Phase 2

Pipelines

```
data_pipeline = ColumnTransformer( transformers= [
        ("num", num_pipeline, numerical_features),
        ("cat_pipeline", cat_pipeline, categorical_features),
        ('date_pipeline', date_pipeline, date_feature),
        ('log_transformer', log_pipeline, log_features),
        ('count_pipeline', count_pipeline, count_features)
],
         remainder='drop',
        n_jobs=-1
    )
```

GridSearch

- Changed our model predictor to KNN and ran a gridsearch to find the best parameters

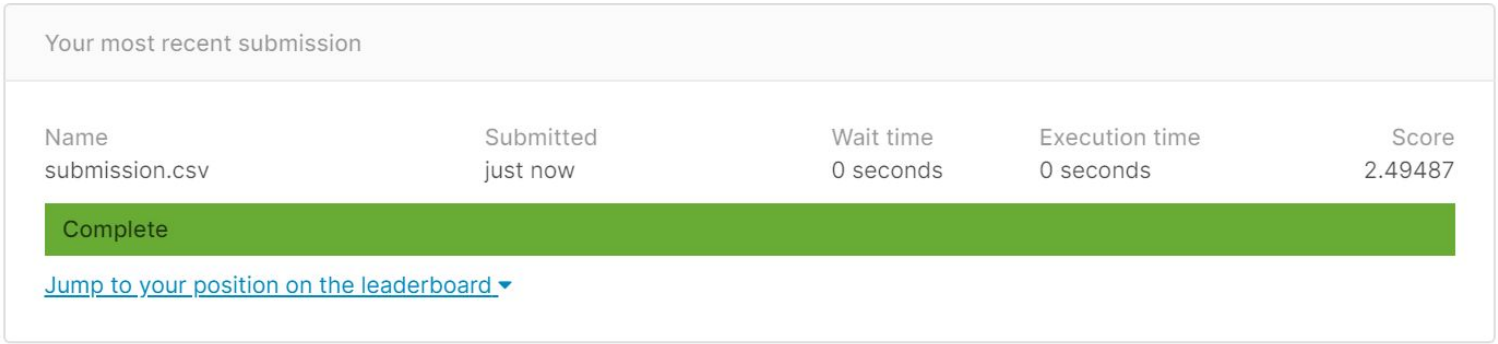
- Linear Regression gave too many infinite predictions!

```
clf_pipe = make_pipeline(
    data_pipeline,
    KNeighborsRegressor())


param_grid = {
    'kneighborsregressor__n_neighbors': list(range(1,6)),
    'kneighborsregressor__weights': ['uniform', 'distance'],
    'kneighborsregressor__algorithm': ['ball_tree', 'kd_tree', 'brute'],
    'kneighborsregressor__leaf_size': list(range(29,32)),
    'kneighborsregressor__p': list(range(1,4))
            }

grid = GridSearchCV(estimator=clf_pipe, param_grid=param_grid,
                    cv=3, scoring='neg_mean_squared_error' ,n_jobs=-1)

start = time()
grid.fit(X_train,y_train)
train_time = np.round(time() - start, 4)

print(grid.best_params_)
```

GridSearch took about 1.5 hours!

Phase 3 Results

- Kaggle Score with All Features:
- Best Kaggle Score: 2.49487
- Puts us at about 900 place on the leaderboard



- Improvement of 0.03992!

Project Conclusion

We were not as successful as we hoped engineering the Phase 3 features.

The model only improved by an extremely small margin: **0.03992**

If we had more time, we would learn a better way to engineer the Phase 3 features, and also run the GridSearch on many different models and hyperparameters.

Burnout was a problem for this project: we spent many hours and didn't gain much!

Thank you!