Machine Learning Group Project

Revenue Prediction for Movies

Abhas Goyal (agoyal@tcd.ie)
Vishal Ruhil (ruhilv@tcd.ie)
Anuradha Vishwakarma (vishwaka@tcd.ie)

1 INTRODUCTION

This project was undertaken with the aim of predicting revenue generated by a movie in the conception stage of a movie, that is having only the basic information available. Using the results of this model, studios can do a cost benefit analysis and decide whether they want to go ahead with the production of a movie, re-evaluate the allocated budget and other factors to make it a profitable venture or scrap the idea. What makes the project interesting is the large number of features and data points available, making feature selection and dimensionality reduction a key aspect of this project.

For this project, Linear, Ridge and Lasso regressions were explored. The required datasets were extracted from TheMoviedb <insert reference>, processed and then imported into the three models to get the predictions. Since getting actual data to test the model on is confidential as studios would not want this information getting released to the public, so to test the model, 10 percent of the data collected was marked as test set and used for validation purposes.

The model can be deployed as an application to be used by the studios with the help of a user interface which will take input in the shape of a form with required fields to be filled out and give the revenue prediction.

2 DATA COLLECTION

The dataset was collected from themoviedb [9] through the use of APIs made available to registered users. The source was chosen because of the volume of data available along with no throttle limits on API calls for getting information. As a first step, list of available genres were extracted using Genre API [3]. For each genre, 5000 movie ids were extracted from the Discover movie API [2], which resulted in 75,000 titles. Since a movie can have multiple genres, the complete list was de-duplicated based on movie ID, which gave 45,000 unique ids . Details for each unique movie such as budget, runtime, popularity, collection details, that is if the movie belonged to a part of series, cast and crew involved in the movie [6][1][5]. The data received from APIs was in JSON format, which was processed using Json library and the output was stored in csv files.

3 DATA PROCESSING

3.1 Identifying important features

The files created as an output from the data collection stage were read in dataframes using pandas library. All the rows which had no information were removed the dataframe. Post that, based on group discussion we dropped specific columns from the dataset which were deemed as information that would not be available at the time of conception stage of a movie, for example, vote average and vote count is information collected from user after the release, runtime and popularity would again be available once the movie has been released. Next, features that were considered to be not relevant in revenue prediction were dropped from the dataset, such as poster path, video, status etc. Language was not considered as the dataset extracted was pertaining only to movies that had original language as english.

3.2 Feature creation and modifications

Next, *collection popularity* field was created by taking the average of popularity of all the movies in a collection. Along with that *Part of series* was created, which is a binary field to track if a movie is a part of existing series. Genre for movies, gender for cast and crew were converted to boolean features with one hot encoding method. And from *belongs to collection* field, id of the collection was extracted so that the data could be joined based on it. Two other boolean fields, namely *Acting Relevance* and *Directing Relevance* were also created, which were labelled as True if an actor was identified as cast or director as crew, for all other cases it was labelled as 0.

3.3 Data Merging and Final dataset

Finally, the movie, cast and crew datasets were joined based on *id* field and collection details was joined using *collection id*. The output contained a total of 105 features, majorly due to one hot encoding of genres and gender fields. We address dimensionality reduction later.

Finally, after merging the datasets, all the records for which *Revenue* field, dependent variable, was found to be 0, were excluded from the analysis as this a required field in supervised machine learning model. With this, the overall dataset was reduced to 7,900 lines.

3.4 Normalization

It was observed that the scale of budget and revenue field was much higher in terms of magnitude compared to all the other fields and was thus standardized with the help of MinMaxScaler, that transforms a field between a 0 and 1 in this case as there are no negative values [4]. The formula used is :

$$\mathbf{X_{std}} = (\mathbf{X} - \mathbf{X.min})/(\mathbf{X.max} - \mathbf{X.min})$$

$$\mathbf{X_{scaled}} = \mathbf{X_{std}} * (\mathbf{max} - \mathbf{min}) + \mathbf{min}$$

MinMaxScaler was chosen over StandardScaler as the latter scales the values around zero mean and range -1 to 1, and since all the values are positive, MinMaxScaler was the appropriate choice.

3.5 Dimensionality reduction

Due to the high number of features present in the final dataset, multi-collinearity was explored between the features. Since the

model uses regression, the independent features should not be highly correlated because if they are correlated, the model can't determine the effect of both the variables properly as both change together [7]. Using *df.corr* method, correlation between independent features were found and for all the features found to more than 0.5 correlated with dependant variable, multi-collinearity using variance inflation factor [7], which is a method to take each feature as dependant variable and regress it against the other variables present, was explored for them. All independent features that had a multi-collinearity factor of over 5 were excluded from the dataset.

3.6 Other Considerations

Since the data collected spans across a multiple decades, taking inflation into account for budget and revenue fields was considered, however it was found that same approach had been followed and did not contribute a lot in explaining the dependent feature [8].

3.7 Missing values

For values that were reported as 0 in budget feature, assumed as data not collected, mean of the budget column was used to fill these rows.

4 METHODOLOGY

Linear Regression, Ridge Regression and Lasso Regression models were used and compared against each other, along with a dummy regressor to establish a baseline model and evaluate the performance of the best model against it.

4.1 Linear Regression

Linear regression model takes the n independent features as input and a target variable to predict as the output. The model assigns weights or constants to each feature and tries to minimize the cost function $J(\theta_0, \theta_1)$, calculated as the mean square error between the true labels and model prediction by changing the weights of each feature.

$$\mathbf{J}(\theta_0, \theta_1) = \frac{1}{2\mathbf{m}} \sum_{\mathbf{i}=1}^{\mathbf{m}} (\mathbf{h}_\theta(\mathbf{x}^{(\mathbf{i})}) - (\mathbf{y}^{(\mathbf{i})}))^2$$

4.2 Ridge Regression

Ridge regression works the same way as Linear regression with the addition of penalty term to the cost function known as L2 penalty. L2 penalty is adding the squares of coefficients or weights or features, which when added to the cost function forces the model to keep the weights low. The penalty term also has a hyperparameter C, which is inversely proportional to penalty, meaning that with larger values of C, the penalty term loses its significance and ridge regression model behaves as a linear regression model. However, decreasing the magnitude of C, the penalty term goes up exponentially and the model minimizes the coefficients.

$$\mathbf{J}(\theta_0, \theta_1) = \frac{1}{2\mathbf{m}} \sum_{\mathbf{i}=1}^{\mathbf{m}} (\mathbf{h}_\theta(\mathbf{x}^{(\mathbf{i})}) - (\mathbf{y}^{(\mathbf{i})}))^2 + \alpha \sum_{\mathbf{j}=1}^{\mathbf{m}} (\theta_\mathbf{j})^2$$

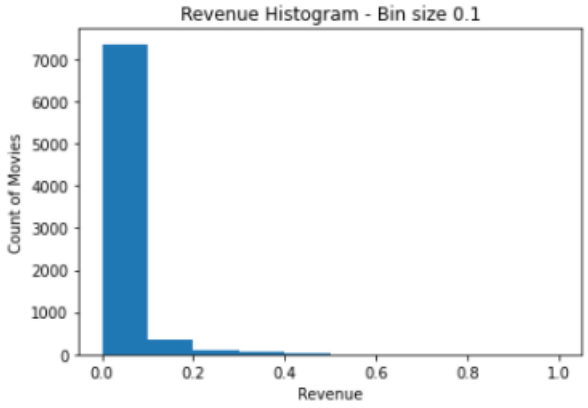


Figure 1: Histogram on Revenue field with bin size 0.1

4.3 Lasso Regression

Lasso regression is similar to Ridge regression with the only change of the penalty term. Cost function in this model has L1 penalty, which add the the coefficients to the cost function rather than the square of coefficients as in the case of Ridge regression. The important distinction between the two penalties is that L1 makes the weights of all non important features 0 whereas L2 makes them really low, but still considers there effect.

$$\mathbf{J}(\theta_0, \theta_1) = \frac{1}{2\mathbf{m}} \sum_{\mathbf{i}=1}^{\mathbf{m}} (\mathbf{h}_\theta(\mathbf{x}^{(\mathbf{i})}) - (\mathbf{y}^{(\mathbf{i})}))^2 + \alpha \sum_{\mathbf{j}=1}^{\mathbf{m}} (\theta_\mathbf{j})$$

4.4 Baseline Predictor

To establish a baseline for regression models discussed above, a dummy regressor was created as a baseline for the above described models.In order to determine the strategy for this regressor, we examined the distribution of target variable. From Figure 1, it was observed that the target variable does not follow a normal distribution but a Poisson distribution, so selecting mean as a strategy would not have been optimal. Therefore, a dummy regressor with 0.5 quantile strategy was selected to get the best results from baseline predictor. The aim is to create a model that predicts revenue better than this baseline model.

4.5 Data segregation

The data was split into three parts, train, validate and test. For training, 70 percent of the data was allocated, 20 percent to validate, that is to evaluate how the model performs and remaining 10 percent for testing the model.

4.6 Model training, evaluation and Hyperparameter tuning

In order to train the model, training data was split into 5 sets using kfold method. For each split, the model was fit and predictions were generated on the validation set. For evaluation of model, root mean square error was considered, which gives information about the difference in values of actual values to the predicted values, and the average of the 5 runs for each model was taken as the final score.

Table 1: Baseline Predictor Results - 0.5 Quantile Strategy

Dataset/Evaluation Metric	R^2	RMSE
Train	-0.126	0.059
Test	-0.121	0.064

Table 2: Baseline Predictor Results - Other Strategies

Dataset/Evaluation Metric		R^2	RMSE
Mean	Train	0.06	0.055
	Test	-0.00051	0.055
Median	Train	-0.126	0.059
	Test	-0.121	0.064

For this error, the range varies from 0 to ∞ and the lower the values of error, the better the model is performing Another statistic chosen to evaluate the model is R^2 statistic, which gives information about how much of the variance in dependent variable is explained by the independent features. The range for this statistic is 0 to 1 and higher the value of R^2, the better the model is performing.

For hyperparameter tuning in Ridge and Lasso regressions, we performed cross validation with 5 kfold sets over a range of C values. For each C value, average of mean square error and its standard deviation was plotted on errorbar, and the lowest value of C for which the error term was the minimum was selected as the hyperparameter value.

4.7 Under and over fitting

A model that has bad accuracy for both train and test set is classified as an underfitted model, that is it completely misses the point of the model. On the otherhand if a model predicts very well on the train set but performs poorly on the test set, it is classified as overfitting model.

5 RESULTS

5.1 Baseline Regression Model

From Table 1, we can see that R^2 score is in negative, which means that the model is very bad in predicting the revenue. Therefore, baseline predictor with other strategies were explored, refer to Table 2. On the basis of above results, dummy regressor with mean strategy was chosen as the baseline predictor.

5.2 Linear Regression Model

To train the model, data was split into train and test. The train data set was further split into train and validate using kfold with n = 5. From Figure 2 and Figure 3, it was observed that the model was performing similar for train and test sets but validate set was not performing the same. This can be interpreted as underfitting of the model in certain cases and can be avoided by adding more data for training, however due to unavailability of more data with labelled target variable, this could not be incorporated. Overall, the average of all runs gave R^2 score of 0.559 and RMSE score of 0.0405

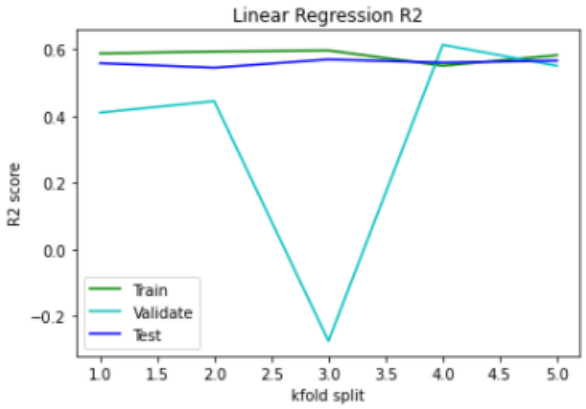


Figure 2: Linear Regression R^2 graph

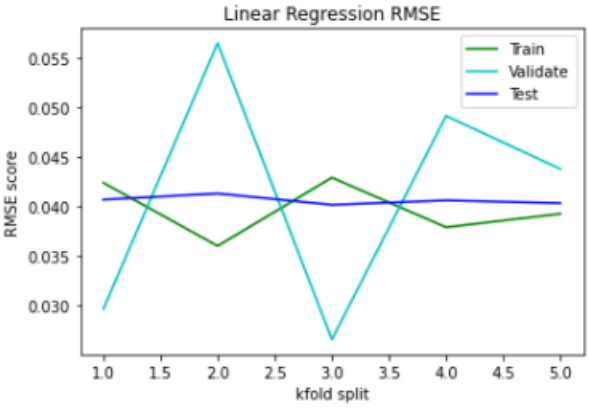


Figure 3: Linear Regression RMSE graph

5.3 Ridge Regression Model

For this model, cross validation was used to select the optimal value of hyperparameter to obtain the max R^2 and minimum RMSE score. The C values for this model were chosen between the range of 0.001 to 0.5. From Figure 4 and Figure 4, it was observed that the model performs best at C = 0.1. Therefore, the final Ridge model was trained with $\alpha = 1/2 * 0.1$ and R^2 score obtained was 0.588 and RMSE score was 0.036 on test data set.

5.4 Lasso Regression Model

For this model, cross validation was used to select the optimal value of hyperparameter to obtain the max R^2 and minimum RMSE score. The C values for this model were chosen between the range of 0.1 to 10000. From Figure 6 and Figure 6, it was observed that the model performs best at C = 1000. Therefore, the final Lasso model was trained with $\alpha = 1/2 * 1000$ and R^2 score obtained was 0.534 and RMSE score was 0.043 on test data set.

6 CONCLUSION

Based on the results obtained above, a comparison was performed of all three models trained and with baseline predictor as well.

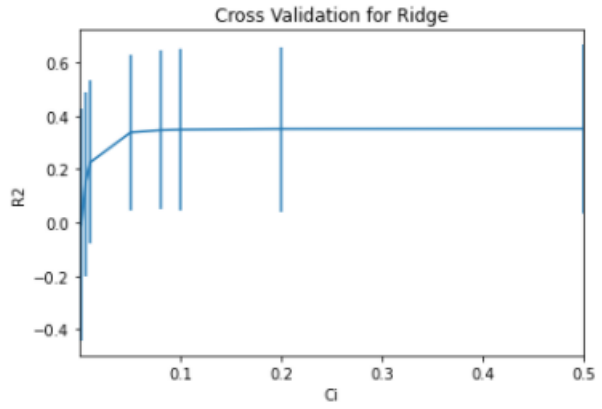


Figure 4: Ridge Regression R^2 cross validation graph

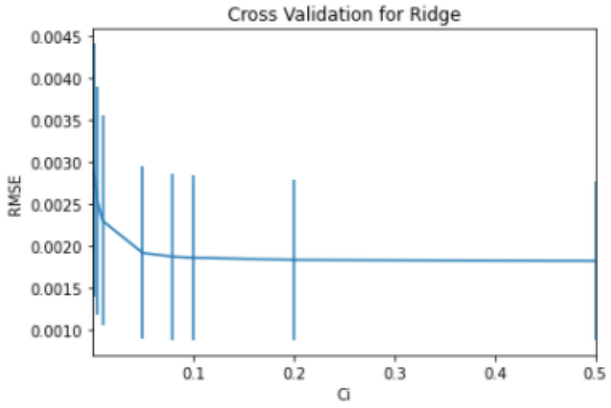


Figure 5: Ridge Regression RMSE cross validation graph

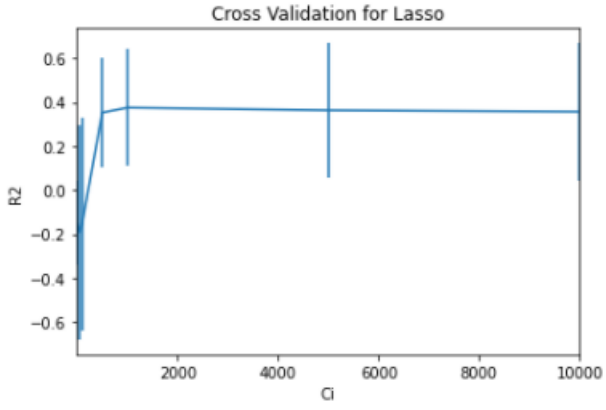


Figure 6: Lasso Regression R^2 cross validation graph

From Table 3, it can be seen that Ridge regression performs the best amongst all the models. Ridge regression model is able to explain 58.8 percent of the variance in revenue price and the average error in prediction of revenue by this model is 0.036, or an error of $10

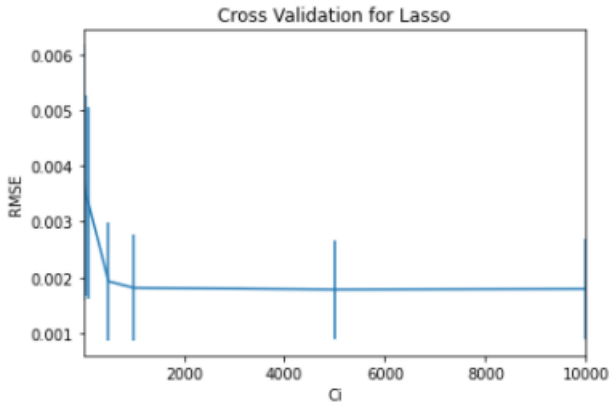


Figure 7: Lasso Regression RMSE cross validation graph

Table 3: Model Comparison

Scoring/Model	Baseline	Linear	Ridge	Lasso
R^2	0.06	0.559	0.588	0.534
RMSE	0.055	0.0405	0.036	0.043

million. When looking at re-scaled values, the error term is quite significant but when compared to the maximum value of revenue, which is in billions, the error term is still small.

The model can be further improved by feature engineering, taking inflation into account for budget and revenue, having a calendar incorporated with holidays and peak seasons to compare with release date, as that can have a significant impact on the revenue generated.

7 CONTRIBUTIONS

7.1 Abhas Goyal

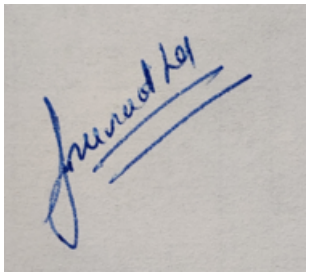
Code contribution : Joining the collected data and all of the preprocessing steps, including multi-collinearity detection leading up to the model training.

Report contribution : Wrote the Data Processing, Methodology and Conclusion sections.

7.2 Anuradha Vishwakarma

Code contribution : Wrote the code for all four models, baseline, Linear, Ridge and Lasso, and charts created for result comparison.

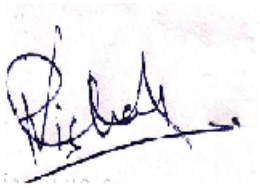
Report Contribution : Wrote the Introduction and Results sections.



7.3 Vishal Ruhil

Code contribution : Wrote the code for collecting data from tmdb APIs.

Report Contribution : Wrote the Data Collection section.



All of the code and files can be accessed on github

REFERENCES

[1] [n. d.]. Collection details. https://developers.themoviedb.org/3/collections/get-collection-details

[2] [n. d.]. Discover. https://developers.themoviedb.org/3/discover/movie-discover

[3] [n. d.]. Genre. https://developers.themoviedb.org/3/genres/get-movie-list

[4] [n. d.]. MinMaxScaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[5] [n. d.]. Movie credits. https://developers.themoviedb.org/3/movies/get-movie-credits

[6] [n. d.]. Movie details. https://developers.themoviedb.org/3/movies/get-movie-details

[7] [n. d.]. Multi-Collinearity. https://towardsdatascience.com/statistics-in-python-collinearity-and-multicollinearity-4cc4dcd82b3f

[8] [n. d.]. Revenue Predictor Project. https://towardsdatascience.com/what-makes-a-successful-film-predicting-a-films-revenue-and-user-rating-with-machine-learning-e2d1b42365e7

[9] [n. d.]. tmdb. https://www.themoviedb.org/settings/api