Machine Learning Final Project*

Movie box office prediction

Pingyi Chen†
University of Pittsburgh
pittsburgh, PA
pic17@pitt.edu

Yimeng Liu‡
University of Pittsburgh
Pittsburgh, PA
yil182@pitt.edu

Yue Li§
University of Pittsburgh
city, country
yul195@pitt.edu

ABSTRACT

Movie industry has a huge market and profit. Predicting the movies performance before it released or at the beginning of their releases could help the cinema to schedule the show times for new movies. Previous works on movie prediction using machine learning approaches to forecast the success of each movie. These studies are mainly methodologically oriented, and focus on the revenue or the rate of new movies. However, in addition to the revenue and rate of the movie, we believe the popularity of a movie is also important. And on a perspective of the cinema, the popularity of a movie is even more important than revenue and rate.

In this study, we focus on the basic features of a movie, i.e. category, language, budget, rate, count of rate, revenue, release time and runtime to predict the popularity of a movie. We implement the machine learning algorithm, such as linear regression, support vector machine(SVM), decision tree, random regression tree,Xgboost and neural network in our models to predict the movie popularity. We found that the best algorithm is linear regression which has an accuracy of 69.10%.

ACM Reference Format:

Pingyi Chen, Yimeng Liu, and Yue Li. 2018. Machine Learning Final Project: Movie box office prediction. In *Proceedings of* . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There are thousands of movies coming out each year. Audience may choose the right movie to see through the prediction of the box office and the movie directors or the produce company would like to learn the prediction of the movies box office so that they can decide the strategy. Our system is aiming for predicting the future box office of a certain movie. Input the key features like director's name, main actors, and the movie type, and then the system would return the possible future box office and how many stars that we would suggest the audience to watch it. Though a success of a movie can be defined on two areas: culture successful and profit, we consider a movie's box office success based on its popularity only.

2 LITERATURE REVIEW

These paper gives us an idea about how our model could focus on and make some difference from the former work. Some of them predict a level of success a movie can be, others of them may also predict genres of the movie from text and reviews of the viewers, however, none of them predict popularity. Still we find some literature that is closely related to our project.

The author[3] talk about the current works on the predicting the movie box office or in the recommended system, all the works are technically- and methodologically-oriented, focusing mainly on what algorithms are better at predicting the movie performance. And in the paper, the author tried a new model called CEM and then examined features by the theory called transmedia storytelling to raise the prediction performance.

Another paper[5] test a series of models on data from companies like IMDb, Rotten Tomatoes, Box Office Mojo and Metacritic. Features are classified into 2 classes, pre-relaese and post-reslease. Features like budget, screens and release date are considered as pre-release, reviews, votes and stars are considered as post-features. But beyond those available features, they also collect some other features in extra, such as effect of actors score and summing up the income of all movies in which that particular actor/actress has starred in. They also proposed a way called one away prediction to calculate accuracy. Prediction is considered true if prediction class is only a class away from the groundtruth.

Weighted linear regression and Polynomial Regression are tested in this paper[6], which gives us another insight to analyse our dataset.

3 RELATED WORK

XGBoost stands for *ExtremeGradientBoosting*, where the term *GradientBoosting* originates from the paper Greedy Function Approximation: A Gradient Boosting Machine, by Friedman. Week learners are processed through this boosting gate to leverage the whole decision making result[4]. This model is tree based, and has a various of learning objective, with which people will be able to suit their data adaptively.

Most of the past studies regarding the movie industry have had the explanatory nature, investigating factors that affect the boxoffice performances of movies. The earliest works include the research conducted by Litman (1983). He has investigated how the production cost, criticsâĂŹ ratings, genre, distributor, release season, and

*Produces the permission block, and copyright information
†Team member
‡Team member
§Team member

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

main actorâĂŹs award history are related to a movieâĂŹs box-office performance. As the movie industry has kept growing since the LitmanâĂŹs study, the exploration of factors affecting movie success has been an interesting research area and thus abounding articles have been published within the area. De Vany and Walls (1999); Elberse (2007), and Nelson and Glotfelty (2012) have examined the relationship between a main actorâĂŹs star power and a movie performance. Basuroy et al.(2003) have investigated how critical reviews affect a movie success, setting star power and budgets as moderators. Prag and Casavant (1994) have had an interest in identifying the relationship between factors such as marketing costs, MPAA ratings, and sequels and a movie success.
Recently, based on the knowledge accumulated from these studies, a few researchers have begun to conduct the studies that have the predictive characteristic. For example, forecasting the movies that are highly possible to succeed is one of the types of such research. Asur and Huberman (2010) have used Twitter data to predict a movie success and Mishne and Glance (2006) have predicted movie sales using web blog data.
Based on all the former work, we decide to use SVM, linear regression, decision tree and NN to apply on the data.

4 DATASET

4.1 Description

We used The Movie Database, also know as TMDB. This is a relatively new database that not so much people explored. So far in 2018, there are 4803 open record released on the Internet. This database contains 2 CSV files, one of which contains actor and cast info, the other one contains 20 other movie features.
Popularity is the label that we are trying to explore. It is said on the official website of TMDB that: "popularity is a very important metric here on TMDB. It helps us boost search results, adds an incredibly useful sort value for , and is also kind of fun to see items chart up and down[1]".
Popularity is skewed, ranging from 0-58.5 in 90 percentile, and 10 percentile of 58.5 - 875. Other features are genres, release date, original language, budget, production company, rating, vote, vote count. Some of their distributions can be seen in figure 1.

4.2 Data preprocessing

The original dataset is incomplete and there are also many records that we consider as noise data. So we removed the records which are lacking in certain features. And the dataset is skewed since the revenues of 1428 among 4803 movies are 0, while the average revenue is approximately 82 million. So we also removed the records with 0 revenue.
There are some features we need to modify before we use them. The first one is the release date. We don't need the date as precisely as they provide in the dataset. And we convert the release date to the release season. We abstract 5 of the most popular genres for all the movies, which are Action, Comedy, Drama, Romance and Thriller. And since release season and genre are all categorical variables, we then convert them into dummy variables. The last feature that need to be modified is the language. The most majority of language is English(93.75%). So we encode English as 1 and other languages as 0.

5 METHODOLOGY

5.1 Linear Regression Approach

After preliminary analyzing the variables separately, we applied the linear regression on the dataset first.Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications.This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine. The linear regression is the simplest model and the first model we want to applied with the data.
First, we plot each feature with our target to see the relationship between them so that we can find if it is linear or not. Through this we can construct a dummy model using for the linear regression first.
Second step is to train the model using different features. Because in the dataset all the features ranges different, like the revenue and vote account, we need to use do data processing to make sure the data would be suitable for the linear regression. Here we used the MaxMinScaler in this step.
Third step is to try different combination of features to find our best result. In this project, we used the linear regression model in the python:

$$Y = \beta 0 + \beta 1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ... + \beta_p X_p + \epsilon \quad (1)$$

Then we change the combination and use cross-validation to find the accuracy and R-2 Score to define the model performance. And the best result coefficient is like below:

Features	coefficient
budget	4.97154816
language	0.37131936
runtime	-11.18874213
revenue	17.14089015
vote average	6.69672898
vote count	222.60633052

5.2 Tree Related Regression Models

We tested 3 tree related regressive models: decision tree, random forest and XGBoost. Results shows that, when tested with only 2 features:vote count and votes, random forest shows R^2 of around 0.5. However, we still decide to establish tree models with all features, to make results comparable to other models.
In this section, features such as budget, revenue and vote count are scaled with standard scalers. Also,we tried to remove records with popularity higher than 58.5 as outliers, which is the last 10 percent of the whole dataset. However, the R^2 showed not much improvement.

5.2.1 Decison Tree. Decision Tree Regression Model is built choosing among maximum depths in
[10,20,50,100,200,300,500,700,1000,2000,3000,5000] by cross validation with cv equals to 5. We use "*mse*" to measure the quality of a split. We also tired some new split methods, such as "$friedmanmse$ and "*mae*". However, not only the validation result isn't improve, but the calculation speed also slows down quite a lot. So, we decide

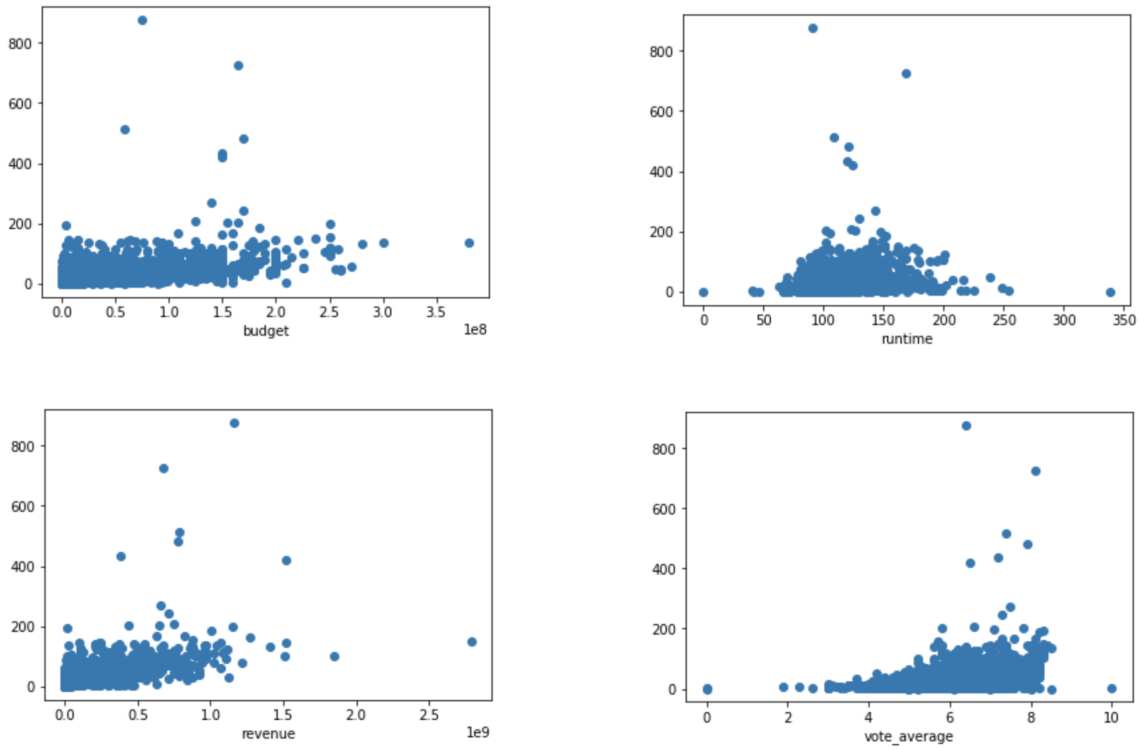

Figure 1: The plot of some features and target

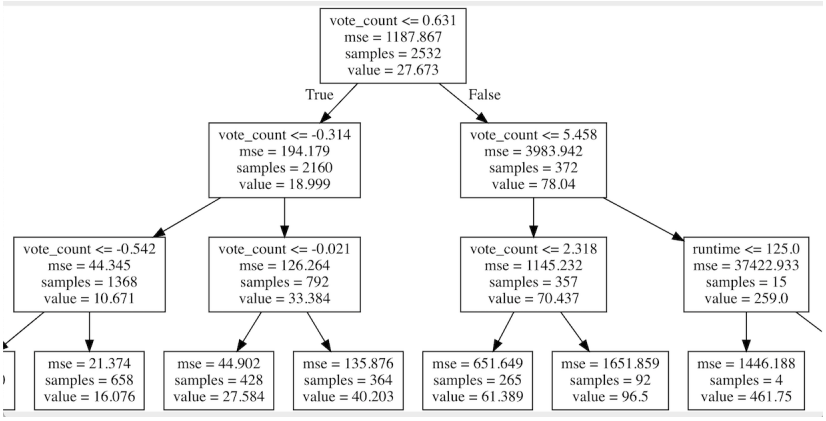

Figure 2: First three layers of our decision tree

to use "*mse*" as a split method. Best result of our test data is 0.23. The tree is visualized in this report.

5.2.2 Random Tree. Random Tree regression model performs the best in our case. Our random tree model selected from cross validation has a R^2 metrics of 0.44 on test data, with a maximum depth of 20, and number of estimators of 10.

5.2.3 XGBoost. XGBoost Regression has a very similar result with Random Tree regression, which is near 0.44. We use the default *gtree* as our booster, and linear regression as our learning objective. Experiments show that with learning rate of 1, which is the maximum learning rate, highest R^2 is obtained. Our final XGBRegressor looks like this:

Parameters	coefficient
min child weight	1
gamma	5
subsample	0.6
colsample bytree	0.8
max depth	7

5.3 Neural Network Models

Neural network is hard to use because the hidden layer is difficult to define. According to Cybenko (1989)[2], one hidden layer is feasible when every processing element utilizes sigmoid transfer function. And two hidden layers can work with parse arbitrary output. Lippmann (1987, 1989) shows that the node number of the second hidden layer is two times the number of output.

However, the studies above classify the outcome value, and predict the class in which the movie should be assigned. In our project, we want to predict the popularity more precisely and we want to get the exact value of the outcome. So we need to build a more complex structure of hidden layer in our model.

We tried one hidden layer structure at first. The range of the number of the units is from 5 to 14. And We found that the best model has 12 units in the hidden layer, and the R-square is 0.4190. Then we added another hidden layer, and tested the performance of two hidden layers model. And the range of the number of units in the first layer is from 9 to 15, the range of the units number in the second layer is from 7 to 11. The best model has 13 units in the first hidden layer and 11 units in the second hidden layer and the R-square score is 0.4326. At last we tested the performance of three hidden layers structure. The range of the number of units in the first layer is from 10 to 14, the range of the number of units in the second layer is from 7 to 11, and the range of the number of units in the third layer is from 2 to 6. And the best structure has 12 units in the first layer, 11 units in the second layer and 4 units in the third layer. And the R-square is 0.427.

The performance is low because the the prediction should be exactly match the actual value, or it will be treated as wrong. It is too strict. For example, if the actual value is 100 and the predicted value is 99 or 101. We believe that the prediction could prove us the correct information and it could be treated as correct prediction. So, we improved our evaluation method. If the predicted value is within ±6 of the target value, we regard the prediction is correct. And the precision of this model increase to 53%. The precision is still low. But if we visualize the predicted popularity and compare it with the actual value in the Figure 3, we found that these two lists of value have a overall linear relation.

5.4 Support Vector Machine(SVM)

A Support Vector Machine(SVM) is a classifier gormamly frfined by a separating hyperplane. The goal is to design hyperplanes that classifies all training vectors in classes.hyperplanes which can classify correctly all the instances in this features have but the best choice will be the hyperplane that leaves the maximum margin from both classes. The margins the closest distance of elements from the hyperplane.

For visualization we have used pythonsâĂŹ Matplotlib library. In our analysis, we applySVC with kernel linear, kernel Gaussian Radial Basis Function (RBF), kernel polynomial. The main difference between SVC with kernel linear and LinearSVC is their implementations. LinearSVC is based on liblinear and SVC kernel linear is based on libsvm library. LinearSVC is more flexible to choose the penalties and loss functions and better on large numbers of samples. Moreover, SVC kernel linear uses one vs one scheme when LinearSVC uses one vs rest scheme for classification. By selecting best parameters we have found different accuracy in different method.

Kernel	ALL FEATURES	BEST RESULT
rbf	0.06807616942967043	0.5801466501803355
linear	0.720195228587404	0.7201952285874049
polynomial	0.43124413709708115	0.7480995237920438

With all the features, we using the features defined in the former method. First, we applied SVM on all the features and get the accuracy below, and use SVM on the features we selected in the former method to evaluate the SVM method in different features.

6 EVALUATION

In our study, we investigate the performance of 5 models, Linear regression, SVM, Decision tree, Random tree and Neural network. And we adopt the R-square and the mean squared error to measure the performance of each prediction model. We also implement cross-validation to select the best parameters in every model. Figure 5 shows the mean squared error of each model. The SVM regression model has the lowest MSE score which is 315. And the performance of Linear Regression model is also well. While the Decision Tree model has the highest MSE. Figure 6 shows the R-squared score of each model. The Linear Regression model has the highest R-squared score, which is 69.10%. And the R-squared score of SVM is a little lower than the Linear Regression model, which is 64.20%. Decision Tree model has the lowest R-squared score. According to the MSE and R-squared score comparison, Linear Regression and SVM perform well comparably. And the Decision Tree' performance is bad.

7 CONCLUSIONS

Low accuracy of tree models on all features are probably due to overfitting problem, because with only 2 features of vote count and votes, the accuracy is much higher. Overfitting of the trees is also proven at cross validation, that the average R^2 is 0.69 during cross validation process which is much higher than 0.44. XGBoost doesn't seem to be a lot better than random tree model might also due to dimension curse problem.

Linear models are robust on performing, even though not very high. But with an mse score of around 400, it is still a good model.

Neural network model performs not good. The problem of this algorithm is that it is very hard to define a good structure of the hidden layer for the model artificially. As we tried 170 structures of the hidden layer, we got the best one which has three hidden layers. However the accuracy is still low. One of the reason might be the predicted target. Predicting the exact value of the popularity is difficult. And the distribution of target data is skewed. The largest popularity is more than 800, while the most majority of the popularity is below 50. If we classify the popularity and convert

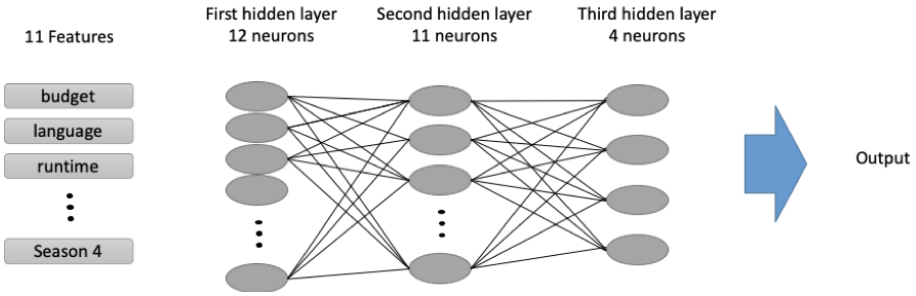

Figure 3: The structure of neural network

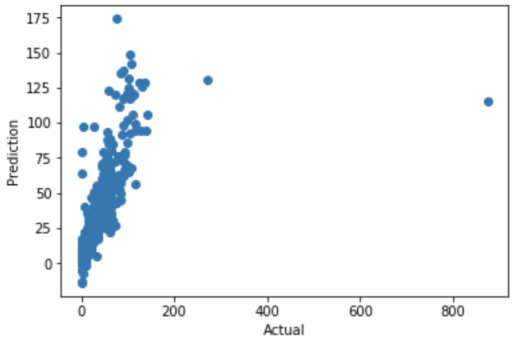

Figure 4: Prediction - Actual popularity

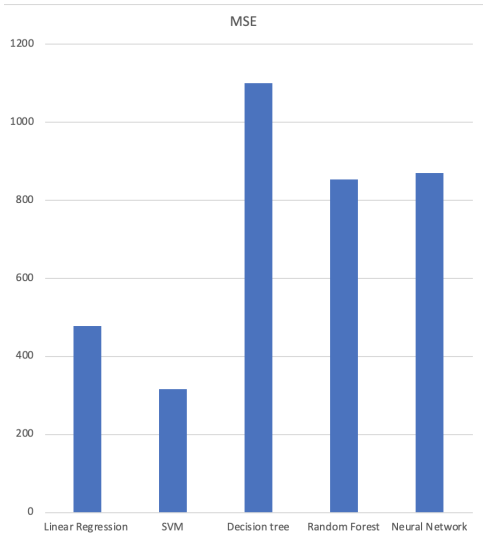

Figure 5: The MSE for each model

the problem to classification the popularity, the accuracy will be much higher. And another reason is the features are limited. We only include a few categories of movies information and there are only 3376 movie records.

So, this research evaluate the performance of all the current models

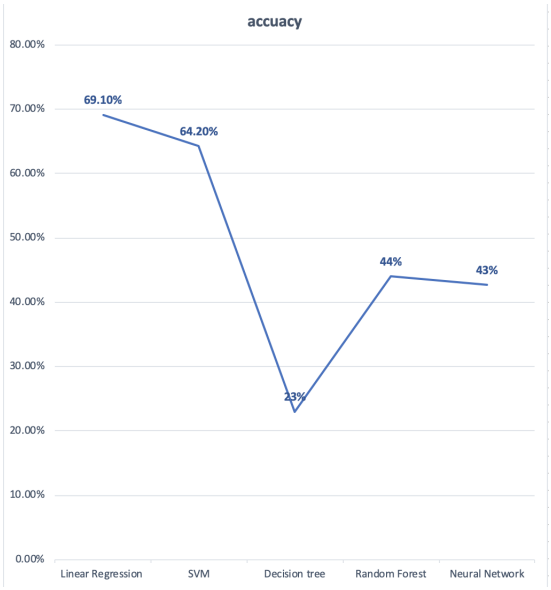

Figure 6: The R-square for each model

we learned in the course when applied on the movie box office. The performance could be determined by numerous elements and aspects and the result could be different because of different dataset. According to out researches, the features – revenue, budget, language and runtime , are the most relevant features in predicting the movie box office. And their relationship is when one is bigger, the other becomes bigger too – nearly linear since the linear regression fit perfect.

For the future work, we will expand the features, such as director, actors, actress and comment reviews. Because these features could be important for the movie's popularity and have more proportion in the prediction. More over, because current model is based on others' work, we can import our own solution of math to build the model. Like the linear regression.

REFERENCES

[1] TMDB API: The Movie Database API
https://developers.themoviedb.org/3/getting-started/popularity

[2] Zhang, Li, Luo, Jianhua, & Yang, Suying(2019). Forecasting box office revenue of movies with BP neural network. *Expert Systems with Applications,36,*6580âĂŞ6587. doi:10.1016/j.eswa.2008.07.064
[3] Lee, Kyuhan and Park, Jinsoo and Kim, Iljoo and Choi, Youngseok(2018).Predicting movie success with machine learning techniques: ways to improve accuracy.*Information Systems Frontiers,20,*577–588.
[4] A Gentle Introduction to XGBoost for Applied Machine Learning
https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
[5] Quader, Nahid, et al(2017).A machine learning approach to predict movie box-office success.*Computer and Information Technology (ICCIT),2017 20th International Conference of. IEEE*
[6] Apte, Nikhil, Mats Forssell, and Anahita Sidhwa. "Predicting Movie Revenue." CS229, Stanford University (2011).
[7] Asur, S., Huberman, B.A.: Predicting the future with social media. CoRR, abs/1003.5699 (2010).
[8] Fox, J.: Applied Regression Analysis, Linear Models, and Related Methods. SAGE Publications (February 1997).
[9] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD 11, 10âĂŞ18 (2009)
[10] Joshi, M., Das, D., Gimpel, K., Smith, N.A.: Movie reviews and revenues: An experiment in text regression. In: Proceedings of NAACL-HLT (2010).
[11] Mishne, G., de Rijke, M.: Capturing global mood levels using blog posts. In: AAAI-CAAW 2006, pp. 145âĂŞ152 (2006)
[12] Tsagkias, E., de Rijke, M., Weerkamp, W.: Predicting the volume of comments on online news stories. In: CIKM 2009, Hong Kong, pp. 1765âĂŞ1768. ACM (2009)
[13] RÂĺuger, S., van Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 191âĂŞ203. Springer, Heidelberg (2010)
[14] Jenkins, H. (2003). Transmedia storytelling: Moving characters from books to films to video games can make them stronger. MIT Technology Review. Retrieved from https://www.technologyreview.com/s/401760/transmediastorytelling/.
[15] Ghiassi, M., Lio, D., Moon, B. (2015). Pre-production forecasting of movie revenues with a dynamic artificial neural network. Expert Systems with Applications, 42(6), 3176âĂŞ3193.
[16] Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. The Journal of Machine Learning Research, 3, 1157âĂŞ1182.
[17] Hastie, T., Tibshirani, R., Friedman, J., Franklin, J. (2009). The elements of statistical learning: data mining, inference and prediction. New York: Springer.