CSE-519
Data Science Fundamentals

PROJECT REPORT

"Predicting the Box Office Gross Collection"

by
Hemant Pandey (110828730)
Naveen Rai (111207633)
Snigdha Kamal (110937472)

TABLE OF CONTENTS

Github link of the project -
https://github.com/nkumarrai/CSE519-2017-111207633/tree/master/Project
Note - this repository is private and shared with the team members and "cse519@cs.stonybrook.edu".

Abstract

The project aims to predict the box office gross sales of upcoming movies. We anticipate that the sales will depend on factors like genre, cast actors, actresses, director, movie budget etc. It would be exciting to validate our predictions on movies releasing in December, just before the project submission.

Introduction

We see a lot of movies getting released almost every weekend (around the world). Some movies perform outstanding on the box office and some fail to get the same response. Popular movies have collected in order of hundreds of millions of dollars during the box-office tenure. There are various factors which determines the opening collection. For instance, If it is christmas, you would like to watch a movie or if it your favourite actor who is playing the lead, there is a high chance you would hit the theatres. In this project, we are going to predict this response in US & Canada - “How much a movie earns during the opening weekend?”

The prediction would be based on various relevant features associated with the movie like genre, cast, plot keywords, movie budget, directors, day/date of release etc. So, we need to find the most important features which contribute to the movie sales and build an optimum regression model to predict them.

Mid-Review Recap

Here is a recap of all the tasks that we accomplished in the mid-review project progress report:

1. Data Scraping:

The box office opening collection was not available readily through the IMDbPY library. We had therefore scraped the data related to box office collections of a movie from the IMDB website using the beautifulsoup and then parsed the generated xml to get the box-office gross sales of the movie. These two data sources were our main source for building our predictive models. We planned to increase the size of the dataset for our advanced models.

2. Models:

a. Baseline Model : This model was built by taking the average gross of all the movies grouped by the features. The features that we employed in this model were: ‘genre’, ‘language’ and ‘country’. This model served as our baseline model against which we would be evaluating all other models.

b. Linear Regression Model : We built a linear regression model to predict the gross collection based on three features as follows : 'Budget', 'Worldwide Gross' and 'Release Year'. This model yielded better results compared to our baseline model.

c. To incorporate the features: ‘genre’, ‘language’ and ‘country’, we created three new variables based on these features. We then used it with the other features which are

'Budget', 'Worldwide Gross' and 'Release Year' and build a new linear regression model. The results were better than the baseline model and comparable to the previous linear regression model.

Some of the features that we planned to incorporate in our final project were:

1. Taking into account the inflation in the gross domestic values over the years as our dataset has movies ranging from 1992 to 2016.
2. Incorporating more training dataset for more recent movies for more accurate predictions.

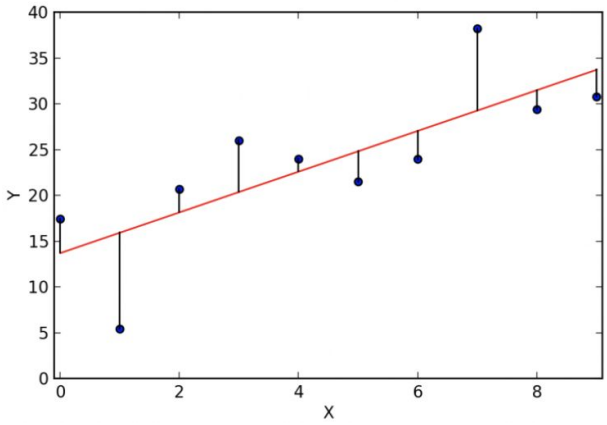
Models Used

Regression Analysis investigates the relationship between a dependent and an independent variable through predictive modelling techniques. It is most commonly used to unearth causal relationships between variables and use them for forecasting and time series modelling. It is due to this reason that we chose to model our problem of predicting the gross sales earnings of a movie through a regression analysis technique by exploring the hidden effects the various features such as budget, language of the movie etc have on the gross sales of the movie.
In regression analysis, we try to fit a line or a curve to the data points in a way which minimizes the differences between the distances of the data points from the line - to yield the line of best fit. Regression Analysis will not only help us to unearth the significant relationships but also reveal the strength of impact of multiple independent variables on our dependent variable.

1. Linear Regression Model

A linear regression model is the simplest and most widely used statistical technique for predictive modelling. It yields an equation relating the features as independent variables to the target variable which is the dependent variable. In our model, the dependent variable is the gross amount earned by the movie and the independent variables used are :
Such an equation takes the form as below where Y is the dependent variable, X's are the independent variables and all thetas are coefficients.

$$\mathbf{Y} = \theta 1X1 + \theta 2X2 + ... + \theta nXn$$



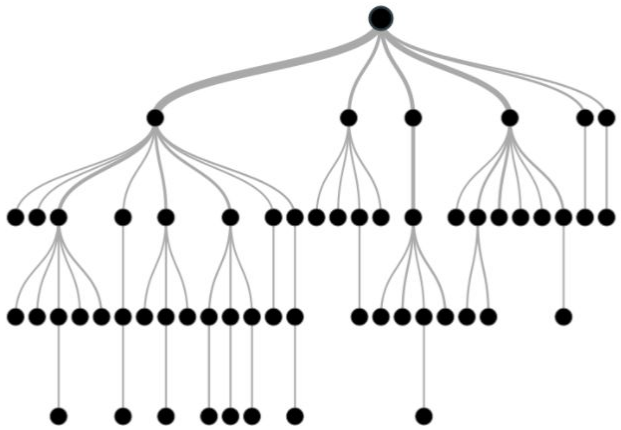
Coefficients represent the weights assigned to the independent features, based on their importance. For example, if we believe that gross income of a movie would have higher dependency upon the budget as compared to the language of the movie, the coefficient of budget would be more than that for the language of the movie. A linear regression equation represents the equation of a line and the problem boils down to choosing the best fit or regression line. The main purpose of the best fit line is to minimize the difference between the predicted and the actual values, called the error as shown below. The residual errors represented by the vertical lines show the difference between predicted and actual value

2. Baseline Model

Baseline model was our first attempt towards this problem. And an attempt to ‘Keep it simple and stupid’. We used features based on ‘genre’, ‘country’ and ‘language’ and found the average of the movies grouped by the features. That was our prediction.

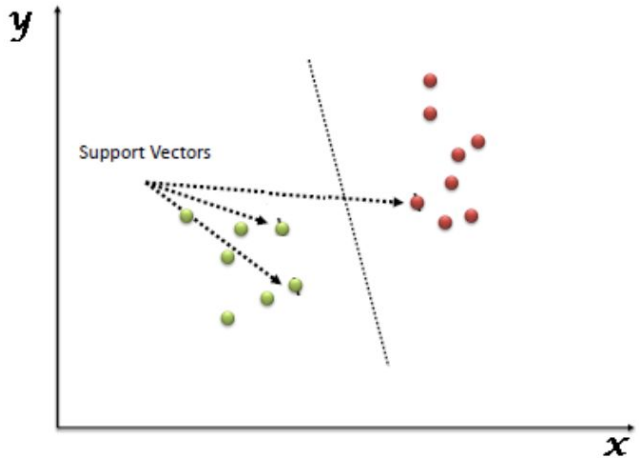
3. Decision Tree Model

Tree-based learning algorithms are one of the most widely used supervised learning methods which empower predictive models with stability and high accuracy. They efficiently map nonlinear relationships which linear models are unable to do. As they can solve regression problems, we have utilized the decision tree model to try and predict the gross movie sales for our project. The decision tree algorithms works by splitting the population into two or more homogenous sub-populations based on the most significant differentiators in the input variables as shown in the figure below.



4. Support Vector Regressor

Support Vector Machines are extremely useful when the data is not linearly separable in the current dimension and this is a possibility that we wished to explore with our dataset. Support Vector Regression is a supervised machine learning algorithm in which each data item is plotted in an n-dimensional space (where n is the number of features) with the value of each coordinate representing the value of each feature. Regression/Classification is then performed by finding the hyperplane which segregates the two classes efficiently as shown below. The correct hyperplane segregates the classes with as large a margin as possible. The regression problem then comes down to finding the optimum margin which separates the features.



Data Scraping & Preprocessing

The screenshot for the python script which generates the IMDB URL mapping for the the list of movies:

```
In [ ]: f = open("name_url_mapping.csv","w")

count = 1
for movie in mList:
    print count
    count+=1
    link = "http://www.imdb.com/find?ref_=nv_sr_fn&q=" + movie + "&s=all"
    skipped_urls = []
    url = ""
    try:
        movie_page = requests.get(link, timeout=30)
        soup = BeautifulSoup(movie_page.content, "xml")
        li = soup.find_all('table',{"class":"findList"})
        #print str(li[0])
        re_obj = re.search('href="(.*?)"><img', str(li[0]))
        if(re_obj):
            url = (re_obj.group(1))
    except Exception as exception:
        skipped_urls.append(link)
        #print(exception)
    print movie,url
    f.write(movie+","+url+"\n")

f.close()
```

```
1
Avatar /title/tt0417299/?ref_=fn_al_tt_1
2
Star Wars Ep. VII: The Force Awakens /title/tt6524062/?ref_=fn_al_tt_1
3
Pirates of the Caribbean: At World's End /title/tt0449088/?ref_=fn_al_tt_1
4
Spectre /title/tt2379713/?ref_=fn_al_tt_1
5
The Dark Knight Rises /title/tt1345836/?ref_=fn_al_tt_1
6
```

Using the URL's from above script, we have scraped various features associated with a movie using BeautifulSoup. We have scraped more data than the mid-review for a better training set. The dataset includes more than 5000 entries. The final data set has 15 attributes which we are going to use in our modelling techniques and a sample can be shown in the following screenshot:

```
In [3]: read_df.head()
```

Out[3]:

ed: 0	Movie	Month	Day	Release Year	Budget($M)	Domestic Gross($M)	Worldwide Gross($M)	URL	Language	Country	Genre	CastID	Rating	DirectorID
0	Avatar	Dec	18	2009	425.0	760.507625	2783.918982	/title/tt0417299/?ref_=fn_al_tt_1	['English']	['USA']	[]	['', '', '', '', '', '', '', '', '', '']	NaN	NaN
1	Star Wars Ep. VII: The Force Awakens	Dec	18	2015	306.0	936.662225	2058.662225	/title/tt6524062/?ref_=fn_al_tt_1	[]	[]	['Talk-Show']	['', '', '', '', '', '', '', '', '', '']	7.1	nm0893659
2	Pirates of the Caribbean: At World's End	May	24	2007	300.0	309.420425	963.420425	/title/tt0449088/?ref_=fn_al_tt_1	['English']	['USA']	['Action', 'Adventure', 'Fantasy']	['nm0000136/', 'nm0001691/', 'nm0089217/', 'nm...	6.8	nm0005222
3	Spectre	Nov	6	2015	300.0	200.074175	879.620923	/title/tt2379713/?ref_=fn_al_tt_1	['English', 'Spanish', 'Italian', 'German', 'F...	['UK', 'USA']	['Action', 'Adventure', 'Thriller']	['nm0185819/', 'nm0910607/', 'nm2244205/', 'nm...	8.4	nm0634240
4	The Dark Knight Rises	Jul	20	2012	275.0	448.139099	1084.439099	/title/tt1345836/?ref_=fn_al_tt_1	['English', 'Arabic']	['UK', 'USA']	['Action', 'Thriller']	['nm0000288/', 'nm0000198/', 'nm0362766/', 'nm...	6.5	nm0893659

There were some redundant entries and unicode strings in the datasets which were cleaned using the pandas library. Also, we kept in mind the inflation of prices across different years to obtain a better prediction model. Using all the required data preprocessing techniques, we have generated the entire dataset into a csv file which is present in the github repository.

- For inflation, we tried some python packages like “economics”, “inflation-cpi” and “easymoney” packages which are available online. Out of these three packages, easymoney worked superbly. With just one api, we were able to apply inflation on all the price related entries in our dataset. (Code snippet follows -)

  ```
  from easymoney.money import EasyPeasy
  ep = EasyPeasy()
  ep.normalize(amount=100,region="USA",from_year=2010,to_year="latest",base_currency="USD", pretty_print=True)
  ```

Implementation details & Results

Throughout the project, we have used mean squared error as our loss in predictions. Let’s start with the baseline method and look at the details of the implementation and results. Also, we divided the dataset into two parts -

a) training set (80% ~ 4000 entries)
b) test set (20% ~ 1000 entries)

1. Baseline model
   a. We had three features ‘genre’, ‘language’ and ‘country’.
   b. We applied the baseline model separately on each of these features. And created the fourth model based on all three features.
   c. Following are the mean squared error losses on the test dataset -

      Baseline model (country) - 2209.57688218
      Baseline model (language) - 1985.75300328
      Baseline model (genre) - 1671.01076743
      Baseline model (country, language, genre) - 1713.6729084
   d. Model with ‘genre’ as the feature has least loss among all four models. But still, the squared loss is in ~1000 range which is pretty bad.
   e. On the positive note, we didn’t expect anything good out of the baseline model. So, let’s look at some more advanced regression models.

2. Linear regression (#1 and #2)
   a. We created two linear regression model as we have mentioned before.
   b. Model #1 includes features: 'Budget', 'Worldwide Gross' and 'Release Year'
   c. Model #2 includes features: 'Budget', 'Worldwide Gross', 'Release Year', ‘genre_v’, ‘language_v’, ‘country_v’.

d. To recap on the new variables in model #2, we first found out all the different type of genres, languages and countries from the complete dataset. And then created the above mentioned variables based on the sum of the entries of the indexes.
e. We ran the model and obtained the below losses on the test data set-
Linear regression model #1 - 223.129695809
Linear regression model #2 - 225.26262933
f. Compared to the baseline model, we have improved our predictions.
g. The predictions of the test dataset are now in (+15/-15) error range now.

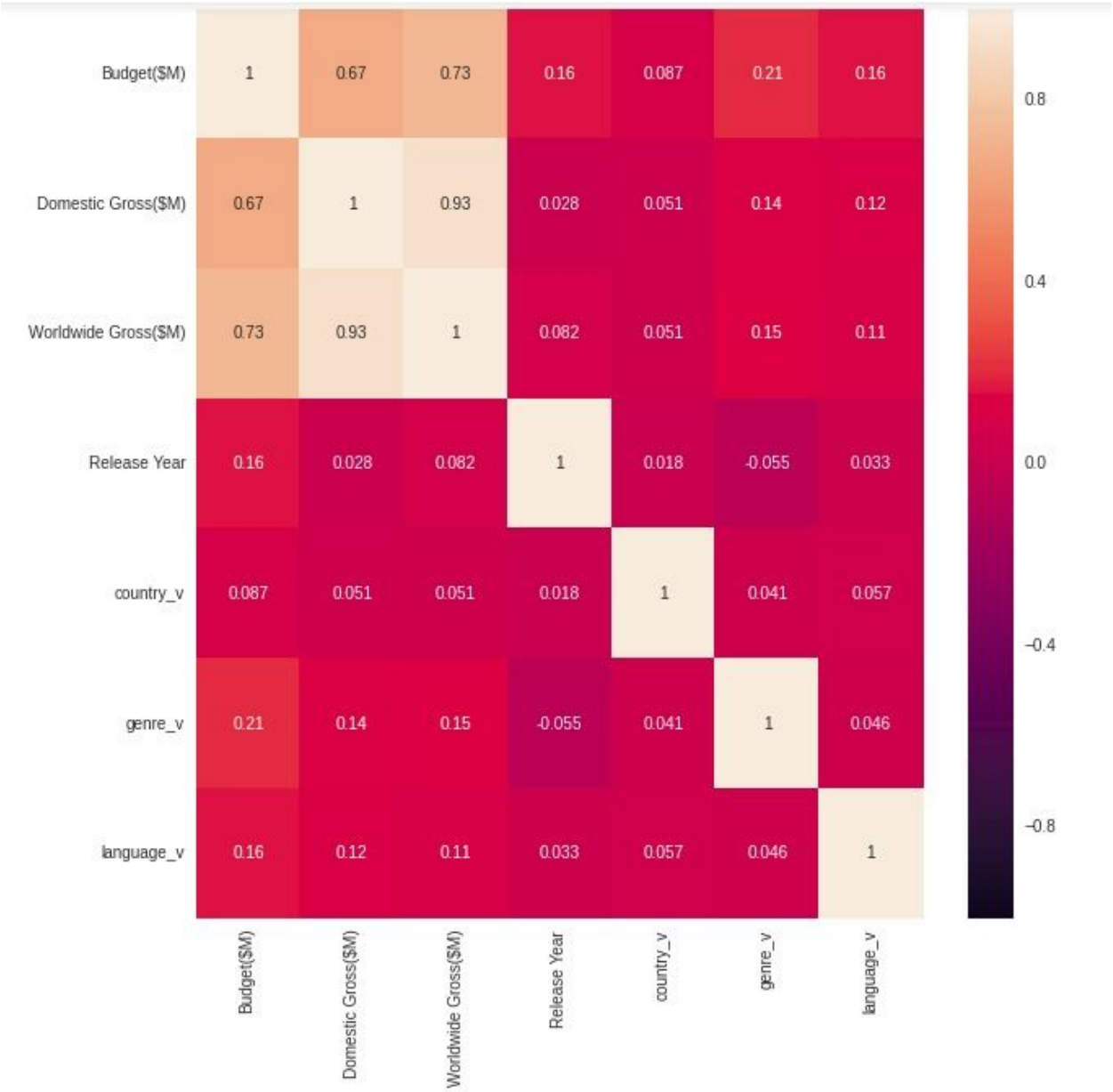
Using the above mentioned six features, we built more advanced models. Our expectation was that 'Decision trees' and 'Support Vector regressor' should outperform the models based on linear regression model.

3. Decision trees
    a. The two parameters for this model are 'max_leaf_nodes' and 'random_state'. We tried to find the best parameters.
    b. For e.g. max_leaf_nodes = 10 provides a test loss of approx 115.
    c. The minimum loss is achieved when the max_leaf_nodes is between 30 - 40.
    d. The minimum loss on the test dataset is 73.4.

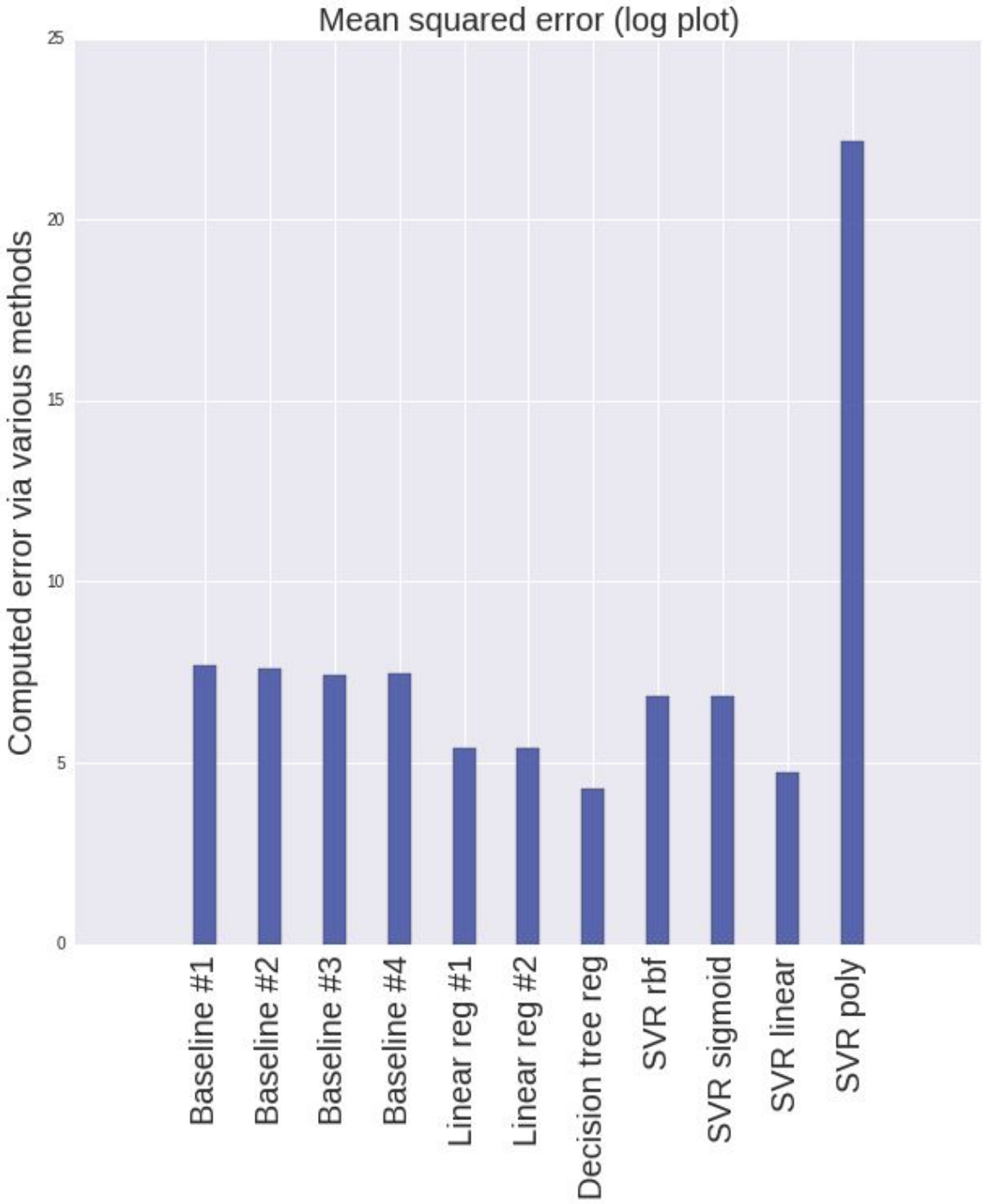
4. Support Vector Regressor
    a. Kernels used: 'Linear', 'RBF', 'Sigmoid', 'Poly'.
    b. We created four models with kernels as mentioned above.
    c. Following are the losses computed on the test data set -
    Support vector regressor (Linear kernel) - 115.001196689
    Support vector regressor (RBF kernel) - 952.946411054
    Support vector regressor (Sigmoid kernel) - 953.67440882
    Support vector regressor (Poly kernel) - 4286140490.77
    d. From the mean squared error losses, it is evident that SVR (linear kernel) outperformed linear regression models, however, other kernels didn't perform so well. They clearly underfit the data by a large amount.

Interesting fact to note is that the decision tree model has outperformed all with the parameters mentioned above. The minimum loss we achieved is ~73 which means the prediction is within error range of (+8/-8).

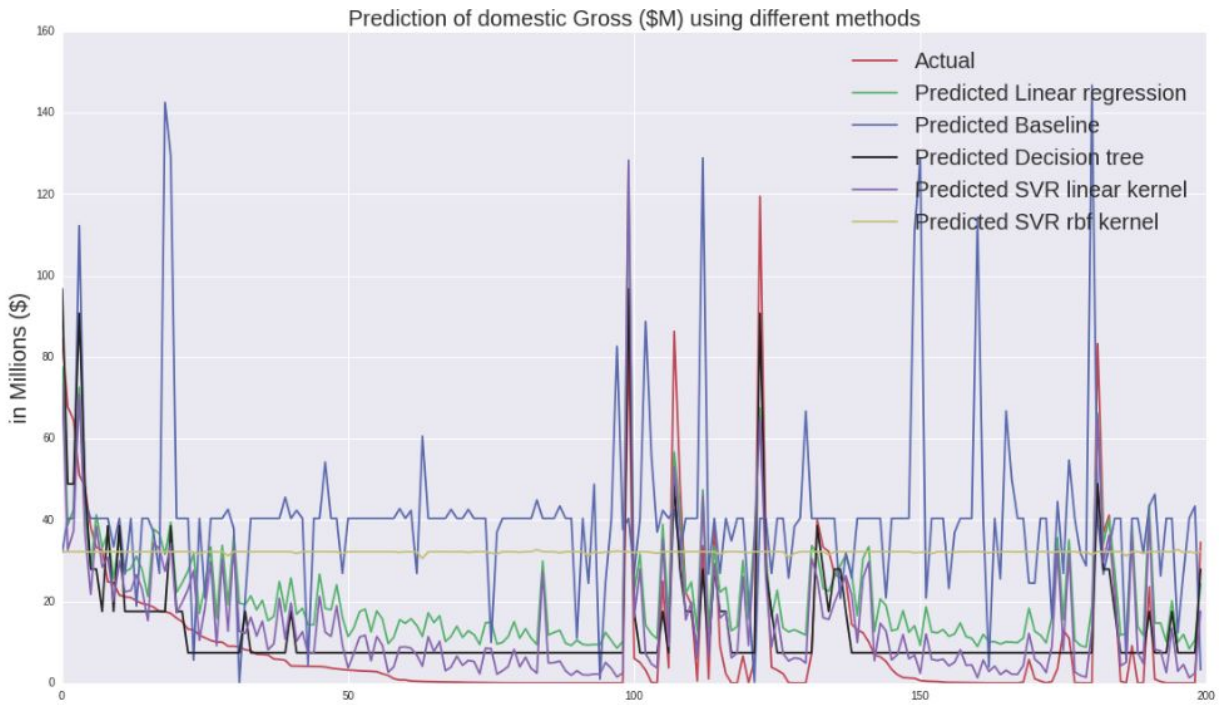
Below are the three plots that we have used to gain insights about the features and for our analysis about the predictions from the models.



a. Correlation Plot between the features used in the regression model (predictor variables) with the target variable(Domestic Gross($M))



b. Plot of log of mean square error versus every model we used for regression



c. Predicted values of the test data set vs the actual test data set (Baseline method, Linear regression, Decision tree, SVR linear kernel and SVR rbf kernel).

Conclusion

Our final project on predicting Gross Movie Sales gave us valuable insights into the data science pipeline - from data scraping, preprocessing and modelling to improving the models and fine tuning parameters for higher accuracy. We initially started with a simple baseline models in accordance with the KISS principle and slowly built upon it using linear regression and then advanced machine learning regression techniques. We learnt the pros and cons of applying every model to our dataset and how parameters should be tuned to increase accuracy - for example non linear kernels in SVR and number of decision tree nodes in decision tree models. In conclusion, this project was a very rewarding and insightful experience.

References

[1] IMDB Dataset: http://www.imdb.com/interfaces/

[2] IMDBPy Python Library : http://imdbpy.sourceforge.net/docs/README.txt

[3] Chen,Perozzi,Skiena,Vector-based similarity measurements for historical figures, Information Systems, 2016. http://dx.doi.org/10.1016/j.is.2016.07.001i

Information courtesy of IMDb (http://www.imdb.com). Used with permission.

[4] Analytics Vidhya for Theory of Models : https://www.analyticsvidhya.com