PREDICTION OF MOVIE SUCCESS USING RANDOM FOREST A MACHINE LEARNING ALGORITHMS AND THEIR COMPARISON

By

Md. Humayun Kabir ID # 2020262003

School of Communication, Business, and Law

Leeds Trinity University, UK

Email: hkabir845@gmail.com

Abstract

Movies are being released in several categories. Some big Hollywood movies are there, and some new movies are also coming out for the industry. Sometimes it happens that a low-budget movie is more popular than a high-budget movie of the same genre. Every movie requires different stakeholders such as the lead actor, supporting actors, the director, and other actors and musicians. The movie's success will be based on various things, such as the actors, the content, and so on. The power of data analytics is not the only prediction of scientific data but also decision making, marketing & economics. The application of data analysis is very extensive. Several diverse businesses like education, construction, commerce, music, and so on. This study uses movie data for data analysis. Data analysis captures useful hidden knowledge from the data. Data is important while it is used well as well as getting patterns. The analysis of data from the business environment can benefit the business organizations in the decision-making process. It will be better to investigate why individuals become successful. Technologically based industries get success in the market because of the intelligent analysis of data and making a decision based on the outcomes of such analysis. Some data-driven operations such as eBay, Amazon, and Facebook are using users' data and making smart decisions to help their users. Offering suggestions and recommendations will certainly help with companies' success.

Keywords—Data analytics, R, Python, Visualization, Decision tree, Random Forest

I. INTRODUCTION

A. Aim and objectives

The research's main objective is to determine a major factor for the movie's success. The research work used mathematical tools such as R, Python, Tableau public, SAS, Apache Spark, Excel, and Rapidminer (a financial data mining tool). The movie data will be analyzed by two statistical frameworks and will show through several analytical techniques. The results will be shared with the current journal population. It will be helpful to apply machine learning algorithms like Random Forest to identify interesting insights in the crowdsourcing community.

B. Selection of dataset

The dataset is obtained from the website Kaggle (**https://www.kaggle.com/rounakbanik/the-movies-dataset?select=movies_metadata.csv**). The dataset includes different variables and features related to the movie.

C. Explanation of Dataset

Our data has more than 5000 records and 28 attributes or variables. The variables that distinguish each record are the columns that build each record. Some other relevant variables are the release date and budget. Each movie that appears on IMDB has a score associated with it. This research study finds that the relationship between the IMDb score and the movies' success, actor popularity and reviews, and movie success.

D. Business case of the project

The work will also build the model for movie success. Machine algorithms will be used to train machine models. Rating websites like IMDB give ratings for movies based on different factors. People decide to watch the movie by the scores given by the website. Movie ratings can determine the success of a movie. The research uses the IMDB movie dataset to find unknown hidden facts from the dataset. For media, it will be useful to make intelligent decisions [1].

E. Selection of environment

Two statistical tools, R and Python, will be used in this research. Both are open-source software for data analysis. The two tools add new libraries with every release. Python is the tool primarily used for data analysis, and R is a general tool. Python and R are both powerful programming languages used for data science.

F. Selection of the technology

Random forest algorithm is one of the most common machine learning algorithms capable of performing both regression and classification tasks. In the forest of many decision trees, the more trees, the more robust and therefore high precision in the model to model multiple decision trees to build the forefront. The reason Random Forest is more likely to find better insights is that movies cost a lot of money no matter what. This analysis is essential to predicting a more accurate success rate.

II. THEORETICAL FEASIBILITY OF THE TECHNOLOGY

Machine Learning

Machine learning is a method by which IT systems use pattern recognition to solve problems. Therefore, LUCC can identify trends and create alternatives with the current methodologies and existing knowledge. Machine learning operates from experiences and concludes these experiences [2].

Types of Machine Learning

There are 3 types of Machine Learning techniques,

1. **Supervised**

Machine learning supervised is the search for algorithms that give rise to generic hypotheses from externally supplied circumstances that predict future circumstances [3].

2. **Unsupervised**

Unsupervised learning is an advanced model methodology where users don't have to track the model. Rather, it helps the model to work itself to discover previously undetected patterns and information [4].

3. **Reinforcement**

The computer is setting up the options. The agent learns how to deal with the unexpected. AI could improve learning by implementing game-like approaches [5].

A statistical model is built for a specific operation or predicting the value. Modeling helps to solve business problems. A model is a simple equation. It has input variables and output variables. The independent variables give us the output value. Explanatory variables are independent variables. The dependent variable is output.
As problems arise, the model must evolve. Modeling is a complicated process. Algorithms are used to implement models. Algorithms can be derived from models. Algorithms can be classified into supervised, unsupervised, and reinforcement learning [6].

A. Random Forest

Classification is predicting the pre-defined class label of the given instance. Classification belongs to supervised machine learning. Classification methods can be classified into two types: binary and multi-class. The machine learning classifiers include logistic regression, KNN, SVM, decision tree, Naïve Bayes, and Random forest (2020). One of the supervised classifications and regression algorithm is Random Forest. The algorithm produces a greater number of trees. Random forests are groups of decision trees. It acts as an ensemble. Decision trees classify the class of an instance. The highest vote from the class will be the final class label. Each decision tree generates its class label. The decision will be made based on the highest number of decision tree classifiers. More trees in the forest improve the accuracy of the model [7].
Figure 25 (Decision Tree)

B. How Random Forest works

Classifiers include both predictors and responses. All predictor variables are used to predict the response variable. Random forests are a collection of decision trees. Random forest techniques select predictors randomly and predict the desired response variable. Random forest considers the average decision tree result. Each decision tree predicts the response variable. The most important variable will be the final decision of the random forest [8].
The decision tree is built with the entire dataset and all predictors whereas a random forest includes multiple decision trees built with a part of the dataset. The decision tree classifier is overfitted. It classifies by building the model with the training instance. Overfitting occurs for predicting the new instance. This algorithm is powerful. The output combines multiple decision trees. Information gain takes place at each node. It predicts the instance where the information gain is high. The search continues until further information is not gained [9].
Decision trees are weak learners. Random forests also train the same weak learners. It uses more than one classifier and combines the outputs of all classifiers. It works similarly to the decision tree. Each decision tree does not provide a complete predictive model. It pulls data for random instances and learns to classify instances. Multiple decision trees are combined to derive the final decision. It improves the classification accuracy and reduces bias [10].

C. Advantages of Random Forest

The algorithm is a powerful ensemble learning (bagging) algorithm. It uses the ensemble learning method. It has many trees and utilizes a subset of data. It uses the combined output of multiple trees. It reduces overfitting as well as variance. It increases accuracy. It can be used for solving both classifications as well as regression problems. It can also work with continuous and categorical variables. It automatically handles the missing values. It follows a rule-based approach. It does not require feature scaling. It handles the nonlinear parameters effectively than curve-based methods. It outperforms the nonlinear independent variables.
It efficiently handles the missing values as well as outliers. It is a stable method. The method is having less impact on noise [11].

D. Limitation of Random Forest

The random forest model has more decision trees. It creates 100 trees in Python. More trees increase the computational power and resources that are available. Besides this, trees or random forests have several other limitations such as: Using any loss function such as cross-entropy for regression or classification. It can be harder to interpret, interpret RFs can be memory-hungry, and it may not be effective in predicting the other three classifications. Our models need longer training [12].

E. Framework

Two statistical tools will be used to collect data, R and Python. Both are open-source statistical tools. The two tools add new libraries with every release. Python is the tool primarily used for data analysis, and R is a general tool. Python and R are both powerful programming languages used for data science [13].

Factors	Why it matters
Cost	Cost association with any work is a burden, fortunately, most of the data analytical tools are open source and free of cost.
Easy to use	Any analytical tool should be user-friendly, and it is very necessary even business point of view. Here in data science most of the tolls are user friendly and easy to use.
Easy to learn	It is a vital point that data analytical tools should be easy to learn. The most popular and famous tools are quite easy to learn.
Community Support	Community support for any IT-based programming language or package is very important, community support is very essential for the development of IT.
Range of libraries	Libraries made it easy for coding work. A data scientist can use and escape a huge coding headache.
Computational speed	Computational speed is a core requirement for data analytics. We need both hardware and tools to be speedy and stable for data statistics because huge hidden calculations are there in data & image processing.
Memory consumption	Saving memories is an important function to create and built an application. Tools that usage fewer memories are always fast, and they become more popular and do well in business too.
Visualization capability	In data science, visualization is an important part. People want to understand from pictorial message most of the time because that is attractive fast and easy to gain knowledge.
System integration	In data science or IT, any system and package that comes with more integration capacity with another system gets more popularity and useful

F. R-code

R is an open-source implementation of the R programming language. The language is used for statistical computing and graphical illustration of results. It is the best tool for performing statistical analysis on the dataset [14].

Advantages

R language has some advantages, such as…

It contains a variety of libraries and tools for specific data operations. The library has various commands for data pre-processing, analysis, visualizations, etc.

R expands packages and libraries, adding more packages and libraries to the R environment. It keeps research and analysis of new methods and techniques.

IDE environment for building applications. Code highlight, compilation, help, visualizations of data, etc.

Interacts with a variety of operating systems, like Unix, Windows, etc.

It offers a web-based dashboard for data analysis and visualization [15].

G. Python

The language was released in the year 1991 and got a good reputation due to its simple nature.

It provides support for various applications such as data science, image processing, websites and back end services, and so on [16].

Advantages

The following are the main advantages of Python such as

It has Simple syntax which can be easily understood

Wider usage of the tools that enable data analysis to clarify queries more quickly through collaboration

It supports third-party modules as well as libraries

It provides support for both procedural as well as to object-oriented language [17]

R code and Python

Both languages are more popular with data scientists. The data from the Stack Overflow Python Survey of 2019 shows that Python is used in 4th place. R is in 16th place.

III. PRACTICAL FEASIBILITY

A.

Cost

Python and R both are open-source platforms of programming language and they are cost-free.

Easy to use

Python is an object-oriented programming language that makes it easier to write large-scale, maintainable, and robust code.

Easy to learn

R is less similar to other common programming languages. Python is simple enough to be a good first programming language to learn.

Community support

R and Python both have large open source communities. Libraries/tools are added continuously to their respective catalogs.

Range of libraries

Both analytical tools are enriched with libraries like:

Python libraries - Numpy, SciPy, Pandas, Keras, SciKit-Learn, TensorFlow, Matplotlib, Seaborn, PyTorch, XGbost etc.

R libraries - tidyverse, dplyr, tidyr, stringr, lubridate, ggplot2, grammar of graphes, ggvis, rgl, htmlwidges, leaflet, dygraphs, DT, diagrammeR, network3D, threeJS etc.

Computational speed

It takes about two minutes and two seconds. The Python code was faster than the R alternative. The R code for this pipeline is 3.3 times faster than the Python code.

Memory Consumption

R's limitations make Python better for data science and machine learning. R is not scalable. Python is a single-threaded language that runs in RAM, which means it is memory-constrained.

Visualization Capacity

Machine learning and advanced analytics are helping humans make sense of large amounts of structured and unstructured data using our natural visual learning ability. Visualization is present here.

Python and R are programming languages that help humans to understand vast datasets.

System Integration

Both Python and R have great packages to solve any kind of problem. There are so many different possibilities to choose from. Python is the obvious choice for these jobs because it is very flexible to integrate other systems.

B. Visualization

Raw data analysis, pre-processing with statistical tools, and data visualization. The initial step loads the dataset imdb movie into the R environment through the reading command.

Figure 1. The command names (data frame) lists all the variables in the dataset

Subset variable

The required variables for analysis will be considered and that will be moved into the “myvars”.

The specified data with specified variables will be a subset with the above code.

The new dataset will have the attributes or columns as in **Figure 2.**

Dimension of Dataset

Now the dataset has 5043 records and 13 attributes

Summary of each attribute in the dataset **Figure 3.**

Figure 4 shows various dimensions

Data pre-processing

Not null values will be removed from the dataset using the command “na.omit”. **Figure 5**. After removing the na values the dimensionality of the dataset gets reduced.

IMDB score and gross

The blot shows the relationship between IMDB score and gross **Figure 6.**

When the IMDB score is increased the number of people watching that movie is also increased until the score 7.5. After the score of 7.5, the number of people watching that movie based on IMDB score is get reduced. **Figure 7**. shows the negative correlation between gross and IMDB after the threshold value 7.5 [18].

IMDB and number of critic reviews

The high IMDB score has a greater number of critic reviews. The number of critic reviews and IMDB score is positively correlated with each other.

Run the plot command **Figure 8.**

The movie released year and number of movies

Run the gplot command **Figure 9.**

From the analysis of movie released year and number of movies released on that, a greater number of movies are released after the year 2000 **Figure 10.**

Movie release year and gross

Run the plot using the command **Figure 11.**

The relationship between movie release year and gross, media industry released more number of movies after the year 2000 as well as got more gross income **Figure 12.**

Correlation analysis

Correlation among the variables such as budget, duration, number of faces in the poster, imdb score, genres, number of critic’s reviews, and Facebook likes for director are analyzed.

In the correlation matrix, positive values represent the high positive correlation between the two variables **Figure 13.**

Negative values indicate a negative correlation between the two analyzed variables. Fewer values indicate that two variables do not correlate that much.

null values will be removed from the dataset using the

Country analysis

While grouping according to the country, **Figure 14.** indicates that more movies are released from the country USA. Least number of movies released from the country

Hong Kong from the top 10 countries

Movie Subset of variable analysis

The relationship between variables such as "num_critic_for_reviews", duration", "facenumber_in_poster","director_facebook_likes","actor_3_facebook_likes","actor_2_facebook_likes","actor_1_facebook_likes","budget","title_year","num_voted_users","imdb_score","gross" are using the scatter plot **Figure 15.**

The following factors are observed in **Figure 16.** such as
Positively correlated variables:
number of votes and number of critics
actor 3 Facebook likes, and actor 2 Facebook likes
number of voted and gross [19]

Correlation plot for the analyzed variables

Figure 17. shows the correlation plot for the analyzed variables.

Model build

The model is developed using the R with random regression **Figure 18.**

PYTHON

The movie data is analyzed in Python. Initially, the dataset will be loaded into an environment similar to R-environment.
The packages such as NumPy and pandas are required to import the dataset into the python environment. The dataset will be read into the python environment.

From the dataset aspect ratio and IMDb link will be removed as well as the dependent variable such as the IMDb score is also dropped.
The column names are displayed while executing the command “column.values. It is illustrated in **Figure 19.**

Data preprocessing is done using Python packages

The imdb score distribution is illustrated in **Figure 20**. using the package matplotlib. The below code illustrates the imdb distribution

The graph **Figure 21** illustrates that on average, most of the movies got a score of 6.5. Very few movies got a score of 9.

Correlation is done with packages Imputer and StandardScalaer. Perform the Impute for removing the outlier in the dataset. From the scipy the classifier used for modeling.

The correlation coefficient of the numeric features is illustrated in **Figure 22. Figure 23.** the correlation matrix shows the relationship between each considered variables. The number of movie critics is highly correlated with the number of movie Facebook likes. Duration is not correlated with any other variables. There is a high correlation between the number of voted users and the number of reviews, gross and movie Facebook likes are also highly related. Model validation is illustrated in **Figure 24.**

IV. DISCUSSION AND CONCLUSION

Two open-source tools like R-code and Python are used to analyze movie data. The purpose of the study is the same on two instruments. The main aim is to conduct predictive and exploratory analyses. For evaluating the pattern of the data set exploratory analysis is used. It includes descriptive information on each of the dataset variables. The study shows that the dataset initially had 5000 records and 28 features (column). Both numerical and categorical data are available. There is also NA and incomplete data in the data collection. It is complex to create a precise model without pre-processing the data. We have prepared the model with a random forest model during this study. Movie quality according to the IMDb rating is graded as standard, good, and poor. The Random Classifier forest constructs a model for movie quality prediction. For both classification and regression, the random forest classifier is used. The two methods for the analysis of data are somewhat similar, both packages and the library are used to carry out different data manipulations. In Python, a continuous variable like IMDb Score has been used for this. The number of movie critics is closely linked to the number of movie Facebook that is common. No other variables are associated with the length. There's a great link between several voters and the number of ratings, as well as gross and movie Facebook likes.

V. RECOMMENDATION

It has been proposed a framework for the combination of multiple decision processes and a random forest algorithm to build recommendable systems. It could be resolved using a single tool either Python or R but for comparison between two tools, it was needed to demonstrate outputs using both tools in this analysis. The Prediction could be done using Linear Regression or Logistic Regression too, however, Random Forest was recommended for better accuracy of the prediction. In addition to recommending or not recommending a customer movie depending on the content and online review (like Facebook likes) of the movie. According to our dataset tests, not only in the practice set but also in the test set, the decision-making threshold pair is pretty good. The model of judgment is lower than the average, with variable precision and probabilistic two-way rough models. In the future, this will be solved using memory-based approaches.

REFERENCES

[1] Agresti, A., 2003. Categorical data analysis (Vol. 482). John Wiley & Sons.

[2] Ibm.com. 2020. *What Is Machine Learning?*. [online] Available at: <https://www.ibm.com/cloud/learn/machine-learning> [Accessed 27 December 2020].

[3] Medium. 2020. *A Brief Introduction To Supervised Learning*. [online] Available at: <https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590> [Accessed 30 December 2020].

[4] Brownlee, J., 2020. *Supervised And Unsupervised Machine Learning Algorithms*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/> [Accessed 30 December 2020].

[5] AltexSoft. 2020. *Reinforcement Learning Explained: Overview, Comparisons And Applications In Business*. [online] Available at: <https://www.altexsoft.com/blog/datascience/reinforcement-learning-explained-overview-comparisons-and-applications-in-business/#:~:text=Reinforcement%20learning%20(RL)%20is%20a,the%20environment%20after%20each%20act.> [Accessed 27 December 2020].

[6] Insights, S., 2020. *Machine Learning: What It Is And Why It Matters*. [online] Sas.com. Available at: <https://www.sas.com/en_in/insights/analytics/machine-learning.html> [Accessed 27 December 2020].

[7] Medium. 2020. *Understanding Random Forest*. [online] Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Accessed 30 December 2020].

[8] Medium. 2020. *Random Forest Simple Explanation*. [online] Available at: <https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d> [Accessed 30 December 2020].

[9] Built In. 2020. *A Complete Guide To The Random Forest Algorithm*. [online] Available at: <https://builtin.com/data-science/random-forest-algorithm> [Accessed 30 December 2020].

[10] Corporate Finance Institute. 2020. *Random Forest - Overview, Modeling Predictions, Advantages*. [online] Available at: <https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/> [Accessed 30 December 2020].

[11] Medium. 2020. *Why Random Forests Outperform Decision Trees*. [online] Available at: <https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5> [Accessed 30 December 2020].

[12] HolyPython.com. 2020. *Random Forest Pros & Cons - Holypython.Com*. [online] Available at: <https://holypython.com/rf/random-forest-pros-cons/> [Accessed 30 December 2020].

[13] TechRepublic. 2020. *The 10 Most Popular Machine Learning Frameworks Used By Data Scientists*. [online] Available at: <https://www.techrepublic.com/article/the-10-most-popular-machine-learning-frameworks-used-by-data-scientists/> [Accessed 30 December 2020].

[14] Tutorialspoint.com. 2020. *R Tutorial - Tutorialspoint*. [online] Available at: <https://www.tutorialspoint.com/r/index.htm> [Accessed 30 December 2020].

[15] DataFlair. 2020. *Pros And Cons Of R Programming Language - Unveil The Essential Aspects! - Dataflair*. [online] Available at: <https://data-flair.training/blogs/pros-and-cons-of-r-programming-language/> [Accessed 30 December 2020].

[16] Programiz.com. 2020. *Learn Python Programming*. [online] Available at: <https://www.programiz.com/python-programming#:~:text=Python%20is%20a%20powerful%20general,learn%20to%20program%20for%20beginners.> [Accessed 30 December 2020].

[17] upGrad blog. 2020. *Top 10 Reasons Why Python Is So Popular With Developers In 2020 | Upgrad Blog*. [online] Available at: <https://www.upgrad.com/blog/reasons-why-python-popular-with-developers/> [Accessed 30 December 2020].

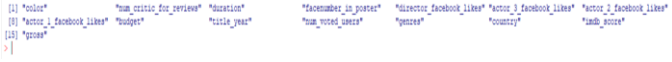
[18] Oghina A, Breuss M, Tsagkias M, De Rijke M. Predicting imdb movie ratings using social media. InEuropean Conference on Information Retrieval 2012 Apr 1 (pp. 503-507). Springer, Berlin, Heidelberg.

[19] Doshi P, Zadrozny W. Movie genre detection using topological data analysis. InInternational Conference on Statistical Language and Speech Processing 2018 Oct 15 (pp. 117-128). Springer, Cham.

APPENDIX

```
> movies <- read.csv('movie_metadata.csv',header=T,stringsAsFactors = F)
> read(movies)
Error in read(movies) : could not find function "read"
> names(movies)
 [1] "color"                 "director_name"             "num_critic_for_reviews"
 [7] "actor_2_name"          "actor_1_facebook_likes"    "gross"
[13] "num_voted_users"       "cast_total_facebook_likes" "actor_3_name"
[19] "num_user_for_reviews"  "language"                  "country"
[25] "actor_2_facebook_likes" "imdb_score"               "aspect_ratio"
>
```

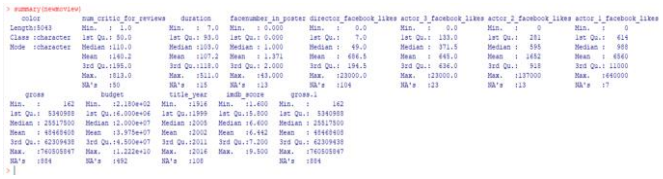
Listing all variables from the dataset - Figure 1.



New dataset with the attributes - Figure 2.

```
> dim(newmoview)
[1] 5043   15
>
```

Summary of each attribute in the dataset -Figure 3.



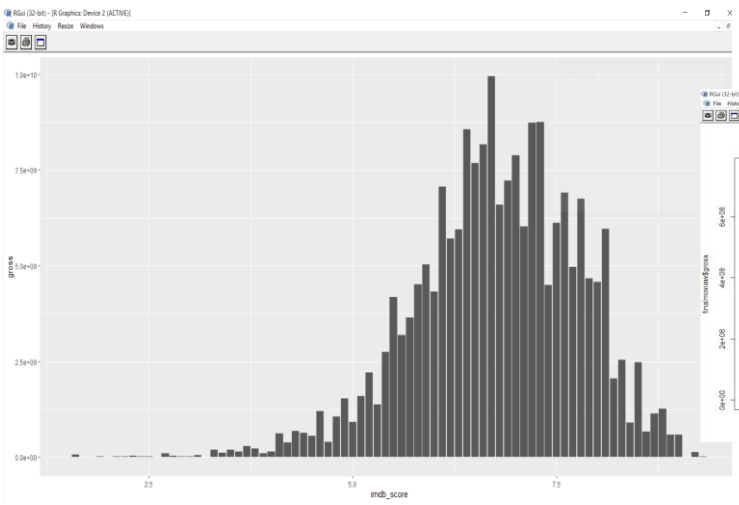
Various dimension of the dataset - Figure 4

```
> finalmoview<- na.omit(newmoview)
>
 > dim(finalmoview)
 [1] 3873   15
 >
```

After removing null values from the dataset - Figure 5.

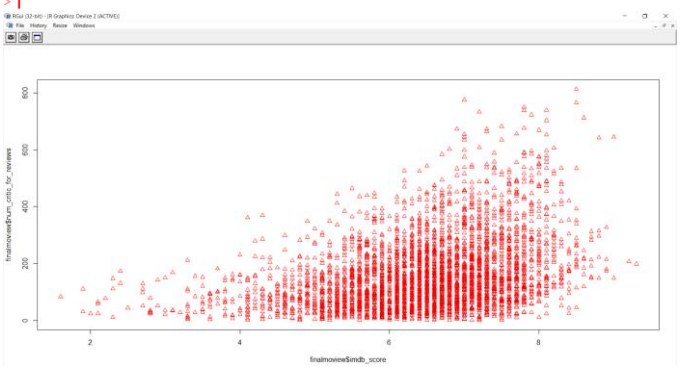
```
> library(ggplot2)
> ggplot(data = finalmoview, mapping = aes(x = imdb_score, y = gross)) +
+   geom_col()
>
```

Relationship between IMDB score and gross - Figure 6.



IMDB score is get reduced - Figure 7.

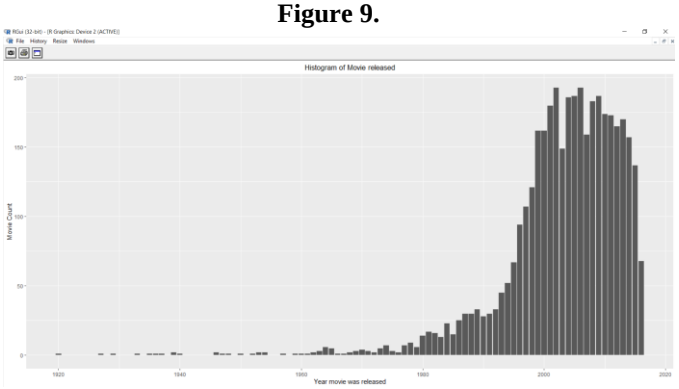
```
> plot(finalmoview$imdb_score, finalmoview$num_critic_for_reviews, pch=2, col='red')
>
```



IMDB and number of critic reviews- Figure 8.

```
> ggplot(finalmoview, aes(title_year)) +
+   geom_bar() +
+   labs(x = "Year movie was released", y = "Movie Count", title = "Histogram of Movie released") +
+   theme(plot.title = element_text(hjust = 0.5))
>
```

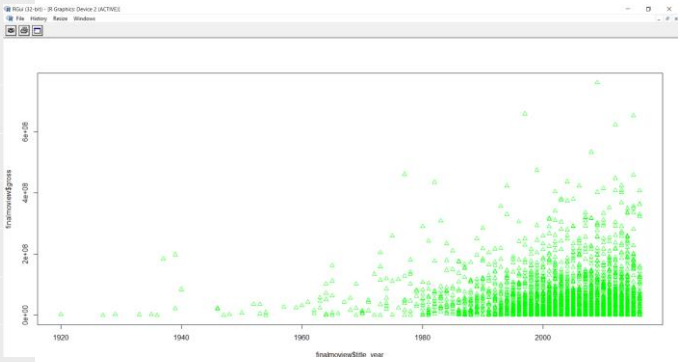
The movie released year and number of movies- Figure 9.



A greater number of movies are released after the year 2000 - Figure 10.

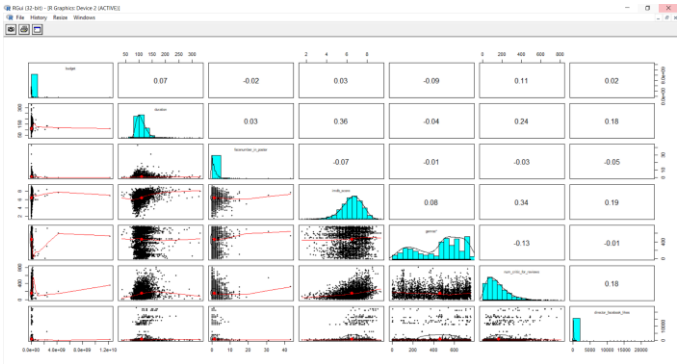
```
>
> plot(finalmoview$title_year, finalmoview$gross, pch=2, col='green')
>
```

Movie release year and gross - Figure 11.



Movie release year and gross - Figure 12.

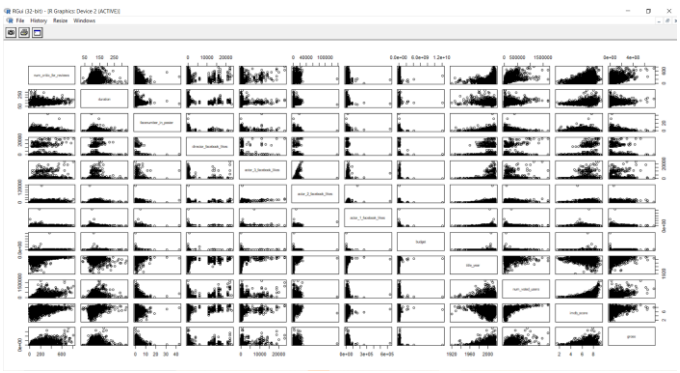
```
> library(psych)
Attaching package: 'psych'
The following objects are masked from 'package:ggplot2':
    %+%, alpha
> pairs.panels(finalmoview[c('budget','duration','facenumber_in_poster','imdb_score','genres','num_critic_for_reviews','director_facebook_likes')])
>
```



Correlation analysis - Figure 13.

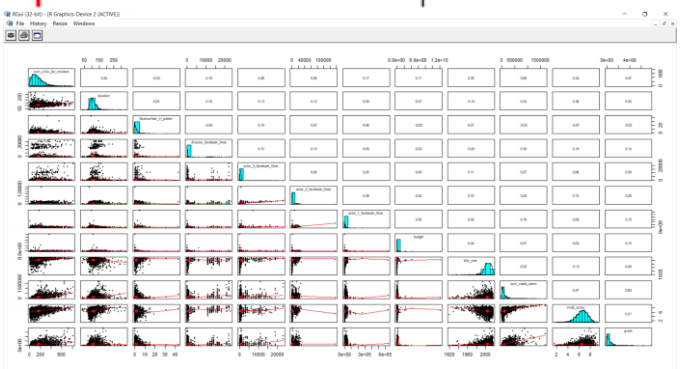
```
> top_10_country <- finalmoview %>%
+   group_by(country) %>%
+   summarise(count = n()) %>%
+   top_n(10) %>%
+   arrange(desc(count))
`summarise()` ungrouping output (override with `.groups` argument)
Selecting by count
> top_10_country
# A tibble: 10 x 2
   country   count
   <chr>     <int>
 1 USA        3062
 2 UK          324
 3 France      105
 4 Germany      82
 5 Canada       63
 6 Australia    41
 7 Spain        22
 8 Japan        17
 9 China        15
10 Hong Kong    13
```

Country analysis - Figure 14.



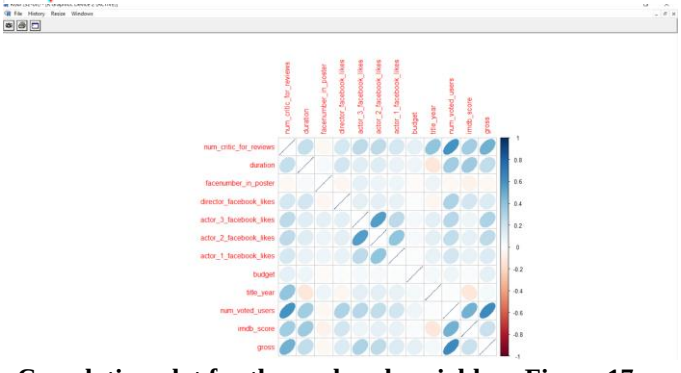
Movie Subset of variable analysis - Figure 15.

```
> pairs.panels(movies_num)
>
```



Movie Subset of variable analysis - Figure 16.

```
> corrplot(M, method = "ellipse")
>
```



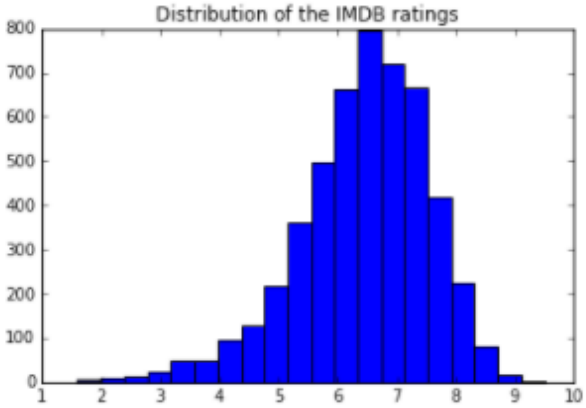
Correlation plot for the analyzed variables - Figure 17.

```
Call:
 randomForest(formula = as.factor(quality) ~ duration + director_facebook_likes +      adj_budg + actor_1_facebook_likes + actor_2_facebook_likes +      actor_
               Type of random forest: classification
                     Number of trees: 9560
No. of variables tried at each split: 2

        OOB estimate of  error rate: 39.58%
Confusion matrix:
       bad good normal class.error
bad     82    6    144   0.6746032
good     9   97    163   0.6394052
normal  52   62    517   0.1806656
```

Model build - Figure 18.

```
['color' 'director_name' 'num_critic_for_reviews' 'duration'
 'director_facebook_likes' 'actor_3_facebook_likes' 'actor_2_name'
 'actor_1_facebook_likes' 'gross' 'genres' 'actor_1_name' 'movie_title'
 'num_voted_users' 'cast_total_facebook_likes' 'actor_3_name'
 'facenumber_in_poster' 'plot_keywords' 'movie_imdb_link'
 'num_user_for_reviews' 'language' 'country' 'content_rating' 'budget'
 'title_year' 'actor_2_facebook_likes' 'imdb_score' 'aspect_ratio'
 'movie_facebook_likes']
```

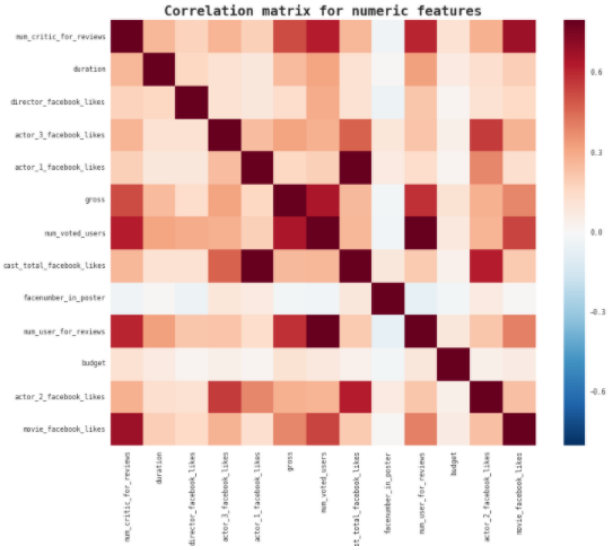
The column names are displayed - Figure 19.



The imdb score distribution is illustrated - Figure 20.

```
[('num_voted_users', (0.41096520027034722, 8.5614202349812934e-205)), ('num_critic_for_revie
ws', (0.29971283201076732, 3.4999174979932841e-105)), ('num_user_for_reviews', (0.2897869244
3885773, 3.7438372081606225e-98)), ('duration', (0.26107064856577861, 2.2913365138738415e-7
9)), ('movie_facebook_likes', (0.24704851902725028, 5.26928820566599e-71)), ('gross', (0.176
36050188406846, 1.6379676123144198e-36)), ('director_facebook_likes', (0.1624675957859003 4,
3.5875676939989255e-31)), ('cast_total_facebook_likes', (0.085787347548007348, 1.04847362660
89975e-09)), ('actor_2_facebook_likes', (0.083550727133696392, 2.812236914134845e-09)), ('ac
tor_1_facebook_likes', (0.075866760509358949, 6.888788561799105e-08)), ('actor_3_facebook_li
kes', (0.052795979345185 47, 0.00017616957697643 63)), ('budget', (0.027357207396971669, 0.052
06100536210327 1)), ('facenumber_in_poster', (-0.062210931032976265, 9.822719464157489 1e-0
6))]
```

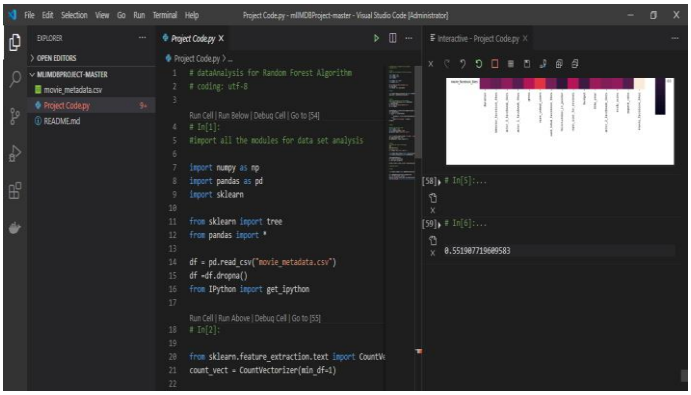
Illustrates that on average, most of the movies got a score of 6.5. - Figure 21.



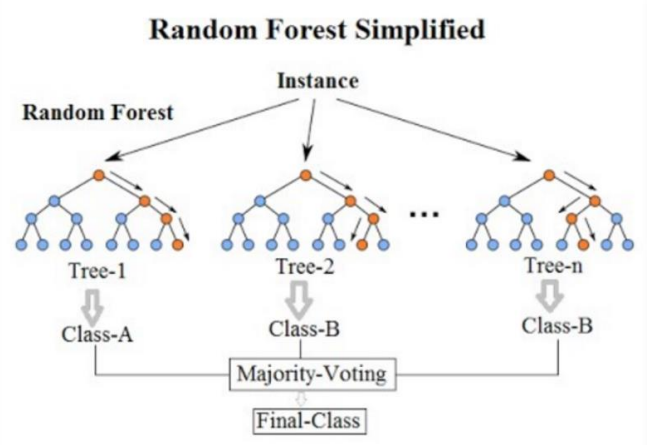
The correlation coefficient of the numeric features Figure 22.

```
Rank: 1
Mean validation score: 0.293 (std: 0.087)
Parameters: {'min_samples_split': 5, 'n_estimators': 800, 'max_depth': 25}

Rank: 2
Mean validation score: 0.292 (std: 0.088)
Parameters: {'min_samples_split': 5, 'n_estimators': 1200, 'max_depth': 25}

Rank: 3
Mean validation score: 0.292 (std: 0.092)
Parameters: {'min_samples_split': 2, 'n_estimators': 800, 'max_depth': 25}
```

The correlation matrix shows the relationship between each considered variables. Figure 23.



The effects of Facebook likes on movie dataset Figure 24



Decision tree - Figure 25

R CODE

```
getwd()
movies <- read.csv('movie_metadata.csv',header=T,stringsAsFactors = F)
read(movies)
names(movies)
movies.req$color[is.na(movies.req$color) & movies.req$title_year>2000] <- "Color
movies.req$color[is.na(movies.req$color)] <- "Black and White"
myvars <- c("color", "num_critic_for_reviews", "duration","facenumber_in_poster","director_facebook_likes","actor_3_facebook_likes","actor_2_facebook_likes", "actor_1_facebook_likes","gross","budget","title_year", "imdb_score","gross")
newmoview <- movies[myvars]
names(newmoview)
dim(movie)
dim(newmoview)
summary(newmovie)
summary(newmoview)
finalmoview<- na.omit(newmoview)
dim(finalmoview)
myvars <- c("color", "num_critic_for_reviews", "duration","facenumber_in_poster","director_facebook_likes","actor_3_facebook_likes","actor_2_facebook_likes", "actor_1_facebook_likes","gross","budget","title_year","num_voted_users","genres","country"
"imdb_score","gross")
myvars1 <- c("color", "num_critic_for_reviews", "duration","facenumber_in_poster","director_facebook_likes","actor_3_facebook_likes",
"actor_2_facebook_likes",
"actor_1_facebook_likes","gross","budget","title_year","num_voted_users","genres","country",
"imdb_score","gross")
newmoview <- movies[myvars1]
names(newmoview)
myvars1 <- c("color", "num_critic_for_reviews", "duration","facenumber_in_poster","director_facebook_likes","actor_3_facebook_likes",
```

```
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","title_year","num_voted
_users","genres","country",
"imdb_score","gross")
newmoview <- movies[myvars1]
names(newmoview)
dim(newmoview)
finalmoview<- na.omit(newmoview)
dim(finalmoview)
ggplot(data = movie, mapping = aes(x = imdb_score, y =
gross)) +
  geom_col()
ggplot(data = finalmovie, mapping = aes(x = imdb_score, y
= gross)) +
  geom_col()
ggplot(data = finalmoview, mapping = aes(x = imdb_score,
y = gross)) +
  geom_col()
library(ggplot2)
ggplot(data = finalmoview, mapping = aes(x = imdb_score,
y = gross)) +
  geom_col()
plot(finalmoview$imdb_score,
"finalmoview$num_critic_for_reviews", pch=2, col='red')
plot(finalmoview$imdb_score,
finalmoview$num_critic_for_reviews, pch=2, col='red')
ggplot(finalmoview, aes(title_year)) +
  geom_bar() +
  labs(x = "Year movie was released", y = "Movie Count",
title = "Histogram of Movie released") +
  theme(plot.title = element_text(hjust = 0.5))
plot(movie$title_year, movie$gross, pch=2, col='green')
plot(finalmoview$title_year, finalmoview$gross, pch=2,
col='green')
pairs.panels(finalmoview[c('budget','duration','facenumber_i
n_poster','imdb_score','genres','num_critic_for_reviews','dire
ctor_facebook_likes')])
library(psych)
pairs.panels(finalmoview[c('budget','duration','facenumber_i
n_poster','imdb_score','genres','num_critic_for_reviews','dire
ctor_facebook_likes')])
top_10_country <- finalmoview %>%
  group_by(country) %>%
  summarise(count = n()) %>%
  top_n(10) %>%
  arrange(desc(count))
library(dplyr)
top_10_country <- finalmoview %>%
  group_by(country) %>%
  summarise(count = n()) %>%
  top_n(10) %>%
  arrange(desc(count))
top_10_country <- finalmoview %>%
  group_by(country) %>%
  summarise(count = n()) %>%
  top_n(10) %>%
  arrange(desc(count))
top_10_country
movies_num <- movie %>%
  select("color",
"duration","facenumber_in_poster","director_facebook_like
s","actor_3_facebook_likes",
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","title_year","num_voted
_users","genres","country",
"imdb_score","gross")
movies_num
movies_num <- finalmoview %>%
  select("color",
"duration","facenumber_in_poster","director_facebook_like
s","actor_3_facebook_likes",
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","title_year","num_voted
_users","genres","country",
"imdb_score","gross")
movies_num
movies_num <- finalmoview %>%
  select("color",
"duration","facenumber_in_poster","director_facebook_like
s","actor_3_facebook_likes",
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","title_year","num_voted
_users","genres","country",
"imdb_score","gross")
movies_num
pairs(movies_num)
movies_num <- finalmoview %>%
  select(
"duration","facenumber_in_poster","director_facebook_like
s","actor_3_facebook_likes",
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","title_year","num_voted
_users",
"imdb_score","gross")
movies_num
pairs(movies_num)
pairs.panels(movies_num)
corrplot(movies_num, method = "ellipse")
corrplot(movies_num, method = "ellipse")
library(corrplot)
corrplot(movies_num, method = "ellipse")
M <- cor(movies_num, use = "complete.obs") #
complete.obs removes all the NA's
M
corrplot(M, method = "ellipse")M <- cor(movies_num, use
= "complete.obs") # complete.obs removes all the NA's
M
corrplot(M, method = "ellipse")
model=lm(movies_num$gross~movies_num$num_critic_fo
r_reviews+movies_num$duration+movies_num$director_fa
cebook_likes+movies_num$actor_3_facebook_likes+movie
s_num$actor_2_facebook_likes+
movies_num$actor_1_facebook_likes+movies_num$budget
+movies_num$num_voted_users+movies_num$imdb_score
)
summary(model)
movies_full <- finalmoview %>%
  select(
"duration","facenumber_in_poster","director_facebook_like
s","actor_3_facebook_likes",
```

```
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","num_voted_users",
"imdb_score","gross")
movies_full
indx = sample(1:nrow(movies_full),
as.integer(0.9*nrow(movies_full)))
indx # ramdomize rows, save 90% of data into index
indx = sample(1:nrow(movies_full),
as.integer(0.9*nrow(movies_full)))
indx # ramdomize rows, save 90% of data into index
movie_train = movies_full[indx,]
movie_test = movies_full[-indx,]
movies_full <- finalmoview %>%
  select( "num_critic_for_reviews",
"duration","facenumber_in_poster","director_facebook_like
s","actor_3_facebook_likes",
"actor_2_facebook_likes",
"actor_1_facebook_likes","budget","num_voted_users",
"imdb_score","gross")
movies_full
indx = sample(1:nrow(movies_full),
as.integer(0.9*nrow(movies_full)))
indx # ramdomize rows, save 90% of data into index
indx = sample(1:nrow(movies_full),
as.integer(0.9*nrow(movies_full)))
indx # ramdomize rows, save 90% of data into index
movie_train = movies_full[indx,]
movie_test = movies_full[-indx,]
pr<-predict.lm(model,newdata =
data.frame(movie_test),interval = 'confidence')
pr
MAE <- function(actual, predicted) {
mean(abs(actual - predicted))
}
MAE(pr, movie_test$gross)
plot(model)
model2=lm(movies_num$gross~movies_num$num_critic_f
or_reviews+movies_num$num_voted_users)
movies_full2 <- finalmoview %>%
  select( "num_critic_for_reviews", "num_voted_users",
"gross")
movies_full2
indx = sample(1:nrow(movies_full2),
as.integer(0.9*nrow(movies_full2)))
indx # ramdomize rows, save 90% of data into index
indx = sample(1:nrow(movies_full2),
as.integer(0.9*nrow(movies_full2)))
indx # ramdomize rows, save 90% of data into index
movie_train = movies_full2[indx,]
movie_test = movies_full2[-indx,]
pr<-predict.lm(model,newdata =
data.frame(movie_test),interval = 'confidence')
pr
MAE <- function(actual, predicted) {
mean(abs(actual - predicted))
}
MAE(pr, movie_test$gross)
plot(model2)
library(car)
residualPlots(model)
residualPlots(model2)
shapiro.test(model2$residuals)
shapiro.test(model$residuals)
ncvTest(model2)
ncvTest(model)
library(QuantPsyc)
install.packages("psych")
install.packages("boot")
library(QuantPsyc)
 install.packages("QuantPsyc")
library(QuantPsyc)
lm.beta(model)
lm.beta(model2)
library(rpart)
library(rpart.plot)
install.packages("rpart.plot")
```

PYTHON CODE

```
# dataAnalysis for Random Forest Algorithm
# coding: utf-8

# In[1]:
#import all the modules for data set analysis

import numpy as np
import pandas as pd
import sklearn

from sklearn import tree
from pandas import *

df = pd.read_csv("movie_metadata.csv")
df =df.dropna()
from IPython import get_ipython

# In[2]:

from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer(min_df=1)


# In[3]:
#import the datasetcsv to the algorithm
from sklearn.preprocessing import LabelEncoder
X = pd.DataFrame()
df = pd.read_csv("movie_metadata.csv")
df = df.dropna()


columnsToEncode =
list(df.select_dtypes(include=['category','object']))
le = LabelEncoder()
for feature in columnsToEncode:
    try:
        df[feature] = le.fit_transform(df[feature])
    except:
        print('Error encoding ' + feature)
```

```
df.head()


# In[4]:

#drawing the math plot chart
import seaborn as sns
import matplotlib.pyplot as plt
get_ipython().magic('matplotlib inline')


corr = df.select_dtypes(include = ['float64', 'int64']).iloc[:,
1:].corr()
plt.figure(figsize=(15, 15))
sns.heatmap(corr, vmax=1, square=True)



# In[5]:
#calculate the score of the dataset
X=df
y=X['imdb_score']
#y.apply(np.round)
X = X.drop(['imdb_score'], axis = 1)

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()
X = scaler.fit_transform(X)
y = np.array(y).astype(int)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=.3, random_state =190)

# ## Random Forest

# In[6]:

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n_estimators=28)
clf = clf.fit(X_train, y_train)
print("Accuracy of RandomForestClassifier:",
accuracy_score(y_test,clf.predict(X_test)))
```