University of Detroit Mercy

College of Engineering and Science

Computer Science and Software Engineering

CSSE-5930 graduate Design Project

Prediction of Movie Ratings Using IMDB Dataset

Prepared by

Karthik Reddy Middela
Sunil Kondreddi
Lei Wang

Project Advisor

Dr. Kevin Daimi

April 2018

Index:

Abstract:

In this paper, techniques from machine learning are used to predict IMDB movie ratings based on attributes from the IMDB movie rating's. This paper provides a quite efficient approach to predict ratings from the IMDB Movie Dataset. We have tried to unveil the important factors influencing the ratings of IMDB Movie Data. By performing an exploratory analysis of the data and observe some interesting phenomenon, which also helps us improve our prediction strategy. Our results finally show the achievement of a good prediction of score on this dataset. This paper is a final report for a class on applied machine learning.

Key Words: Prediction, movie dataset, linear regression, Bayesian networks, artificial neural network.

1.Introduction:

Nowadays, Movies have become a part of our life. Every year hundreds of movies are produced and released. It includes both good movies and poor ones. Movies are enjoyable and relaxing and the biggest online movie database, provides us a new way to evaluate the goodness of a movie based on users' review's. It is still hard to evaluate the greatness of the movie from the IMDB score for two reasons. First, the IMDB score is based on users review. It will not be reliable until a few months after the movie release. The second reason is that the IMDB score can be misinterpreted by the general audience, as other platforms such as Netflix provides ratings that are higher than IMDB. In this paper, a work using information from IMDB and social media to predict movies rating score is presented. The work would be valuable for movie lovers to decide whether they will watch the movie once it released for movie lovers. Also, for the movie industry, it could be used to evaluate what kind of movie will likely to be liked by the average audience. The dataset used in our paper comes from IMDB Movie Dataset on Kaggle. It contains 28 variables for 14,726 movies spanning across 100 years in 66 countries. [1] [3] [5].

2. Tools Used:

2.1

R is a programming language and software environment for statistical analysis, graphics representation and reporting. In R, can read data from files stored outside the R environment. We can also write data into files which will be stored and accessed by the operating system. R can read and write into various file formats like csv, excel, xml etc. in this project csv file is used. For loading the csv file in to R use the following command. [10]

```
movie <- read.csv ("C:/Users/karht/Videos/grad project papers/movie.csv")

View(movie)
```

before doing this, install csv packages for importing the dataset in to R. package can be installed by typing the following command

```
library(readr) # installs the packages.
```

- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- R has an effective data handling and storage facility,
- R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- R provides a large, coherent and integrated collection of tools for data analysis.
- R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

2.2. R Studio:

R Studio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. [10].

2.2.1. The usual R Studio screen has four windows:

1. Console.
2. Workspace and history.
3. Files, plots, packages and help.
4. The R script and data view.

The R script is where you keep a record of your work. Packages can also be installed from the package tab. The package tab shows the list of add-ons that are included in the installation of R Studio. There is another way for installing the packages by clicking install packages (or) by typing the script.

library(rgl) # this will automatically check the rgl package.

These rgl package is useful for the 3D images. Basically, these packages don't come with original R install.

Workspace tab:

The workspace tab stores any object, value, function or anything you create during your R session.

History tab:

The history tab keeps a record of all previous commands. It helps when testing and running processes. Here you can either save the whole list or you can select the commands you want and send them to an R script to keep track of your work.

3. Methods Used:

Linear regression is one of the most common supervised learning approaches for prediction. In linear regression the goal is to find a linear relationship between inputs and output values you are trying to predict. Artificial Neural Network is also used for prediction, such as predicting stocks, predicting currency exchange rates and analysis of data. It can model a system with an unknown input-output relation. This network with no knowledge can be trained with set of paired input-output data to get desired outputs for known inputs. It adapts response to changes in surrounding environment. The Bayesian Method will work with limited data. Causality provides a formal language for conceptualizing, Necessary to predict the effect of interventions.

In this project three types of prediction methods are implemented. They are linear regression, artificial neural network and the Bayesian method. All these methods are used for prediction in R for using these methods install the packages that are mentioned below.

Linear regression can be used by installing the package “alr3”

```
install. Packages(“alr3”) # installs the package

library(alr3) # loads the package in to R
```

Artificial neural network can be used by installing the package “neuralnet”

```
install. Packages(“neuralnet”) # installs the packages

library(neuralnet) # loads the packages in o R
```

And for using the Bayesian method install package “bnlearn”

```
install. Packages(“bnlearn”) # installs the packages

library(bnlearn) # loads the package in to R
```

For plotting the results install package “Plot3D”

```
Install. Packages(“plot3D”) # installs the package

Library(plot3D) # loads the packages in to R
```

“corrplot” package is used for measuring the linear dependence between two variables

Install. Packages(“corrplot”) # install the package

Library(corrplot) # loads the packages in to R

The first step for calling the methods in to R is to install the package and then for loading the methods in to R write a script for calling linear regression by library(“alr3”). Artificial neural network can be called by writing library(“neuralnet”). And for calling the Bayesian method write library(“bnlearn”).

3.1. Linear Regression: Regression is a part of machine learning. To overcome the outliers, we use regression. Regression analysis is a predictive modelling technique. It estimates the relationship between a dependent(target) and an independent variable(predictor). It is widely used statistical tool; the purpose of this method is to create a relationship between two variables. Regression basically divided into three types. Linear regression is about relating the two variables in an equation format (Mathematically). In a graph, linear relationship represents as a straight line: when represented mathematically. We use the general mathematical equation y = mx + c where y is the response variable, x is a predictor variable and m and c are constants and which are also called as coefficients. Since we are doing prediction we use only Linear regression.

Straight line regression analysis involves a single predictor variable and response variable

$$Y = b + wx$$

- The variance of y is constant
- B and W are regression coefficients
  - B: y-intercept
  - W: the slope of the line
  - Need of estimating the regression coefficients

For instance, Maxus sale corporation is in the business of selling laptops. They realized the advantages of forecasting very early in their business. They also realized to perform effective forecast, they need to keep track of their current and past sales numbers. Currently, there are in the process of forecasting their sales number for each quarter of coming year. To do this, they have pulled up their sales number for the past 12 quarters. These number are as follows

Quarter	Sales
1	600
2	1550
3	1500
4	1500
5	2400
6	3100
7	2600
8	2900
9	3800
10	4500
11	4000
12	4900

Use the least square methods to find the forecast of the next 4 quarters. Also find out standard error of estimate.

Y = a +bx

We should find the residuals by the formula

$(y_1-y_1^1)$^2 + $(y_2-y_2^1)$^2 + ………………………... $(y_{12}-y_{12}^1)$^2

A = y-bx̄

B= ∑x̄ y(bar) -nx̄y(bar)   x̄ = ∑x /n = 6.5 ; y(bar) = ∑y /n = 2799.7

  = ∑x^2- nx^2

B= (268200-12*6.5*2799.7)/ (650-12*(6.5) ^2 = 359.6

$A = y\text{-}b\bar{x} = 2799.7\text{-}359.6 * 6.5 = 441.77$

$Y = 441.77 + 359.6x$

Now place x = 13, 14, 15 and 16

$Y_{13} = 5116.78$; $Y_{14} = 5476.17$; $Y_{15} = 5835.77$; $Y_{16} = 6195.37$

3.2. Artificial Neural Network:

This algorithm performs statistical modelling, "provide a new alternative to logistic regression". The Artificial Neural Network is used for predictive modelling. Artificial Neural Networks are patterned based on the structure of the human brain. Artificial Neural Network was developed based on the analysis of human brain. However, in the late 1980s ANN was developed, basically Neural Nets purpose is to find the mathematical relationships between the inputs.

Artificial Neural Network is defined as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."

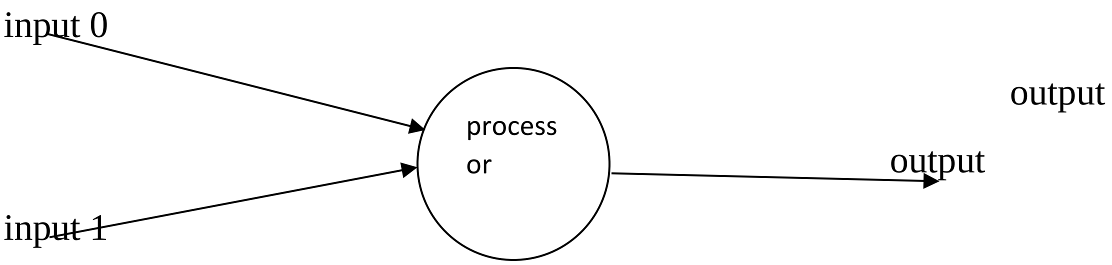
Neural nets are data driven machine learning algorithms. Like smoothing algorithms, they learn patterns from data. Like regression, they were designed for capturing a relationship between inputs and output variable, using hidden data. Neural Nets are also used for forecasting time series (numerical/binary).

Advantages of using Artificial Neural Network are first "Neural Network require less formal statistical training to develop." Based on the certain data set and neural network software, working of artificial neural networks can be developed by newcomers to neurocomputing and that is done with in less period.

Second, Artificial Neural Networks are trained based on different algorithms

Third, Artificial Neural Network can detect relationship between predictor and variables and finally the Artificial Neural Network can detect non-linear relationship between dependant and independent variables.

For instance,



Let assume input layer with two inputs and call them as x1 and x2

Input 0: x1 = 12
Input 1: x2 = 4

And weighted inputs are inputs from input layer which are weighted in the hidden layer i.e. multiplied by some value (often a number between -1 and 1). When creating a perceptron, we'll typically begin by assigning random weights. Here, let's give the inputs the following weights:

Weight 0: 0.5
Weight 1: -1

We take each input and multiply it by its weight.

Input 0 * Weight 0 ⇒ 12 * 0.5 = 6

Input 1 * Weight 1 ⇒ 4 * -1 = -4

The output layer is the Output = sign(sum) ⇒ sign (2) ⇒ +1

The purpose of using Artificial Neural Network is for Massive parallelism, distributed representation, learning ability, generalization ability, fault tolerance. There are various application using Artificial Neural Network, it is used for pattern classification, clustering/categorization, function approximation, prediction/forecasting, optimization, content-addressable memory, control.

3.3. Bayesian Method:

In 1983, Dr. Judea Pearl created Bayesian Network. Bayesian Network is the combination of Artificial Intelligence and Statistics. Bayesian Networks "function by creating a probabilistic model that can be used to query possible outcomes from input data." Bayesian Network is used in many areas such as predictive modelling, pattern recognition, classification and Regression. Bayesian Network deals with the probabilistic relationship between the random variables. In general, random variable is defined as in statistics the items involved are vary in unexplained manner and that are known as random variables. Based on the analysis of input data, a Bayesian network assigns probability factors to various results. Bayesian Network uses training data. When the data is trained, Bayesian network can be queried to make predictions regarding new data.

There are several advantages of using Bayesian Network compared to other machine learning algorithms such as Artificial Neural Networks, Decision Tree, SVM etc. The first advantage is it handles the missing values. In general, only few algorithms handle missing data. In that scenario of missing data few machine learning algorithms will eliminate or extrapolate the data. Second advantage of Bayesian Network can be used for query. For instance, "an SVM or ANN is typically trained to predict a specific outcome given specific input." The important note on Bayesian Network is "random variables do not have fixed roles of input or output."

The main advantages of using Bayesian network is when probability of parent is given, we can easily find probability of children and other main advantage is the Bayesian Network can cope with missing data problem.

Bayesian Network can predict the future based on past observations which shows way to itself naturally to applications requiring predictive analytics.

For instance, approximately 1% of women aged 40-50 have breast cancer. A woman with breast cancer has a 90% chance of a positive test from a mammogram, white woman without has a 10% chance of a false positive result. What is the probability a woman has breast cancer given that she had a positive test?

Let B = "The woman has breast cancer".

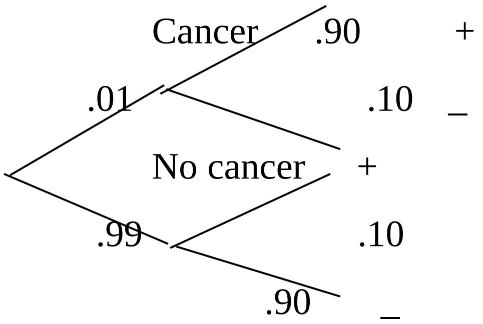
And A = "a positive test".

P (cancer/+) =?

P (A/B) = [P (B/A). P(A)]/P(B)

P (cancer/+) = P(+/cancer). P(cancer)/P (+)

Tree diagram:



P (cancer/+) = [(.90) (.01)]/ [(.01) (.90) +(.99) (.10)]

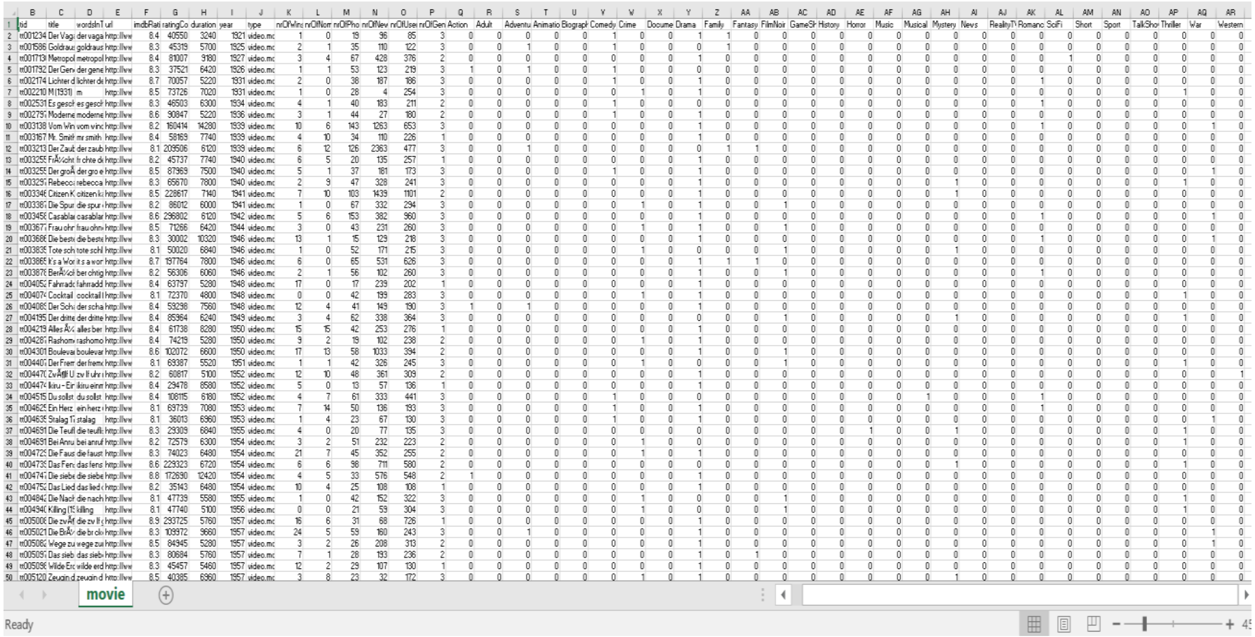
= 9/108 = .08333333 = 8.3%

This shows probability of cancer given the test result is positive is 8.3%.

4. Dataset Collection and Preprocessing:

4.1 Dataset collection:

The dataset is collected from Kaggle.com. the initial dataset contained of around 15,000 movies and 48 different variables, which covers all different type of movies which includes authors name and also the title names of the movie. The below figure shows the IMDB movie dataset.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
1	tid	title	wordsInT	url	imdbRati	ratingCo	duration	year	type	nrOfWins	nrOfNom	nrOfPho	nrOfNew	nrOfUse	nrOfGen	Action	Adult	Adventu	Animatio	Biograph	Comedy	Crime	Docume	Drama	Family	Fantasy	FilmNoir	GameSh	History	Horror	Music	Musical	Mystery	News	RealityT	Romanc	SciFi	Short	Sport	TalkSho	Thriller	War	Western
2	tt001234	Der Vag	der vaga	http://ww	8.4	40550	3240	1921	video.mc	1	0	19	96	85	3	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	tt001586	Goldraus	goldraus	http://ww	8.3	45319	5700	1925	video.mc	2	1	35	110	122	3	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	tt001713	Metropol	metropol	http://ww	8.4	81007	9180	1927	video.mc	3	4	67	428	376	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
5	tt001792	Der Gen	der gene	http://ww	8.3	37521	6420	1926	video.mc	1	1	53	123	219	3	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	tt002174	Lichter d	lichter de	http://ww	8.7	70057	5220	1931	video.mc	2	0	38	187	186	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	tt002210	M (1931)	m	http://ww	8.5	73726	7020	1931	video.mc	1	0	28	4	254	3	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
8	tt002531	Es gesch	es gesch	http://ww	8.3	46503	6300	1934	video.mc	4	1	40	183	211	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	tt002797	Moderne	moderne	http://ww	8.6	90847	5220	1936	video.mc	3	1	44	27	180	2	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	tt003138	Vom Win	vom winc	http://ww	8.2	160414	14280	1939	video.mc	10	6	143	1263	653	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
11	tt003167	Mr. Smith	mr smith	http://ww	8.4	58169	7740	1939	video.mc	4	10	34	110	226	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	tt003213	Der Zaub	der zaub	http://ww	8.1	209506	6120	1939	video.mc	6	12	126	2363	477	3	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	tt003255	FrÃ¼cht	fr chte de	http://ww	8.2	45737	7740	1940	video.mc	6	5	20	135	257	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	tt003255	Der groÃ	der gro e	http://ww	8.5	87969	7500	1940	video.mc	5	1	37	181	173	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	tt003297	Rebecca	rebecca	http://ww	8.3	65670	7800	1940	video.mc	2	9	47	328	241	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
16	tt003346	Citizen K	citizen ka	http://ww	8.5	228617	7140	1941	video.mc	7	10	103	1439	1101	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
17	tt003387	Die Spur	die spur	http://ww	8.2	86012	6000	1941	video.mc	1	0	67	332	294	3	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	tt003458	Casablan	casablan	http://ww	8.6	296802	6120	1942	video.mc	5	6	153	382	960	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
19	tt003677	Frau ohn	frau ohne	http://ww	8.5	71266	6420	1944	video.mc	3	0	43	231	260	3	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	tt003686	Die beste	die beste	http://ww	8.3	30002	10320	1946	video.mc	13	1	15	129	218	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
21	tt003835	Tote sch	tote schl	http://ww	8.1	50020	6840	1946	video.mc	1	0	52	171	215	3	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
22	tt003865	It's a Wor	it s a wor	http://ww	8.7	197764	7800	1946	video.mc	6	0	65	531	626	3	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	tt003878	BerÃ¼ch	ber chtig	http://ww	8.2	56306	6060	1946	video.mc	2	1	56	102	260	3	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
24	tt004052	Fahrradc	fahrradd	http://ww	8.4	63797	5280	1948	video.mc	17	0	17	239	202	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	tt004074	Cocktail	cocktail f	http://ww	8.1	72370	4800	1948	video.mc	0	0	42	199	283	3	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
26	tt004089	Der Scha	der scha	http://ww	8.4	59298	7560	1948	video.mc	12	4	41	149	190	3	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	tt004195	Der dritte	der dritte	http://ww	8.4	85964	6240	1949	video.mc	3	4	62	338	364	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
28	tt004219	Alles Ã¼	alles ber	http://ww	8.4	61738	8280	1950	video.mc	15	15	42	253	276	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	tt004287	Rashomo	rashomo	http://ww	8.4	74219	5280	1950	video.mc	9	2	19	102	238	2	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	tt004301	Boulevar	boulevar	http://ww	8.6	102072	6600	1950	video.mc	17	13	58	1033	394	2	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	tt004407	Der Frem	der fremc	http://ww	8.1	69387	5520	1951	video.mc	1	1	42	326	245	3	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
32	tt004470	ZwÃ¶lf U	zw lf uhr	http://ww	8.2	60817	5100	1952	video.mc	12	10	48	361	309	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
33	tt004474	Ikiru - Eir	ikiru einn	http://ww	8.4	29478	8580	1952	video.mc	5	0	13	57	136	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	tt004515	Du sollst	du sollst	http://ww	8.4	108115	6180	1952	video.mc	4	7	61	333	441	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
35	tt004625	Ein Herz	ein herz	http://ww	8.1	69739	7080	1953	video.mc	7	14	50	136	193	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
36	tt004635	Stalag 17	stalag	http://ww	8.1	36013	6960	1953	video.mc	1	4	23	67	130	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
37	tt004691	Die Teufl	die teufl	http://ww	8.3	29309	6840	1955	video.mc	4	0	20	77	135	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
38	tt004691	Bei Anru	bei anruf	http://ww	8.2	72579	6300	1954	video.mc	3	2	51	232	223	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
39	tt004729	Die Faus	die faust	http://ww	8.3	74023	6480	1954	video.mc	21	7	45	352	255	2	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	tt004739	Das Fen	das fens	http://ww	8.6	229323	6720	1954	video.mc	6	6	98	711	580	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
41	tt004747	Die siebe	die siebe	http://ww	8.8	172690	12420	1954	video.mc	4	5	33	576	548	2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	tt004752	Das Lied	das lied	http://ww	8.2	35143	6480	1954	video.mc	10	4	25	108	108	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	tt004842	Die Nach	die nach	http://ww	8.1	47739	5580	1955	video.mc	1	0	42	152	322	3	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
44	tt004940	Killing (19	killing	http://ww	8.1	47740	5100	1956	video.mc	0	0	21	59	304	3	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
45	tt005008	Die zwÃ¶	die zw lf	http://ww	8.9	293725	5760	1957	video.mc	16	6	31	68	726	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	tt005021	Die BrÃ¼	die br ck	http://ww	8.3	109972	9660	1957	video.mc	24	5	59	160	243	3	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
47	tt005082	Wege zu	wege zu	http://ww	8.5	84945	5280	1957	video.mc	3	2	26	208	313	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
48	tt005097	Das sieb	das sieb	http://ww	8.3	80684	5760	1957	video.mc	7	1	28	193	236	2	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	tt005098	Wilde Erc	wilde erd	http://ww	8.3	45457	5460	1957	video.mc	12	2	29	107	130	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	tt005120	Zeugin d	zeugin d	http://ww	8.5	40385	6960	1957	video.mc	3	8	23	32	172	3	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Figure 1: IMDB movie review data set

4.2 Data Set Description:

The IMDB Movies Dataset contains information about movies which includes name of the movie, No. of articles posted and rating of the movie etc. Information about these movies was downloaded with wget for creating a movie recommendation app. The data was preprocessed and cleaned to be ready for machine learning applications.

Our dataset contains 15,000 observations which includes different of variables. Every variable has their own length in bytes and can be calculated by using back – of – the – envelope calculation.

Formula:

$$\text{Number of megabytes} = M = \frac{N*V*W + 4*N}{1024^2}$$

Where,

N = number of observations.

V = number of variables.

W = average width in bytes of a variable.

Find W (width), width

Integers	
-123 = 123	1
-32,767 = 32,740	2
-2,147,483,647 = 2,147,483620	4

Float	
Single precision	4
Double precision	8

strings	total number of length

Now, we are calculating the width of the variable.

Average width of variable $W = \frac{\text{total no of length}}{\text{Variable}}$

$$W= \frac{18}{1} = 18$$

$$M = \frac{15000*1*18 + 4*15000}{1024^2}$$

M= 270,00.057 bytes.

Fields	Description	Length in bytes
Content	describe the content of the movie	270,00.057 bytes.
title	mentions the movie name and the year	300,000.057 bytes.
words In Title	mentions only the movie title	300,000.057 bytes.
URL	link which specifies the movie title and the content in the movie	300,000.057 bytes.
IMDB Rating	movie ratings	60,000.05722 bytes.
Rating Count	counting total no of ratings given by the users	30,000.05722 bytes.
duration	length of the movie in seconds	30,000.04139 bytes.
year	type specifies which category the movie belongs to	30,000.05722 bytes.
No of Wins	total no of awards won	165,000.05722 bytes.
No of Nominations	total no of times the movie got nominated	15,000.01430 bytes.
No of Photos	no of photos taken in entire film	15,000.01430 bytes.

No of News Articles	articles published on the movie	15,000.01430 bytes.
No of User Reviews	it defines the total no reviews given to the movies	15,000.01430 bytes.
No of Genre	types of categories on movie.	15,000.01430 bytes.

Table 1: dataset description

4.3 Data Preprocessing:

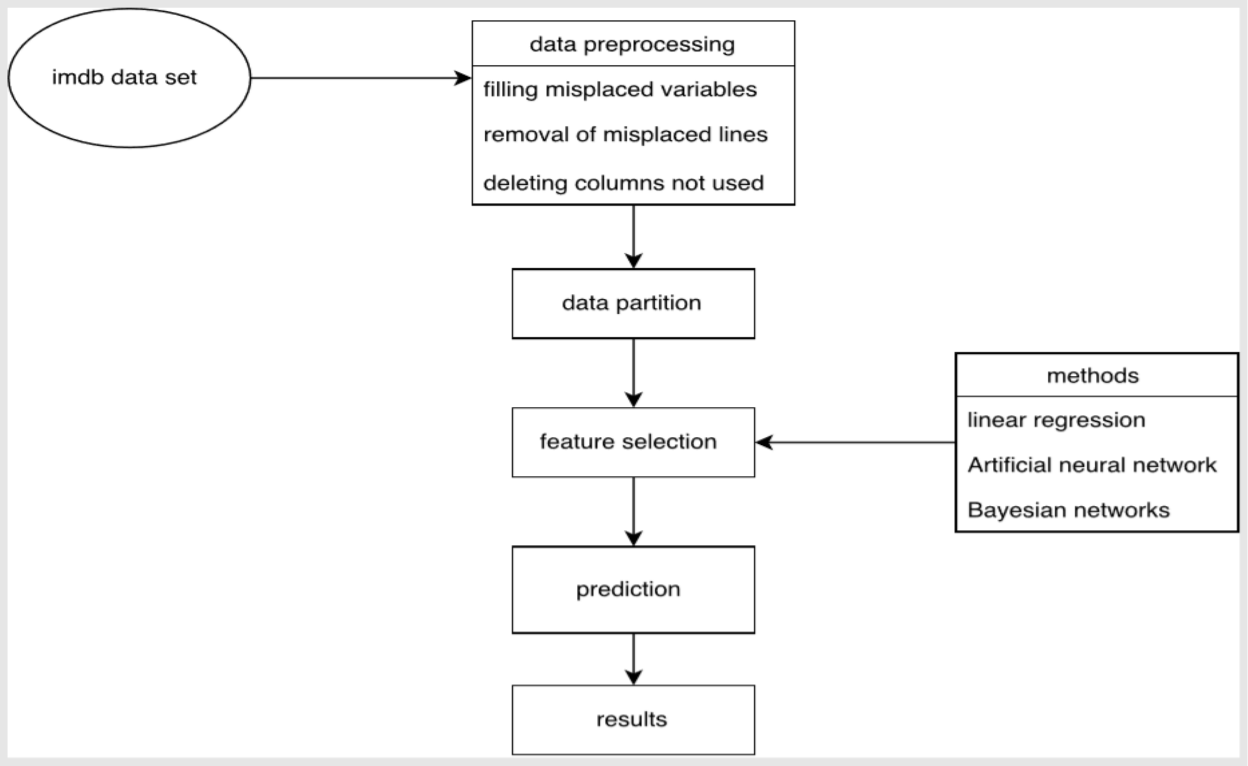


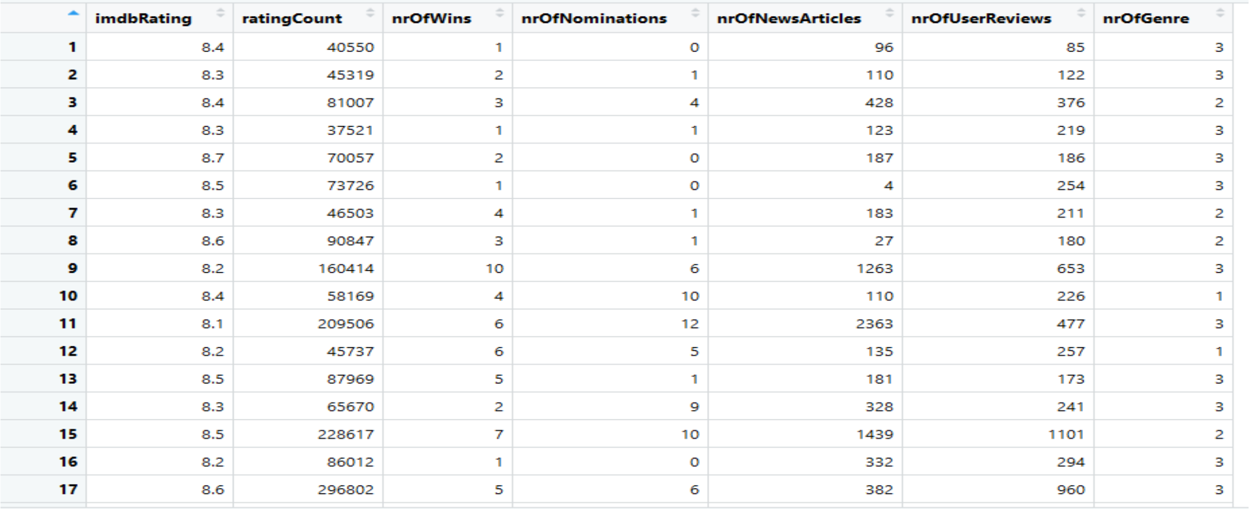
FIGURE 2: DATA PREPROCESSING

Data preprocessing can be defined in four different ways they are:

1. Data Cleaning
2. Data Integration.
3. Data Transformation.
4. Data reduction.

Data cleaning: Data cleaning is for attributes such as missing values, noisy data, inconsistency data.

Our project is all about predicting the IMDB movie ratings for that it is necessary to modify our dataset by deleting the columns that are not used. Once, the data preprocessing is done our data set looks like this as shown in the below figure.

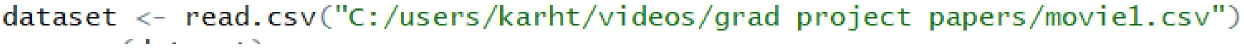


	imdbRating	ratingCount	nrOfWins	nrOfNominations	nrOfNewsArticles	nrOfUserReviews	nrOfGenre
1	8.4	40550	1	0	96	85	3
2	8.3	45319	2	1	110	122	3
3	8.4	81007	3	4	428	376	2
4	8.3	37521	1	1	123	219	3
5	8.7	70057	2	0	187	186	3
6	8.5	73726	1	0	4	254	3
7	8.3	46503	4	1	183	211	2
8	8.6	90847	3	1	27	180	2
9	8.2	160414	10	6	1263	653	3
10	8.4	58169	4	10	110	226	1
11	8.1	209506	6	12	2363	477	3
12	8.2	45737	6	5	135	257	1
13	8.5	87969	5	1	181	173	3
14	8.3	65670	2	9	328	241	3
15	8.5	228617	7	10	1439	1101	2
16	8.2	86012	1	0	332	294	3
17	8.6	296802	5	6	382	960	3

Figure 3: data preprocessing

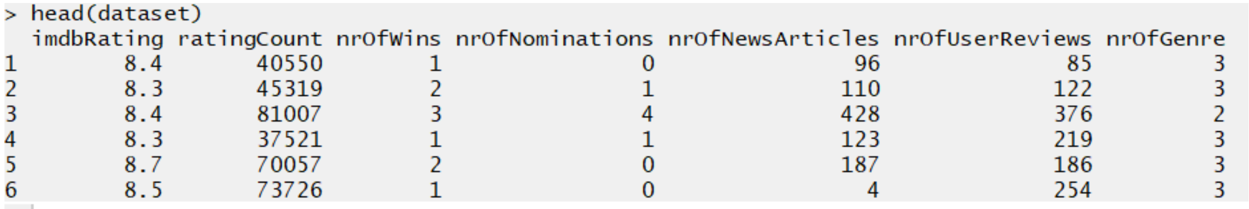
4.4. Data Normalization:

Before doing the data partition upload the data set in to R studio.



```
dataset <- read.csv("C:/users/karht/videos/grad project papers/movie1.csv")
```

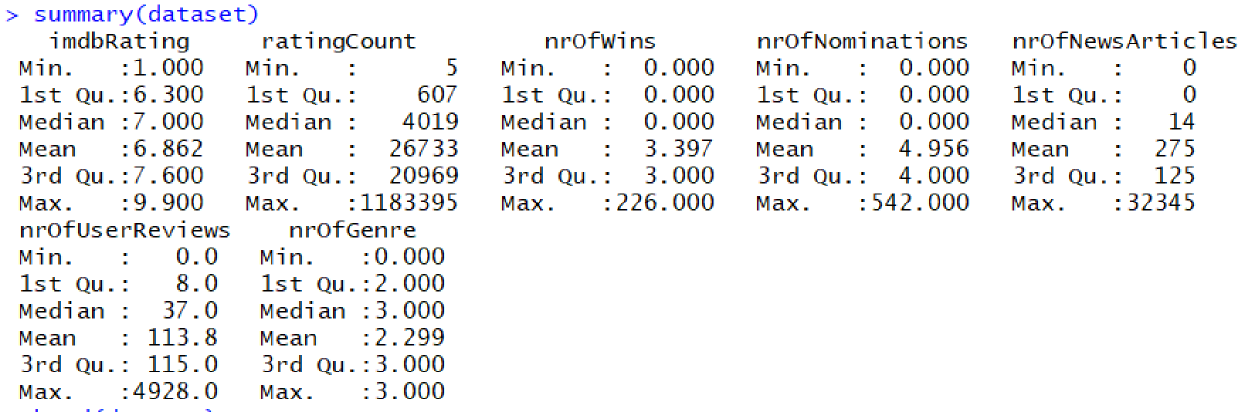
Now lest have a look in to the datatset using the head() function



```
> head(dataset)
  imdbRating ratingCount nrOfWins nrOfNominations nrOfNewsArticles nrOfUserReviews nrOfGenre
1        8.4       40550        1               0               96              85         3
2        8.3       45319        2               1              110             122         3
3        8.4       81007        3               4              428             376         2
4        8.3       37521        1               1              123             219         3
5        8.7       70057        2               0              187             186         3
6        8.5       73726        1               0                4             254         3
```

FIGURE 4: Head of data set

For getting the clear understand of the data lest summarize the data

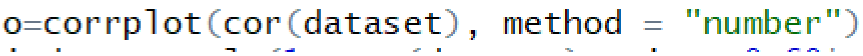


```
> summary(dataset)
   imdbRating      ratingCount          nrOfWins       nrOfNominations    nrOfNewsArticles
 Min.   :1.000   Min.   :      5   Min.   :  0.000   Min.   :  0.000   Min.   :    0
 1st Qu.:6.300   1st Qu.:    607   1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:    0
 Median :7.000   Median :   4019   Median :  0.000   Median :  0.000   Median :   14
 Mean   :6.862   Mean   :  26733   Mean   :  3.397   Mean   :  4.956   Mean   :  275
 3rd Qu.:7.600   3rd Qu.:  20969   3rd Qu.:  3.000   3rd Qu.:  4.000   3rd Qu.:  125
 Max.   :9.900   Max.   :1183395   Max.   :226.000   Max.   :542.000   Max.   :32345
 nrOfUserReviews      nrOfGenre
 Min.   :   0.0   Min.   :0.000
 1st Qu.:   8.0   1st Qu.:2.000
 Median :  37.0   Median :3.000
 Mean   : 113.8   Mean   :2.299
 3rd Qu.: 115.0   3rd Qu.:3.000
 Max.   :4928.0   Max.   :3.000
```

Visualize the dataset using the “correlation” packge. For using this package

Install.packae(“corrplot”) #installs the packes

Library(corrplot) # ;loads the package into R studio



```
o=corrplot(cor(dataset), method = "number")
```

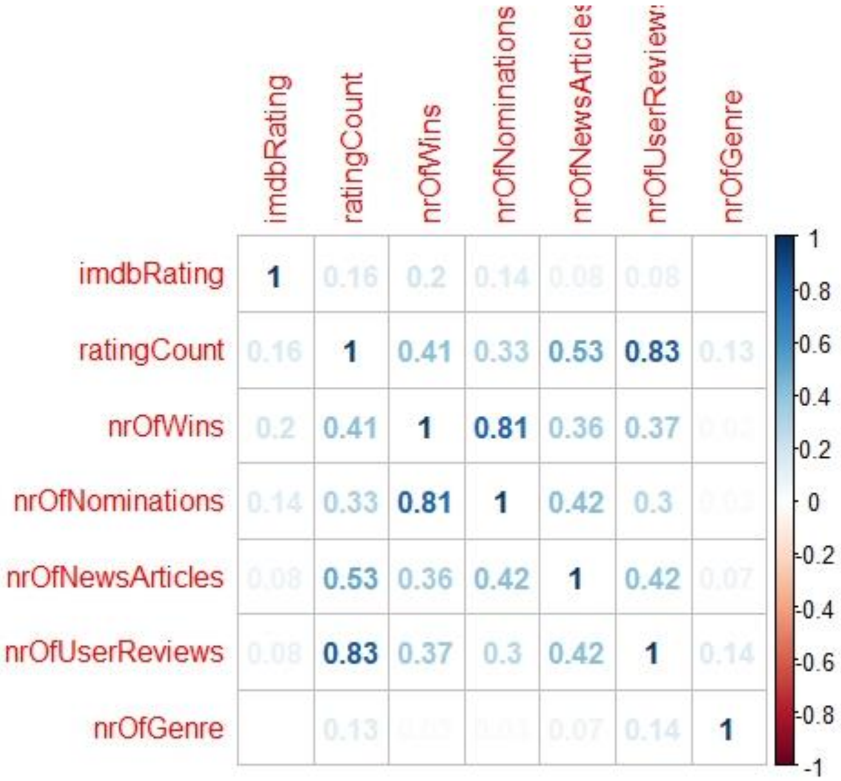
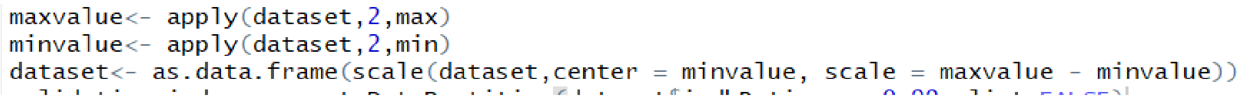


Figure 5: correleration plot for dataset

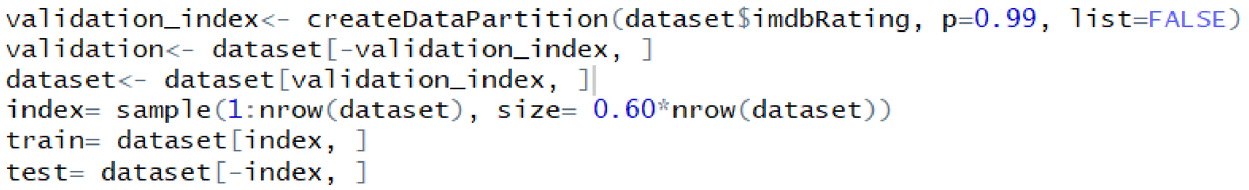
in our data we have different types of value which might affect in our prediction accuracy. For improving the prediction accuracy data normalization and scaling is done because every time they give the same results. Min-max normalization technique is used which applies the k-nearest neighbors method.



```
maxvalue<- apply(dataset,2,max)
minvalue<- apply(dataset,2,min)
dataset<- as.data.frame(scale(dataset,center = minvalue, scale = maxvalue - minvalue))
```

4.5. Data Partition:

first, separate the validation set from the entire dataset contains around 100 rows which will be used for validating the results. Later, divide the dataset is divided into 60% as training data and 40% that will hold back as a testing set. The main purpose of having more training data is because more training data gives the better result than having more testing data.



```
validation_index<- createDataPartition(dataset$imdbRating, p=0.99, list=FALSE)
validation<- dataset[-validation_index, ]
dataset<- dataset[validation_index, ]
index= sample(1:nrow(dataset), size= 0.60*nrow(dataset))
train= dataset[index, ]
test= dataset[-index, ]
```

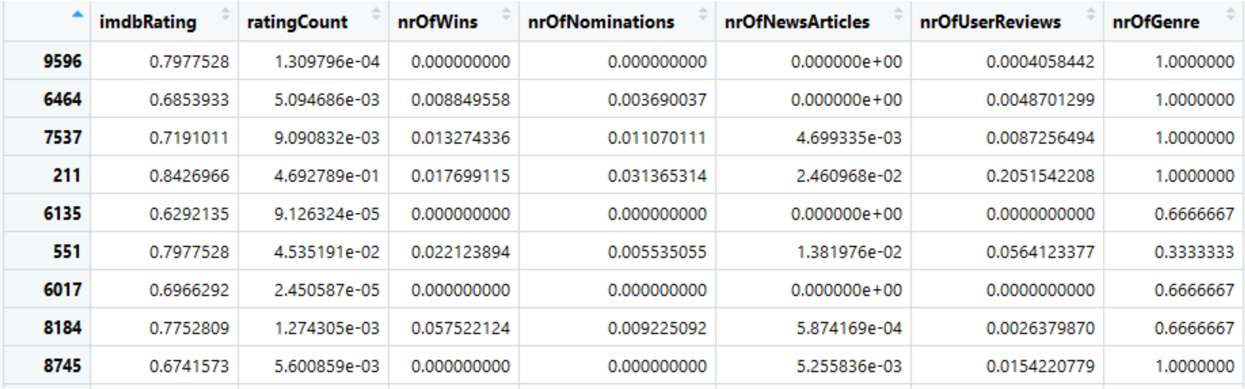
Once the data set has divided the training and the testing data contains different observations in dataset shown in table 1. You can see the observations in the right-side console in the R Studio. Since the data is divided into 60% training and 40% testing, you can observe training data has 9097 observations and testing data has 3900 observations. The below table 2: shows clearly about observations and variables for both training data and testing data.

	Rows	Variables
training	9097	7
testing	3900	7

Table 2: training and testing data

Training The Data:

Once the data partition is done train the model on training data. And the visualize the results for training data. Our training data looks as shown in figure6:



	imdbRating	ratingCount	nrOfWins	nrOfNominations	nrOfNewsArticles	nrOfUserReviews	nrOfGenre
9596	0.7977528	1.309796e-04	0.000000000	0.000000000	0.000000e+00	0.0004058442	1.0000000
6464	0.6853933	5.094686e-03	0.008849558	0.003690037	0.000000e+00	0.0048701299	1.0000000
7537	0.7191011	9.090832e-03	0.013274336	0.011070111	4.699335e-03	0.0087256494	1.0000000
211	0.8426966	4.692789e-01	0.017699115	0.031365314	2.460968e-02	0.2051542208	1.0000000
6135	0.6292135	9.126324e-05	0.000000000	0.000000000	0.000000e+00	0.0000000000	0.6666667
551	0.7977528	4.535191e-02	0.022123894	0.005535055	1.381976e-02	0.0564123377	0.3333333
6017	0.6966292	2.450587e-05	0.000000000	0.000000000	0.000000e+00	0.0000000000	0.6666667
8184	0.7752809	1.274305e-03	0.057522124	0.009225092	5.874169e-04	0.0026379870	0.6666667
8745	0.6741573	5.600859e-03	0.000000000	0.000000000	5.255836e-03	0.0154220779	1.0000000

Figure 6: Training data

Summary of training data:

```
> summary(train)
   imdbRating      ratingCount          nrOfWins       nrOfNominations
 Min.   :0.0000   Min.   :0.0000000   Min.   :0.00000   Min.   :0.000000
 1st Qu.:0.5955   1st Qu.:0.0005349   1st Qu.:0.00000   1st Qu.:0.000000
 Median :0.6742   Median :0.0034545   Median :0.00000   Median :0.000000
 Mean   :0.6576   Mean   :0.0223794   Mean   :0.01551   Mean   :0.009155
 3rd Qu.:0.7416   3rd Qu.:0.0177949   3rd Qu.:0.01327   3rd Qu.:0.007380
 Max.   :1.0000   Max.   :1.0000000   Max.   :1.00000   Max.   :1.000000
 nrOfNewsArticles    nrOfUserReviews      nrOfGenre
 Min.   :0.0000000   Min.   :0.000000   Min.   :0.0000
 1st Qu.:0.0000000   1st Qu.:0.001623   1st Qu.:0.6667
 Median :0.0004638   Median :0.007711   Median :0.6667
 Mean   :0.0081709   Mean   :0.023169   Mean   :0.7641
 3rd Qu.:0.0038878   3rd Qu.:0.023539   3rd Qu.:1.0000
 Max.   :1.0000000   Max.   :1.000000   Max.   :1.0000
```

visualizing the training data:

Boxplot:

Boxplot is used for displaying the distribution of dataset. Boxplot contains five different levels they are,

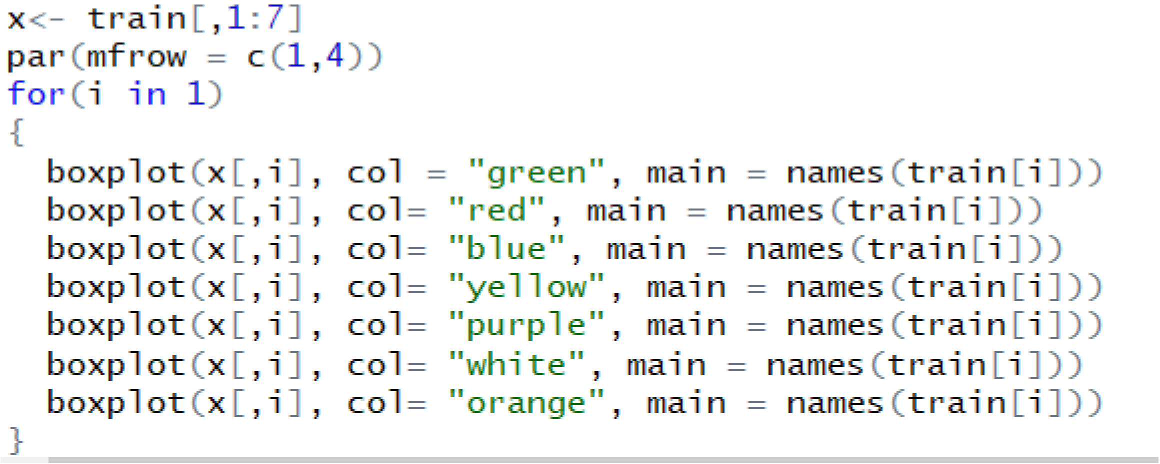
1. Minimum
2. First quartile
3. Median
4. Third quartile
5. Maximum

Inside the rectangle box the straight line shows the median. Above and below shows the maximum and the minimum For, using boxplot install package called “plot3D”. We have two different ways for installing the packages.

1. Using the function install. Packages(“plot3D”)
2. In R studio click on packages button and search for the package that you need and install the package.

After installing the package. Load the package in to R studio by using a function

library(caret) # load the package in to R studio.



```
x<- train[,1:7]
par(mfrow = c(1,4))
for(i in 1)
{
  boxplot(x[,i], col = "green", main = names(train[i]))
  boxplot(x[,i], col= "red", main = names(train[i]))
  boxplot(x[,i], col= "blue", main = names(train[i]))
  boxplot(x[,i], col= "yellow", main = names(train[i]))
  boxplot(x[,i], col= "purple", main = names(train[i]))
  boxplot(x[,i], col= "white", main = names(train[i]))
  boxplot(x[,i], col= "orange", main = names(train[i]))
}
```

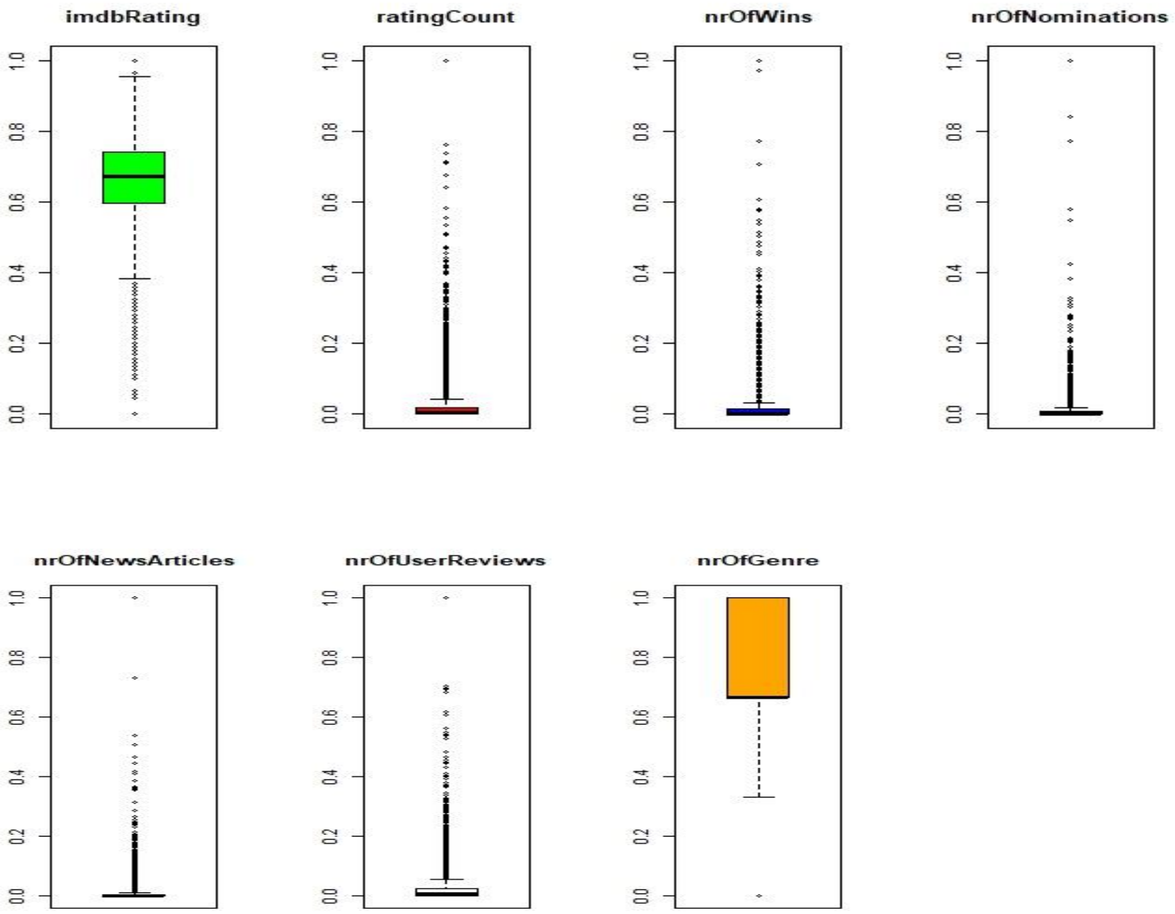


Figure 7: Boxploty for training data.

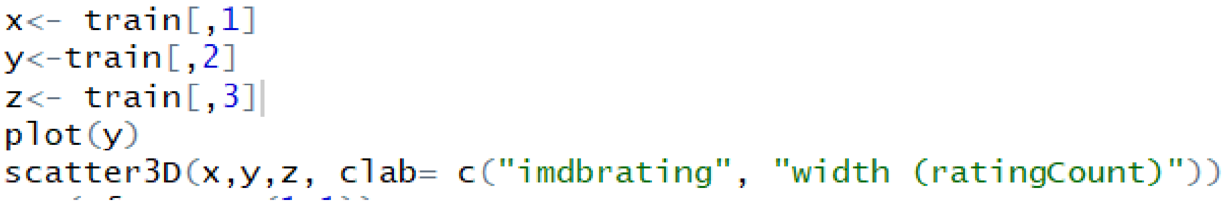
Scatter3D:

This is the another method for visualizing the dataset known as scatter plot. For using this method first pacakges needed to be installed

Install.packages(“plot3D”)

Once the packages are installed. Load the package in to R console using the function

library(plot3D)



```
x<- train[,1]
y<-train[,2]
z<- train[,3]
plot(y)
scatter3D(x,y,z, clab= c("imdbrating", "width (ratingCount)"))
```

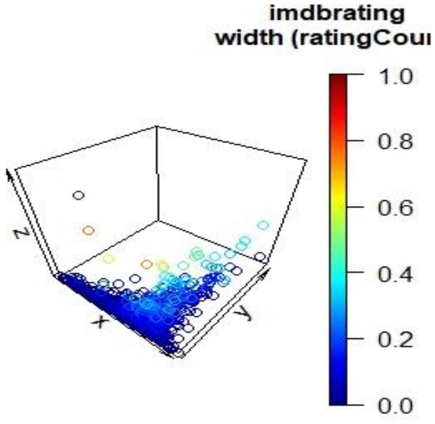
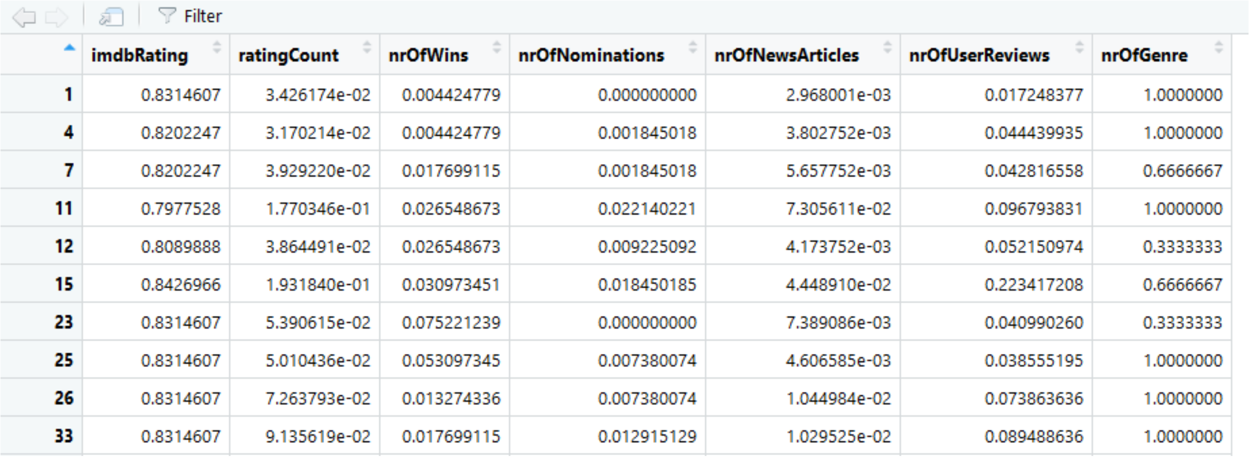


Figure 8 : Scatter3D plot for training data.

Testing Data:

Our test data consists of 3900 rows and 7 variables. Now, we test the model on test data. When testing is done visualize the data using different type of techniques. Below figure 8 shows testing data.



	imdbRating	ratingCount	nrOfWins	nrOfNominations	nrOfNewsArticles	nrOfUserReviews	nrOfGenre
1	0.8314607	3.426174e-02	0.004424779	0.000000000	2.968001e-03	0.017248377	1.0000000
4	0.8202247	3.170214e-02	0.004424779	0.001845018	3.802752e-03	0.044439935	1.0000000
7	0.8202247	3.929220e-02	0.017699115	0.001845018	5.657752e-03	0.042816558	0.6666667
11	0.7977528	1.770346e-01	0.026548673	0.022140221	7.305611e-02	0.096793831	1.0000000
12	0.8089888	3.864491e-02	0.026548673	0.009225092	4.173752e-03	0.052150974	0.3333333
15	0.8426966	1.931840e-01	0.030973451	0.018450185	4.448910e-02	0.223417208	0.6666667
23	0.8314607	5.390615e-02	0.075221239	0.000000000	7.389086e-03	0.040990260	0.3333333
25	0.8314607	5.010436e-02	0.053097345	0.007380074	4.606585e-03	0.038555195	1.0000000
26	0.8314607	7.263793e-02	0.013274336	0.007380074	1.044984e-02	0.073863636	1.0000000
33	0.8314607	9.135619e-02	0.017699115	0.012915129	1.029525e-02	0.089488636	1.0000000

Figure 8 Testing data

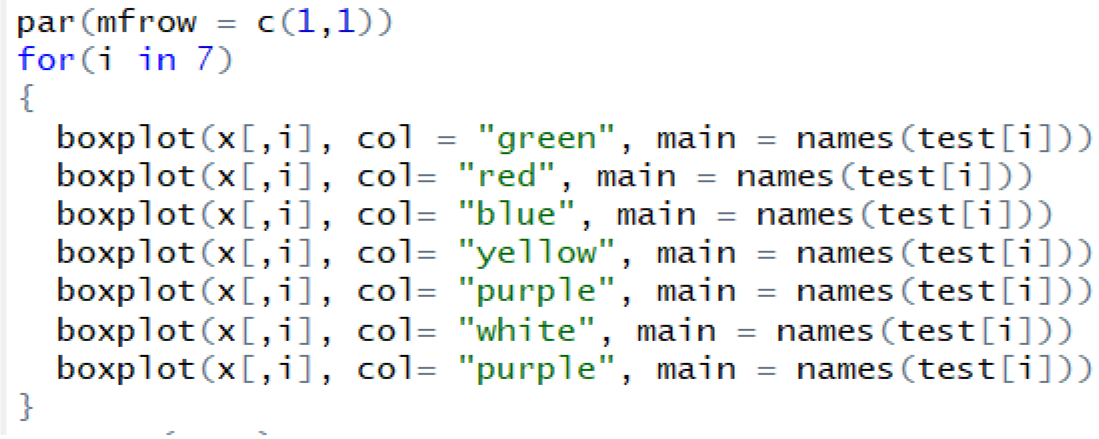
```
> summary(test)
  imdbRating      ratingCount          nrOfWins      nrOfNominations    nrOfNewsA
rticles
 Min.   :0.03371   Min.   :0.0000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.
0000000
 1st Qu.:0.59551   1st Qu.:0.0004749   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.
:0.0000000
 Median :0.67416   Median :0.0032694   Median :0.00000   Median :0.000000   Me
dian :0.0004019
 Mean   :0.66014   Mean   :0.0228953   Mean   :0.01432   Mean   :0.009129   Mean
:0.0090008
 3rd Qu.:0.74157   3rd Qu.:0.0176607   3rd Qu.:0.01327   3rd Qu.:0.007380   3rd Q
u.:0.0038646
 Max.   :0.98876   Max.   :0.9705254   Max.   :0.78319   Max.   :0.976015   Max.   :
0.7909105
 nrOfUserReviews      nrOfGenre
 Min.   :0.000000   Min.   :0.0000
 1st Qu.:0.001623   1st Qu.:0.6667
 Median :0.007508   Median :1.0000
 Mean   :0.022958   Mean   :0.7694
 3rd Qu.:0.023336   3rd Qu.:1.0000
 Max.   :0.797484   Max.   :1.0000
```

Boxplot:

As mentioned before install the package “caret”

Load the package in to R studio

Library(caret)



```
par(mfrow = c(1,1))
for(i in 7)
{
  boxplot(x[,i], col = "green", main = names(test[i]))
  boxplot(x[,i], col= "red", main = names(test[i]))
  boxplot(x[,i], col= "blue", main = names(test[i]))
  boxplot(x[,i], col= "yellow", main = names(test[i]))
  boxplot(x[,i], col= "purple", main = names(test[i]))
  boxplot(x[,i], col= "white", main = names(test[i]))
  boxplot(x[,i], col= "purple", main = names(test[i]))
}
```

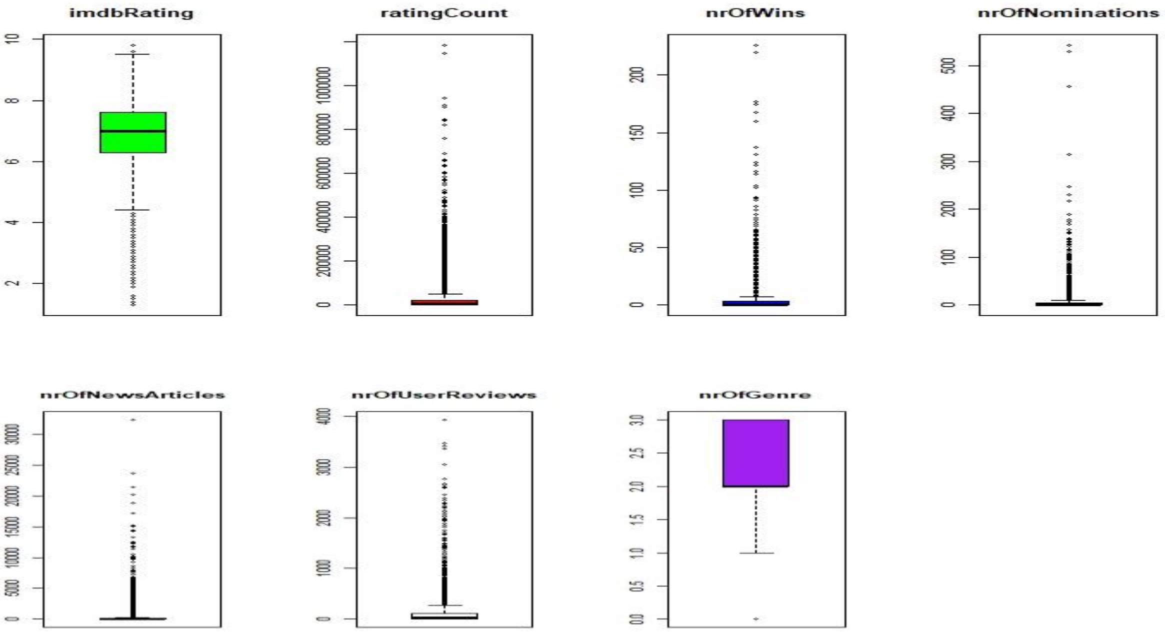
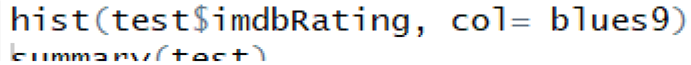


Figure 9 : boxplot test data

Histogram for test data:



```
hist(test$imdbRating, col= blues9)
summary(test)
```

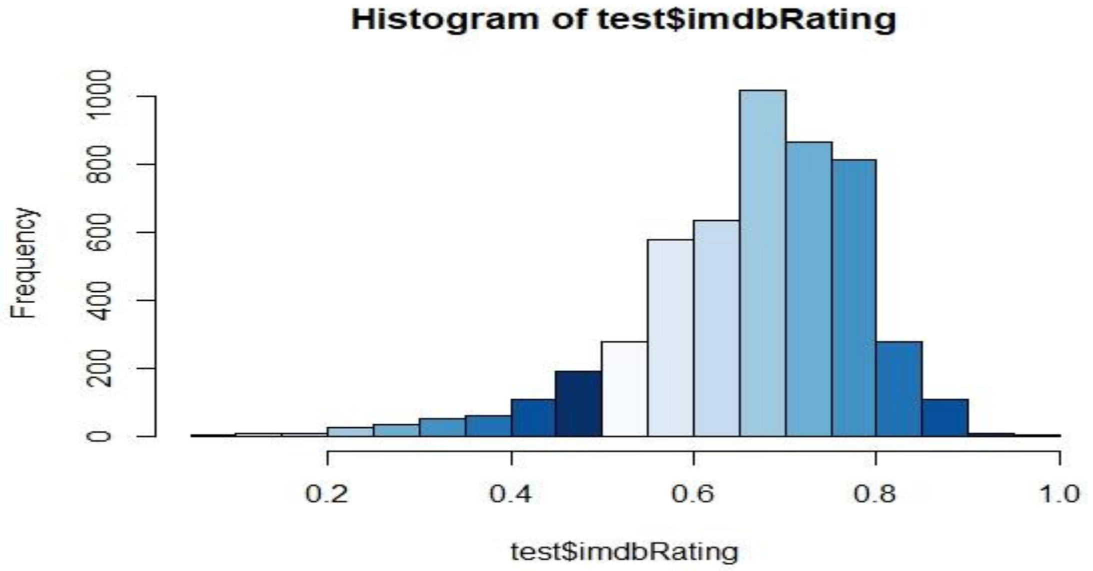
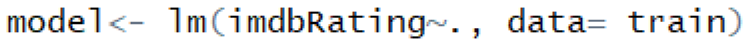
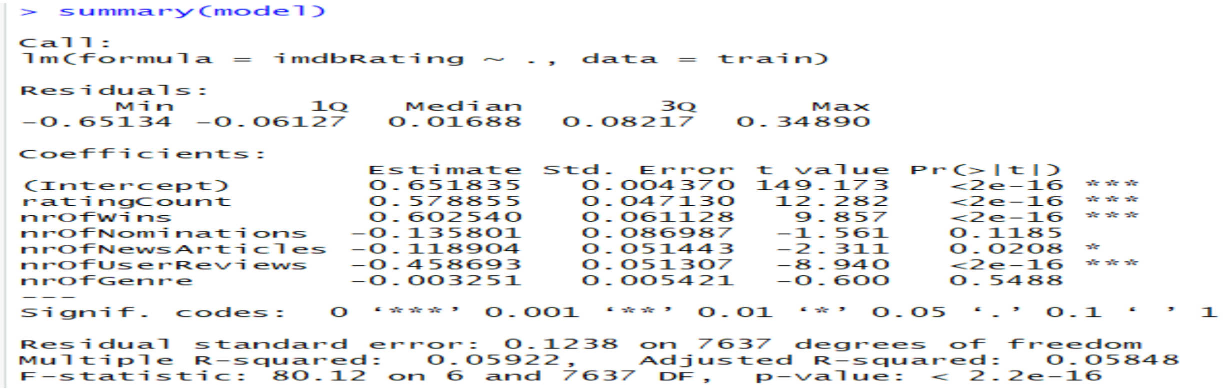


Figure 10: Histogram plot for test data

After training and testing the data apply the model on train data

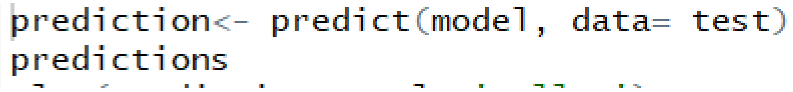


```
model<- lm(imdbRating~., data= train)
```

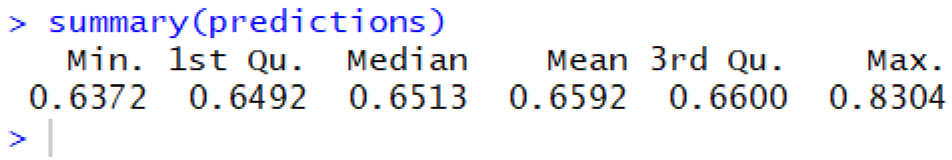


```
> summary(model)

Call:
lm(formula = imdbRating ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.65134 -0.06127  0.01688  0.08217  0.34890 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)        0.651835   0.004370 149.173   <2e-16 ***
ratingCount        0.578855   0.047130  12.282   <2e-16 ***
nrOfWins           0.602540   0.061128   9.857   <2e-16 ***
nrOfNominations   -0.135801   0.086987  -1.561   0.1185    
nrOfNewsArticles  -0.118904   0.051443  -2.311   0.0208 *  
nrOfUserReviews   -0.458693   0.051307  -8.940   <2e-16 ***
nrOfGenre         -0.003251   0.005421  -0.600   0.5488    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1238 on 7637 degrees of freedom
Multiple R-squared:  0.05922,	Adjusted R-squared:  0.05848 
F-statistic: 80.12 on 6 and 7637 DF,  p-value: < 2.2e-16
```

Now predict the model on the test data



```
prediction<- predict(model, data= test)
predictions
```



```
> summary(predictions)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6372  0.6492  0.6513  0.6592  0.6600  0.8304 
> 
```

After implementing train and test models we get the results as shown in the below figure**:**

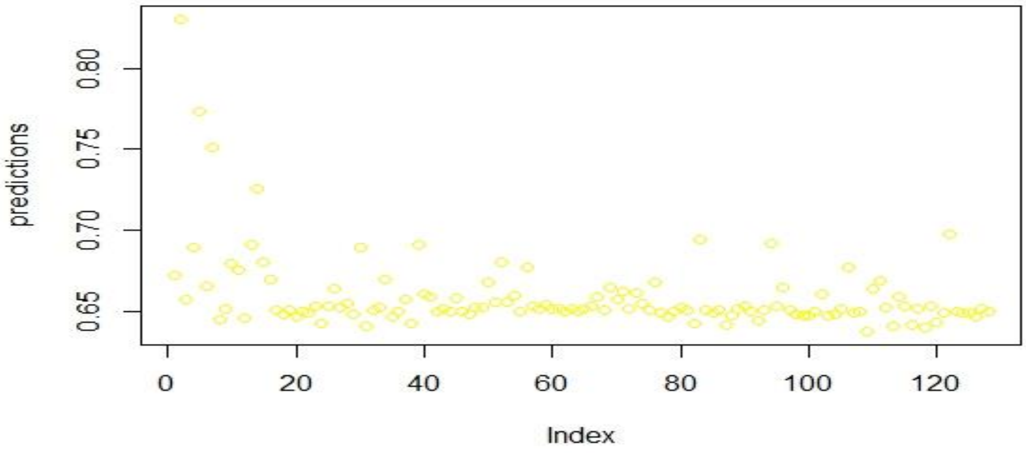


Figure 11: train and test graph

5.Construction of prediction models

5.1 Linear regression:

step1: install the packages that are necessary for using linear regression

install. Packages(“alr3”)

install. packages(“caret”)

install. Packages(“plot3D”)

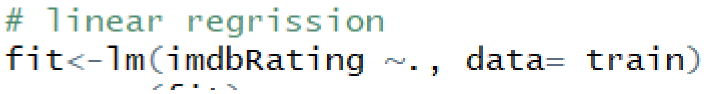
Step2: load the packages in to the R console using a function called

library(alr3)

library(caret)

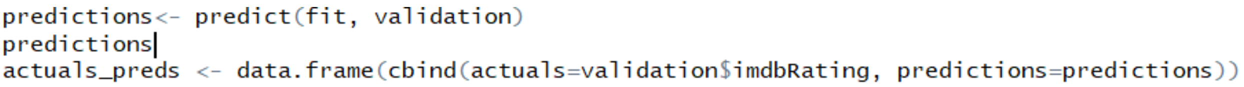
library(caret)

Step3: fit the linear regression model to the test data using lm() function. The lm() function is filled with two arguments formula and the data. The data is typically data.frame and the formula is a object class formula.



```
# linear regrission
fit<-lm(imdbRating ~., data= train)
```

Step4: after the fitting the model predict the model using the validation set.

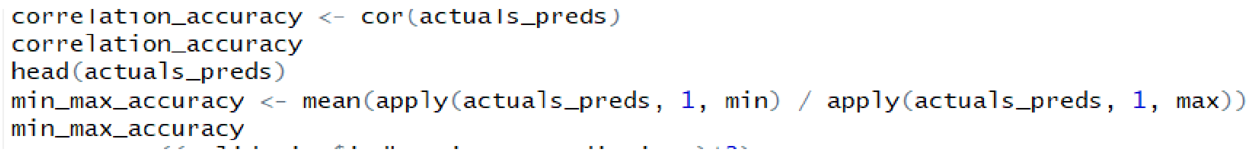


```
predictions<- predict(fit, validation)
predictions
actuals_preds <- data.frame(cbind(actuals=validation$imdbRating, predictions=predictions))
```

Step5: once the prediction is done calculate the accuracy of the prediction and the error rates of the predicted values. Actuals and the predicted values are used for calculating the accuracy. calculate the accuracy using min-max formula.

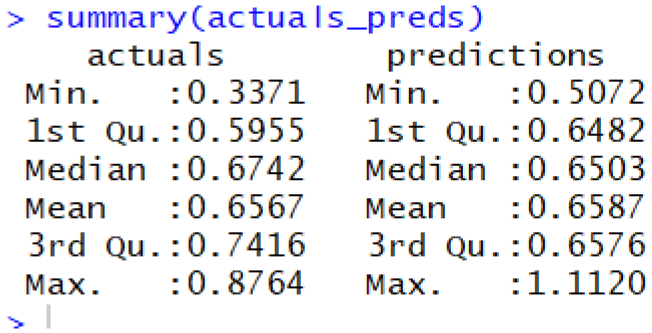
Minmax accuracy = mean(min(actual, predictions)/max(actuals, predictions))

By calculating the minmax accuracy we got 86% of accuracy.



```
correlation_accuracy <- cor(actuals_preds)
correlation_accuracy
head(actuals_preds)
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
min_max_accuracy
```

```
> head(actuals_preds)
    actuals predictions
3  0.8409091   0.6662198
5  0.8750000   0.6765640
6  0.8522727   0.6694844
7  0.8295455   0.6710333
8  0.8636364   0.6920494
11 0.8068182   0.7128138
```



```
> summary(actuals_preds)
    actuals          predictions
 Min.   :0.3371   Min.   :0.5072
 1st Qu.:0.5955   1st Qu.:0.6482
 Median :0.6742   Median :0.6503
 Mean   :0.6567   Mean   :0.6587
 3rd Qu.:0.7416   3rd Qu.:0.6576
 Max.   :0.8764   Max.   :1.1120
>
```

The below figure 12: shows the histogram graph of predicted values

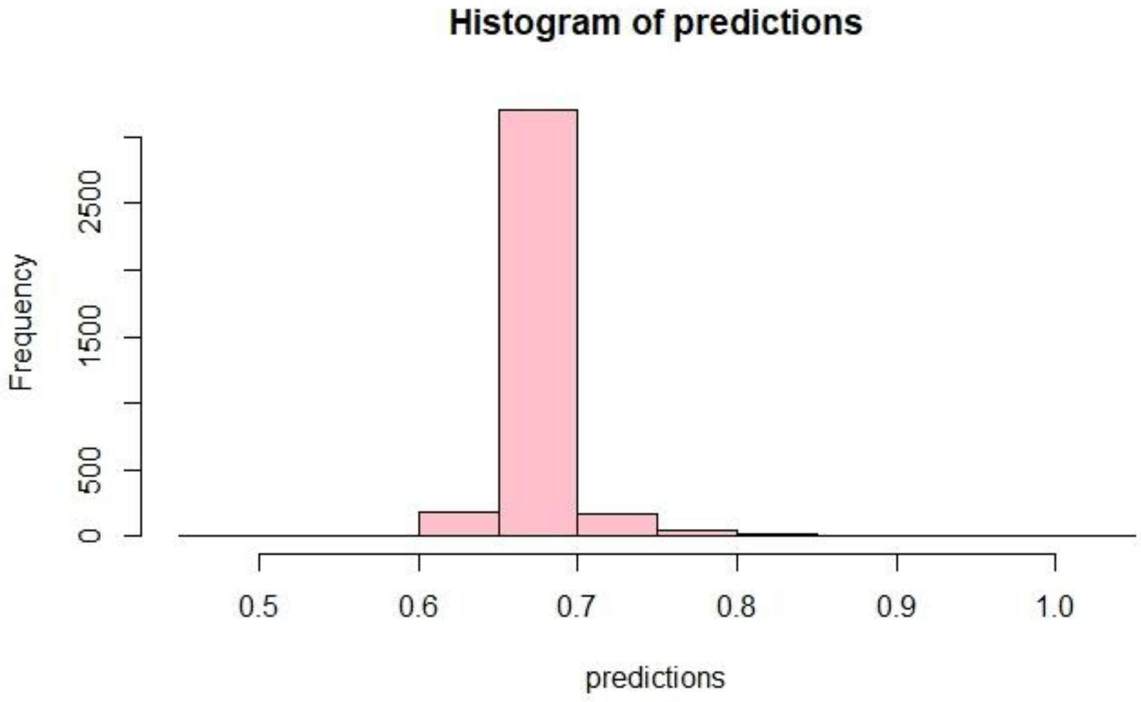
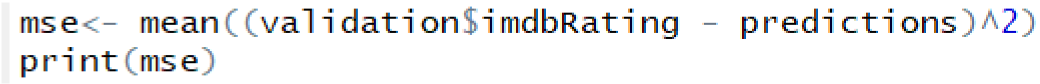


Figure 12: histogram for linear regression

Step5: mean square is calculated to find the accuracy of the model.

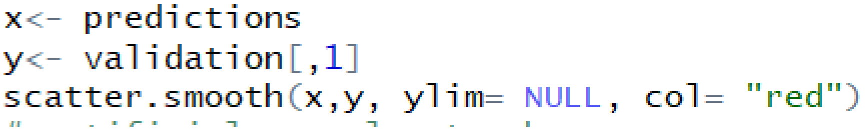


```
mse<- mean((validation$imdbRating - predictions)^2)
print(mse)
```

```
> print(mse)
[1] 0.01576418
```

Step6: now visualize the linear regression models using scatterplot

Scatterplot for prediction results



```
x<- predictions
y<- validation[,1]
scatter.smooth(x,y, ylim= NULL, col= "red")
```

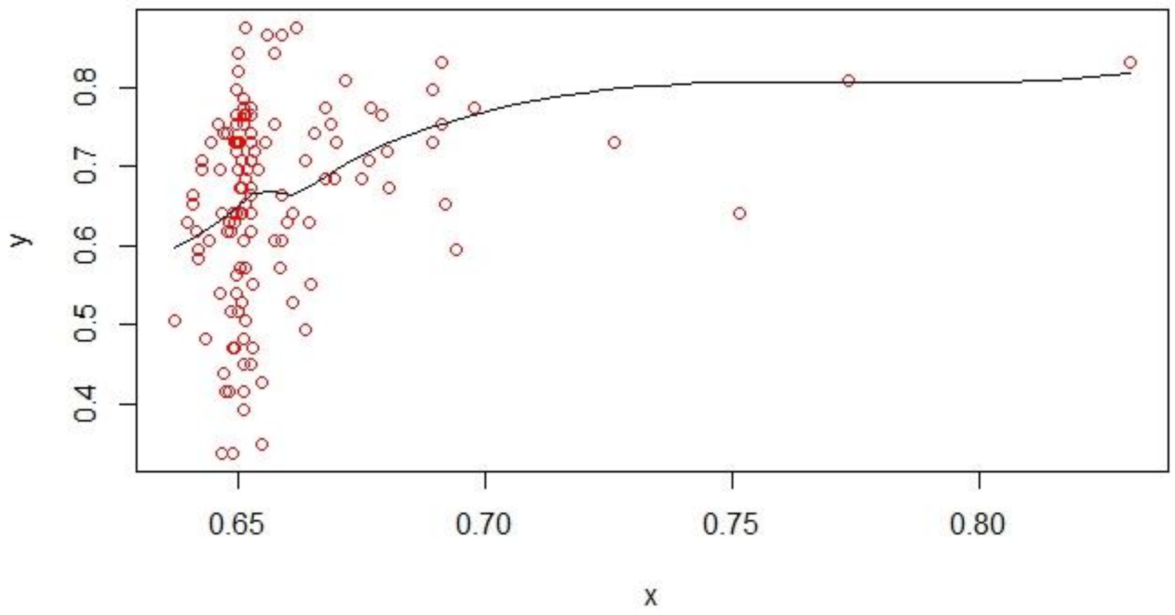


Figure 13: scatter. Smooth plot for linear regression

From the above experiment the predicted output is 0.7 and the mean square error rate of 0.01576 with 86% of accuracy. the below plot shows the actual and the predicted results for the linear regression prediction model.

5.2 Artificial Neural Network:

While forming a neural network data normalization should be done. Because the predicted values of the neural network can be compared to the actual data. If data normalization is not done then we may get different predicted values every time. We used max-min normalization techniques for scaling the data.

Now the scaled data is used for the fitting the neural network and we visualize the neural network with weights for each variable.

Step1: packages need to be installed to the neural net

Install. Packages(“neuralnet”)

Install.packaes(“rcolorbrewer”)

Install. Packages(“plot3D”)

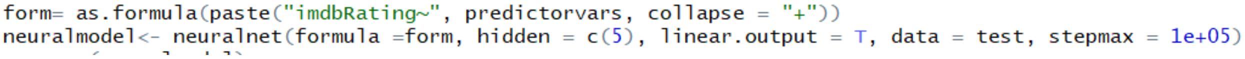
Step2: when the packages are installed. Load the packages in to R console by

library(neuralnet)

library(plot3D)

library(rcolorbrewer)

Step3: when the packages are loaded and ready for use. Fit the neural net package for the testing set.



```
form= as.formula(paste("imdbRating~", predictorvars, collapse = "+"))
neuralmodel<- neuralnet(formula =form, hidden = c(5), linear.output = T, data = test, stepmax = 1e+05)
```

Our model has five hidden layers. The below figure explains the neural network the black lines shows the connections with weights and the weights are calculated using back propagation algorithm. And the blue line displays bias terms.

We can plot the neural net buy plot() function,

plot(neuralmodel)

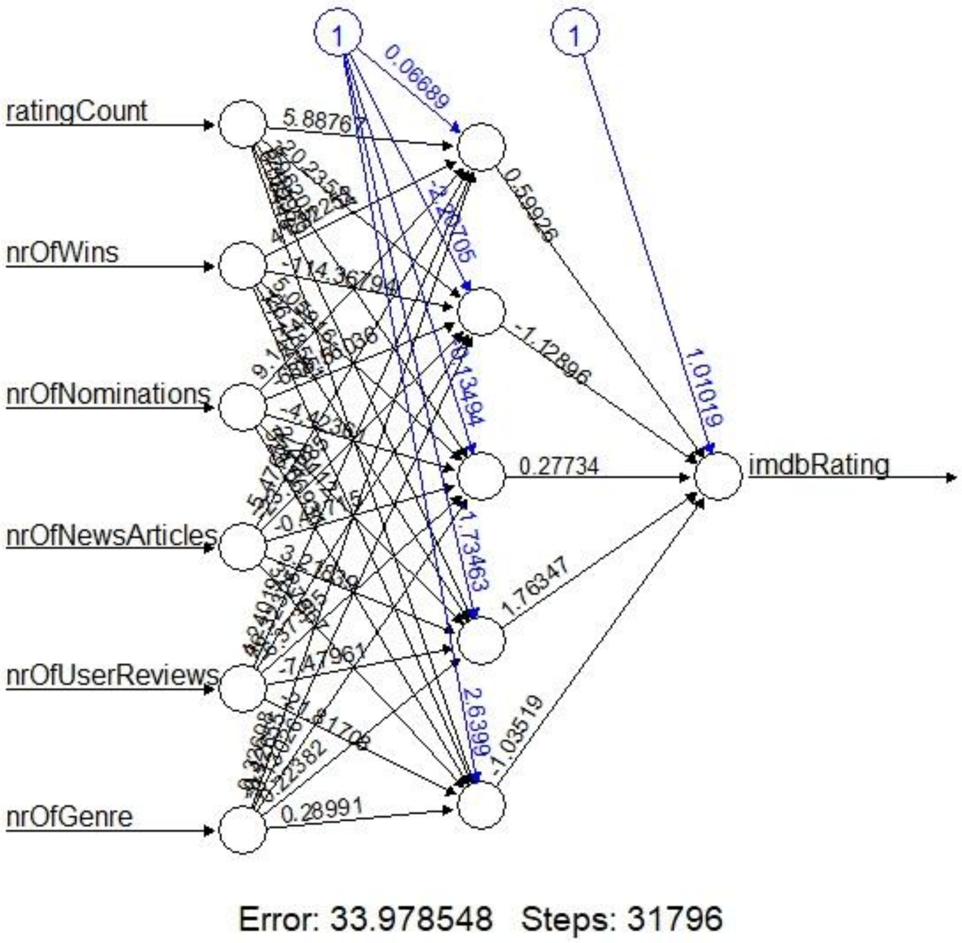
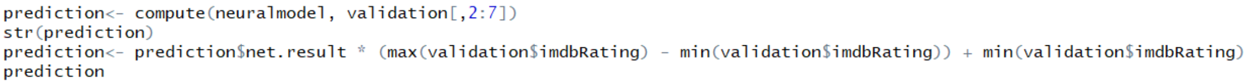


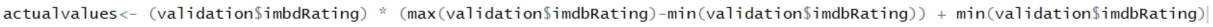
Figure 14: Neural Network Model

Step4: after fitting the neural network. Calculate the prediction based on the neural model using validation set.

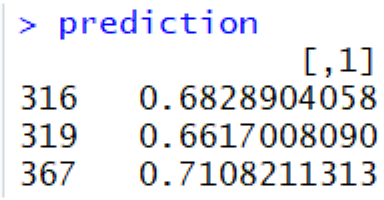


```
prediction<- compute(neuralmodel, validation[,2:7])
str(prediction)
prediction<- prediction$net.result * (max(validation$imdbRating) - min(validation$imdbRating)) + min(validation$imdbRating)
prediction
```

Setp5: finding the actual values for the neural model.



```
actualvalues<- (validation$imbdRating) * (max(validation$imdbRating)-min(validation$imdbRating)) + min(validation$imdbRating)
```



```
> prediction
                  [,1]
316    0.6828904058
319    0.6617008090
367    0.7108211313
```

Step6: the predicted values are visualized using the histograma() function. The below figure shows the histogram of neural net predicted values

```
hist(prediction, col= blues9)
```

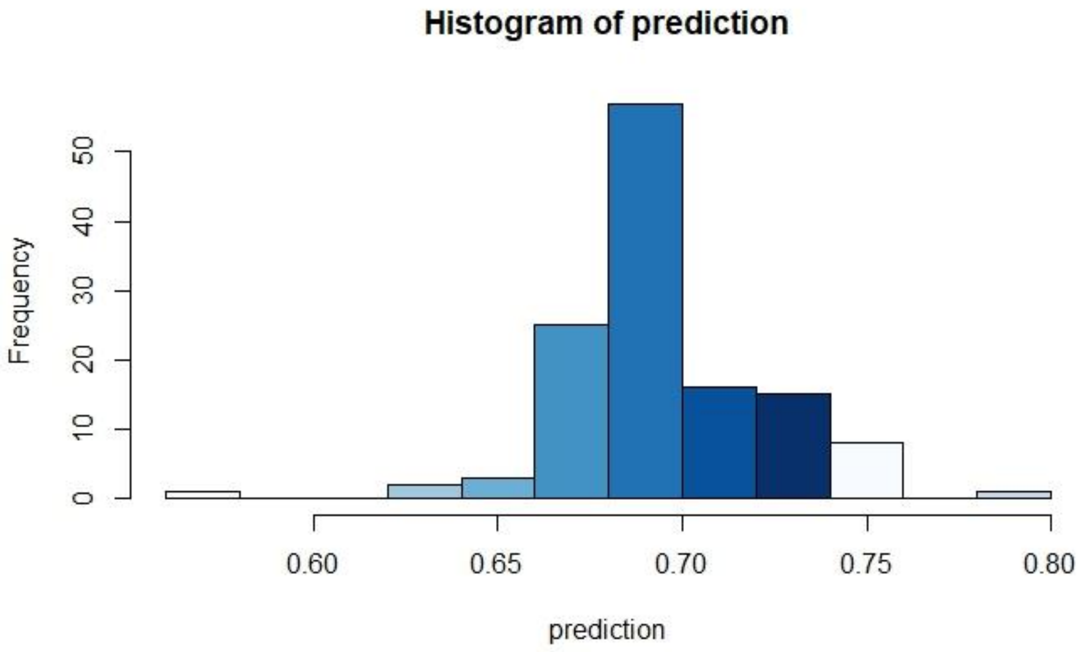
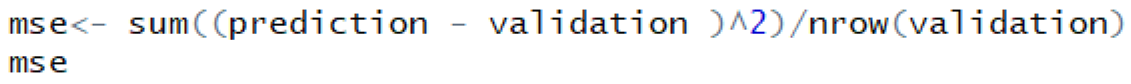


Figure 15: histogram for neural network model

Step7: predict the rating using the neural network model. Once the prediction is done compare the ratings with the real ratings. And for finding the mean square error calculate the MSE of the predicted vs real value. The mean square error for the model is



```
mse<- sum((prediction - validation )^2)/nrow(validation)
mse
```

```
> mse
[1] 2.427057287
```

Step8: visualizing the neural net model using scatter model

From the above experiment the predicted output is 0.6 and the mean square error rate is 2.424 with 67% of accuracy. the below plot shows the actual and the predicted results for the artificial neural network prediction model.

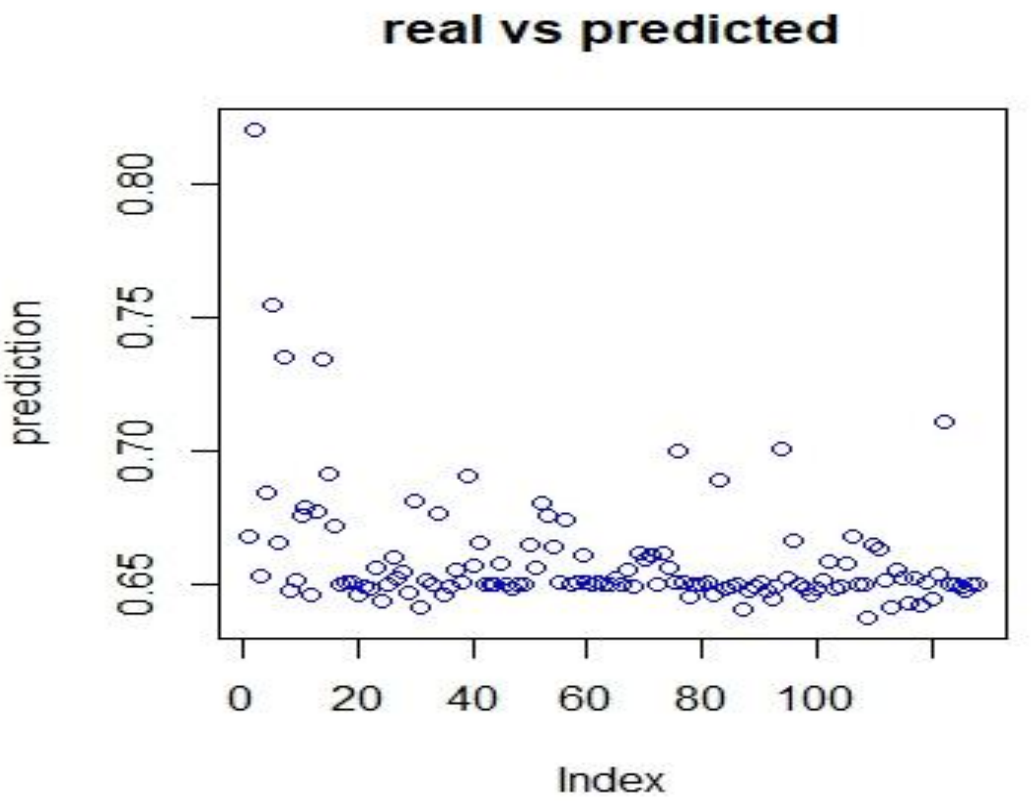


Figure 16: Predicted values for neural network

.

5.3 Bayesian Method:

Bnlearn Is a package for graphical structure of the Bayesian networks. For using the Bayesian networks in R., the packages need to be installed

Step1: the packages need to be installed

Install. Packages("bnlearn")

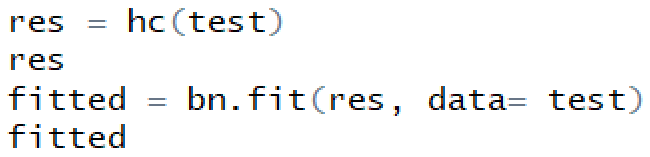
Install.packaes("plot3d") # this package is used for the graphical representation.

Step2: once the packages are installed load the packages in to the R console.

Library(bnlearn)

Library(plot3d)

Step3: fit the bnlearn model to the testing dataset.



```
res = hc(test)
res
fitted = bn.fit(res, data= test)
fitted
```

Step4: visualize the Bayesian network residual

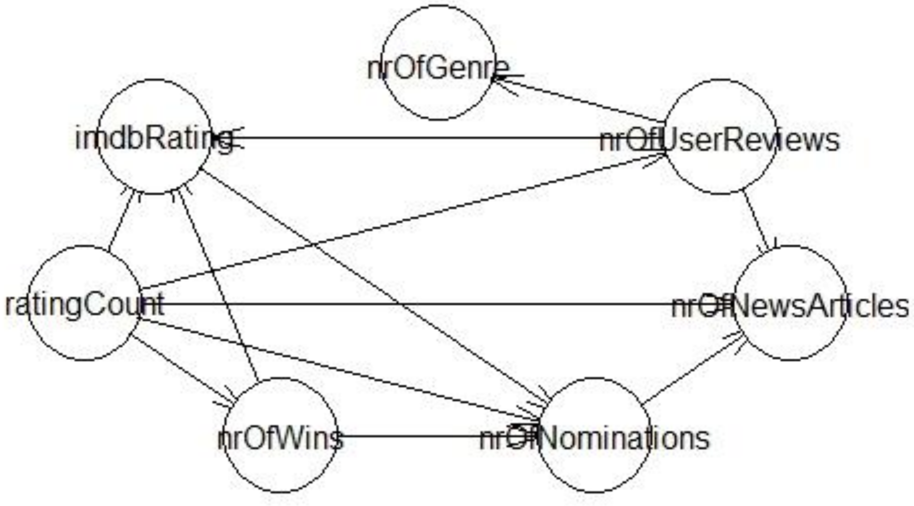
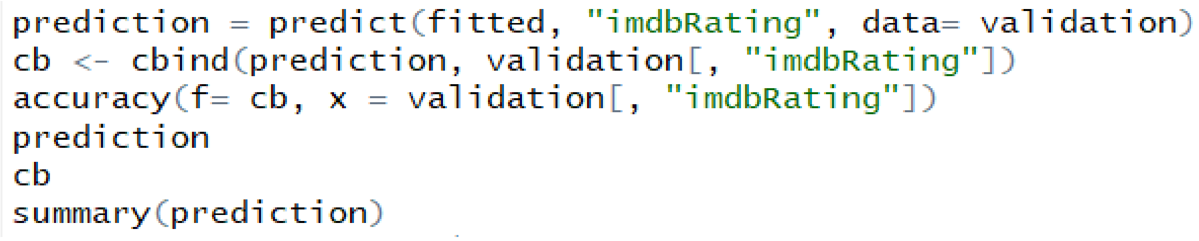


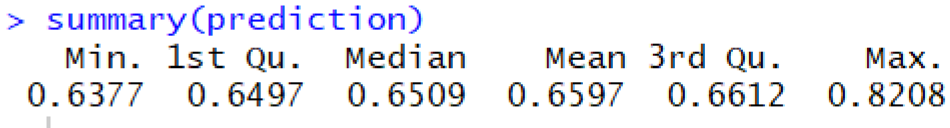
Figure 17: Bayesian network Diagram

The above figure17: represents the conditional probability of IMDB movie rating and the other continuous variables how they are connected to each and produce the output values.

Step4: calculate the prediction on the fitted model and find the accuracy of the model. The predicted values can be visualized by using a function



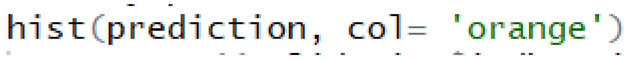
```
prediction = predict(fitted, "imdbRating", data= validation)
cb <- cbind(prediction, validation[, "imdbRating"])
accuracy(f= cb, x = validation[, "imdbRating"])
prediction
cb
summary(prediction)
```



```
> summary(prediction)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6377  0.6497  0.6509  0.6597  0.6612  0.8208 
```

Step5:

The below figure shows the histogram of the predicted values.



```
hist(prediction, col= 'orange')
```

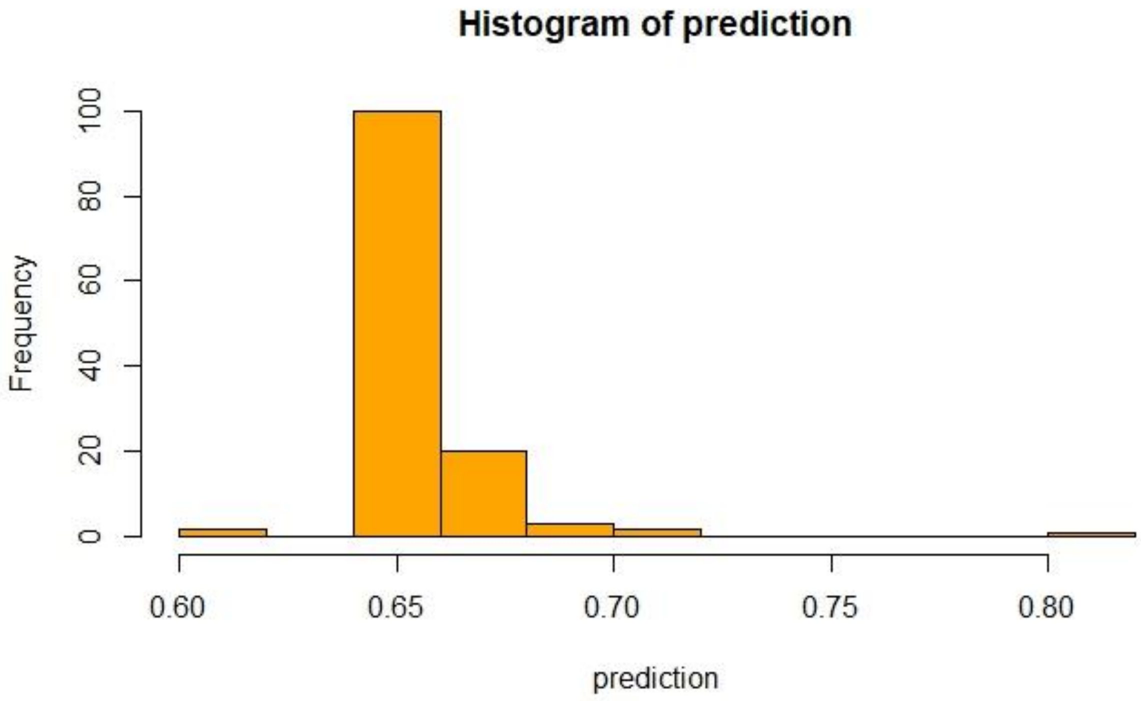
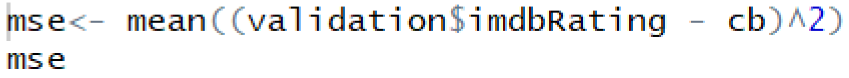
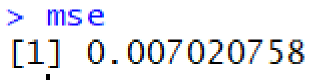


Figure 18: histogram for Bayesian network

Step5: once, we get the actual predicted values calculate the mean square error with the actual values to find the actual predicted values.



```
mse<- mean((validation$imdbRating - cb)^2)
mse
```



```
> mse
[1] 0.007020758
```

Therefore, the average prediction value for the Bayesian method is 0.6 which is similar to the artificial neural network. But the error rate is less compared to neural network and high compared to linear regression. The error for Bayesian method is 0.07020 and the accuracy of the model is 60%. The below plot visualizes the results for Bayesian method.

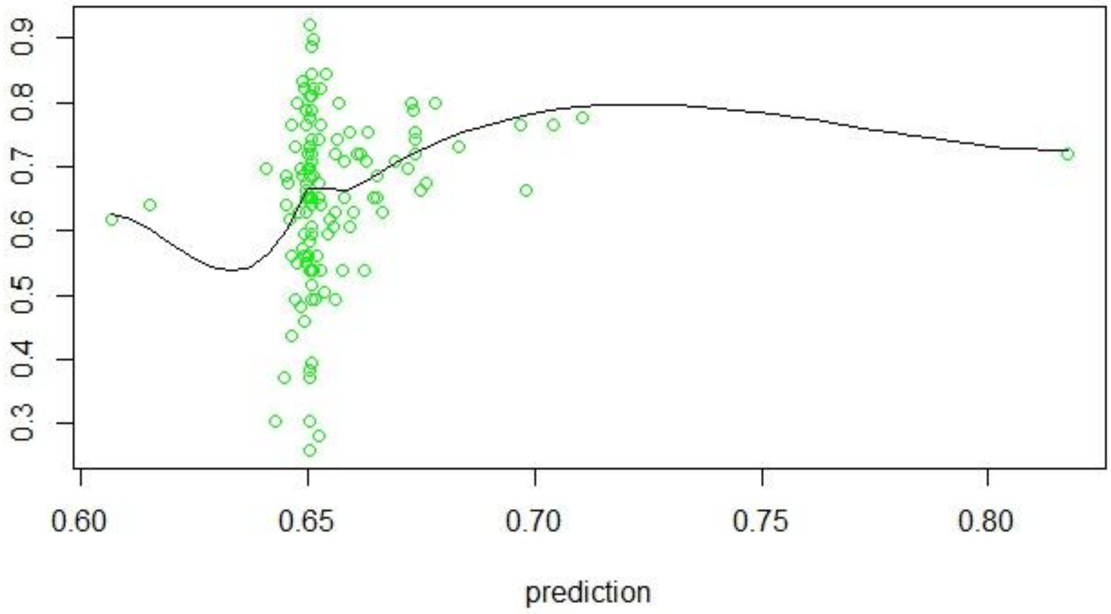


Figure 19: predicted values for Bayesian network

6. Results:

Finally, the results we find out using the three predictions methods are as follows: linear regression was about 86% accuracy and the error rate was less than 10%, artificial neural network was about 67% accuracy and error rate was 20% and Bayesian method was about 60%accuracy and error rate was 7%.

	Linear regression	Artificial neural network	Bayesian method
Mse	0.01576	2.427057	0.8002252
Predicted values	0.7	0.6	0.6
Success rate	86%	67%	60%

7. **Conclusion**:

This paper is all about predicting the IMDB movie rating. We performed data analysis which helps us improving the predicting strategy. After implementing the three prediction models. Linear regression represents the more accurately. And our results show that we successfully attained the good prediction rating. In future work would like to work on the comparing the more prediction models with linear regression.

8.Software Ethics Our Project:

1. As a software engineers our team will act consistently with the public interest.

2. To be sincere and honest to state claims or estimates based on the available data.

3. As a software engineer it is necessary to make that the code developed by our team should meet the standards.

4. To be away from plagiarism.

5. Capable of managing time and reach the milestone on time.

6. Help co-workers and colleagues with professional development, and to support them in following the above code of ethics.

References:

1. Mestyán M, Yasseri T, Kertész J (2013) Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data. PLoS ONE 8(8): e71226. https://doi.org/10.1371/journal.pone.0071226

2. Prediction of Score on IMDB Movie Dataset Xueshi Hou A53093396 University of California, San Diego x7hou@ucsd.edu

3. Darin Im, Minh Thao, Dang Nguyen, "Predicting Movie Success in the U.S. market," Dept.Elect.Eng, Stanford Univ., California, December, 2011

4. Sagar V. Mehta, Rose Marie Philip, Aju Talappillil Scaria, "Predicting Movie Rating based on Text Reviews," Dept.Elect.Eng, Stanford Univ., California, December, 2011

5. Deniz Demir, Olga Kapralova, Hongze Lai, "Predicting IMDB Movie Ratings Using Google Trends," Dept.Elect.Eng,Stanford Univ., California, December, 2012

6. Malik, J. S., Goyal, P., & Sharma, A. K. (2010). A Comprehensive Approach Towards Data Preprocessing Techniques & Association Rules. In Proceedings of The4th National Conference

7. A data mining approach to analysis and prediction of movie ratings M. Saraee, S. White & J. Eccleston University of Salford, England

8. Han, J., Kamber, M., Data Mining Concepts and Techniques, Morgan Kaufmann Publishers: San Francisco, 2001.

9. https://www.tutorialspoint.com/r/r_overview.htm

10. https://www.r-project.org/

Appendix:

```
library(caret)

library(plot3D)

library(RColorBrewer)

library(alr3)

library(corrplot)

library(neuralnet)

library(bnlearn)

dataset <- read.csv("C:/users/karht/videos/grad project papers/movie1.csv")

head(dataset)

summary(dataset)

o=corrplot(cor(dataset), method = "number")

apply(dataset,2,range)

maxvalue<- apply(dataset,2,max)

minvalue<- apply(dataset,2,min)

dataset<- as.data.frame(scale(dataset,center = minvalue, scale = maxvalue - minvalue))

validation_index<- createDataPartition(dataset$imdbRating, p=0.99, list=FALSE)

validation<- dataset[-validation_index, ]

dataset<- dataset[validation_index, ]

index= sample(1:nrow(dataset), size= 0.60*nrow(dataset))

train= dataset[index, ]

test= dataset[-index, ]

x<- train[,1:7]

par(mfrow = c(1,4))

for(i in 1)
```

```
{
 boxplot(x[,i], col = "green", main = names(train[i]))
 boxplot(x[,i], col= "red", main = names(train[i]))
 boxplot(x[,i], col= "blue", main = names(train[i]))
 boxplot(x[,i], col= "yellow", main = names(train[i]))
 boxplot(x[,i], col= "purple", main = names(train[i]))
 boxplot(x[,i], col= "white", main = names(train[i]))
 boxplot(x[,i], col= "orange", main = names(train[i]))
}
par(mfrow = c(1,2))
hist(train$imdbRating, col= blues9)
x<- train[,1]
y<-train[,2]
z<- train[,3]
plot(y)
scatter3D(x,y,z, clab= c("imdbrating", "width (ratingCount)"))
x<- [,1:7]
par(mfrow = c(1,1))
for(i in 7)
{
 boxplot(x[,i], col = "green", main = names(test[i]))
 boxplot(x[,i], col= "red", main = names(test[i]))
 boxplot(x[,i], col= "blue", main = names(test[i]))
 boxplot(x[,i], col= "yellow", main = names(test[i]))
 boxplot(x[,i], col= "purple", main = names(test[i]))
```

```
  boxplot(x[,i], col= "white", main = names(test[i]))
  boxplot(x[,i], col= "purple", main = names(test[i]))
}
hist(test$imdbRating, col= blues9)
summary(test)
print(summary(test))
model<- lm(imdbRating~., data= train)
summary(model)
prediction<- predict(model, data= test)
predictions
plot(predictions, col= blues9)
summary(predictions)
# linear regrission
fit<-lm(imdbRating ~., data= test)
summary(fit)
predictions<- predict(fit, validation)
predictions
actuals_preds <- data.frame(cbind(actuals=validation$imdbRating, predictions=predictions))
correlation_accuracy <- cor(actuals_preds)
correlation_accuracy
head(actuals_preds)
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
min_max_accuracy
mse<- mean((validation$imdbRating - predictions)^2)
```

```
print(mse)
plot(actuals_preds, col= "blue")
abline()
hist(predictions, col= 'pink')
x<- predictions
y<- validation[,1]
scatter.smooth(x,y, ylim= NULL, col= "red")
summary(actuals_preds)
# artificial neural network
set.seed(100)
allvars<- colnames(dataset)
predictorvars<- allvars[!allvars%in%"imdbRating"]
predictorvars<- paste(predictorvars, collapse = "+")
form= as.formula(paste("imdbRating~", predictorvars, collapse = "+"))
neuralmodel<- neuralnet(formula =form, hidden = c(5), linear.output = T, data = test, stepmax = 1e+05)
summary(neuralmodel)
plot(neuralmodel)
prediction<- compute(neuralmodel, validation[,2:7])
str(prediction)
prediction<- prediction$net.result * (max(validation$imdbRating) - min(validation$imdbRating)) + min(validation$imdbRating)
prediction
hist(prediction, col= blues9)
actualvalues<- (validation$imbdRating) * (max(validation$imdbRating)- min(validation$imdbRating)) + min(validation$imdbRating)
```

```
mse<- sum((prediction - validation )^2)/nrow(validation)
mse
plot(prediction, col = 'blue', main = 'real vs predicted')
abline(test)
x<- prediction
y<- validation[,2]
scatter.smooth(x,y, ylim= NULL, col= "red")
print(mse)
# bayesian method
res = hc(test)
res
plot(res)
fitted = bn.fit(res, data= test)
fitted
prediction = predict(fitted, "imdbRating", data= validation)
cb <- cbind(prediction, validation[, "imdbRating"])
accuracy(f= cb, x = validation[, "imdbRating"])
prediction
cb
summary(prediction)
hist(prediction, col= 'orange')
mse<- mean((validation$imdbRating - cb)^2)
mse
plot(cb, col = blues9)
x<- cb
```

```
y<- validation$imdbrating
scatter.smooth(x, y, ylim= NULL, col= 'green')
```