Shree Rahul Education Society's (Regd.)
SHREE L. R. TIWARI COLLEGE OF ENGINEERING
Kanakia Park, Near Commissioner's Bungalow, Mira Road (East), Thane 401107, Maharashtra
(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)
NAAC Accredited | ISO 9001:2015 Certified
Tel. No.: 022-28120144 / 022-28120145 | Email: slrtce@rahuleducation.com | Website: www.slrtce.in

Movies On OTT Analysis

BACHELOR OF ENGINEERING
IN INFORMATION TECHNOLOGY

By

Devesh Shetty (Roll no. 68)

Under the Guidance of

Mrs. Deepali Patil

Assistant Professor, Department of Information Technology

Shree Rahul Education Society's (Regd.)
SHREE L.R. TIWARI College of Engineering
(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)
ISO 9001:2008 Certified.

Near Commissioner's Bungalow, Kanakia Park, Mira Road (E), Thane-401107, Maharashtra.

Department of Information Technology 2020-21

Shree Rahul Education Society's (Regd.)
SHREE L. R. TIWARI COLLEGE OF ENGINEERING
Kanakia Park, Near Commissioner's Bungalow, Mira Road (East), Thane 401107, Maharashtra
(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)
NAAC Accredited | ISO 9001:2015 Certified
Tel. No.: 022-28120144 / 022-28120145 | Email: slrtce@rahuleducation.com | Website: www.slrtce.in

A Mini Project Report On

Movies On OTT Analysis

Submitted to Mumbai University

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN INFORMATION TECHNOLOGY

By

Devesh Shetty (Roll no. 68)

Under The Guidance Of

Mrs. Deepali Patil

Assistant Professor, Department of Information Technology

Shree Rahul Education Society's (Regd.)
SHREE L.R. TIWARI
College of Engineering
(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)
ISO 9001:2008 Certified.

Near Commissioner's Bungalow, Kanakia Park, Mira Road (E), Thane-401107, Maharashtra.

Department of Information Technology 2020-21

Shree Rahul Education Society's (Regd.)
SHREE L. R. TIWARI COLLEGE OF ENGINEERING
Kanakia Park, Near Commissioner's Bungalow, Mira Road (East), Thane 401107, Maharashtra
(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)
NAAC Accredited | ISO 9001:2015 Certified
Tel. No.: 022-28120144 / 022-28120145 | Email: slrtce@rahuleducation.com | Website: www.slrtce.in

UNIVERSITY OF MUMBAI

CERTIFICATE

This is to certify that the project titled "Movies On OTT Analysis" has been completed under our supervision and guidance by the following students:

Devesh Shetty

In the partial fulfillment of degree of Bachelor of Engineering in Information Technology branch as prescribed by the University of Mumbai during the academic year 2020-2021. The said work has been assessed and is found to be satisfactory.

Signature of the Internal Examiner

Name: Prof. Deepali Patil

Date: _____

Signature of the External Examiner

Name: _____

Date: _____

Signature of the H.O.D.

Name: Prof. Sunil Yadav

Date: _____

Signature of the Principal

Name: Dr. S. Ram Reddy

Date: _____

DECLARATION

We do hereby declare that the work embodied in the project entitled "**Movies On OTT Analysis"** is the outcome of our original work under the guidance and supervision of **Prof. Deepali Patil**. This piece of work or any part of it has not been submitted previously for the award of any other degree, diploma, or other title to any other institution.

We also declare that this written submission represents our ideas in our own words and where others ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: May, 2021

Devesh Shetty
Roll No.: 68
Exam. Seat No.:

ACKNOWLEDGEMENT

A few sublime human experiences defy expressions of any kind, and a feeling of true gratitude is one of them. I, therefore, find words quite inadequate to express my indebtedness to my Guide **Prof. Deepali Patil** for their virtuous guidance, encouragement and help throughout this work. Their deep insight into the problem and the ability to provide solutions has been immense value in improving the quality of project at all stages. This experience of working with them shall ever remain a source of inspiration and encouragement for me.

I express my thanks to **Prof. Sunil Yadav**, HOD (IT), SLRTCE, Mira Road, for extending his support that he gave truly help the progression of the project work.

My sincere thanks to **Dr. S. Ram Reddy**, Principal, SLRTCE, Mira Road for providing me the necessary administrative assistance in the completion of the work.

I am extremely grateful to the celebrated authors whose precious works have been consulted and referred in my project work. I also wish to convey my appreciation to my friends who provided encouragement and timely support in the hour of need.

Special thanks to my Parents whose love and affectionate blessings have been a constant source of inspiration in making this a reality.

All the thanks are, however, only fraction of what is due to Almighty for granting me an opportunity and the divine grace to successfully accomplish this assignment.

Devesh Shetty

ABSTRACT

Movies are considered to be an important art forms, a world wide source of entertainment, and a powerful medium for educating or indoctrinating citizens. As far as the current pandemic situation is concerned, OTT platforms act as one of the most entertaining factors and a significant stress reliever for people around the globe.

This project aims to explore all the movies in popular OTT platforms, in order to gain interesting insights. This is carried out with the aid of a Kaggle dataset, collected from Netflix, Prime Video, Hulu and Disney+ API.

Dataset contains the complete information of all the movies, their ratings and the corresponding OTT platforms in which they are available. It provides detailed information such as Year of release, Genre, IMDb rating, Director and the Language of each movie.

Here in this project we are using Multiple linear regression and Random forest to analyse our data and to get meaning full insights from the data collected from different OTT platforms collected from Netflix, Prime Video, Hulu and Disney+. Furthermore, the result obtained from each of these algorithms are compared to understand their respective suitability under varied conditions

Keywords: Multiple linear regression, Random forest Algorithms, OTT Platform Analysis, Netflix, Disney, Prime Video, Hulu , R programming.

LIST OF FIGURES

Chapter 1

Introduction

Chapter 1: Introduction

This chapter will introduce the reader with Movies on OTT Analysis System. It will light up the topics like the description of the project and the former formulation of the problem behind it as well as what motivated the makers of the project to take a decision to make this project and its related problem solution and thus covering up the scope of the project.

1.1 Description

Traditionally TV has been the source of entertainment along with recording it within CDs ,after the boom of technology and internet there have been many OTT (Over The Top) apps available in the market which has now a days almost replaced the traditional ways of consuming content, OTT apps like Netflix, Prime Video, Disney Plus, Hulu have emerged and are preferred more due to their ease and self paced content consumption.

Here the data is analysed and some visualization and manipulation are carried to get a more precise and a graphical picture of the entire dataset and this will help to shed views of a particular product or topic. If people find topics relevant or interesting, then they would desire to share their opinion about the topic. The topic could be a product or any other object. Understanding this can help us decide and OTT platforms the type of movies that are popular among the people rather it be depending on reviews, rating or age group.

In this project we are also using algorithms like Multiple linear regression and Random forest to analyse our data and to get meaning full insights from the data collected from different OTT platforms collected from Netflix, Prime Video, Hulu and Disney+. Furthermore, the result obtained from each of these algorithms are compared to understand their respective suitability under varied conditions

1.2 Problem Formulation

These days, technology has got its new and higher pace. This development has changed human's way of watching or consuming content now people can also rate the content or movie according to their views about the movies their opinions, sentiments and views can be expressed on various platforms like IMDb, Rotten Tomatoes in which they do so these review and rating platforms are nothing but a way through which people express their thoughts or views regarding the movie. The platforms like IMDb, Rotten Tomatoes will help people to express and others to judge the movie based on the review of people. So to check how ratings are affected by other factors like runtime, directors etc. there is a need of data which is required to basically process and analyse the data based

On the dataset obtained from kaggle we can further process the data depending upon the data we can find possible ways to do this. R is an open-source approach used for analyzing on-line reviews to perform analysis and visualization on our data.

1.3 Motivation

Comments, reviews, and opinion of the people about the movie play an important role to determine whether a given population is satisfied with the movie or not. It helps in predicting the popularity of the movie among a wide range and a wide variety of people on a particular event of interest like the review of a movie roaming around the world. These data are essential for OTT platforms to perform analysis. This analysis can help us and the OTT platforms to perform and improve on various aspects like movie recommendation or to judge the type of movie that they push on their platforms, the popularity of the movie and so on. This analysis can help the OTT platforms to cater a new range of audience and will help their audience to get better experience ,better movies and will also help them to decide which genres movie will work the best among the audiance. Twitter generates huge data that cannot be handled manually to extract some useful information and therefore, the ingredients of automatic classification are required to handle those data. This gain interesting insights. This is carried out with the aid of a Kaggle dataset, collected from Netflix, Prime Video, Hulu and Disney+ API. This interesting insights will help us better understand the current situation or trends of movies that are extremely popular among the OTT platforms.

1.4 Proposed Solution

OTT platforms like Netflix, Prime Video, Hulu and Disney+ has acquired immense popularity and interest with the people around the world. IMDb and Rotten Tomatoes is one of the effective sites for any OTTs to acquire intelligence to get information about which movies people are talking and which OTT platform is popular in the world. Popular movies helps to engage the users and directly communicates with them and in response, users to provide word-of-mouth marketing for the movie which in turn helps these platforms cater more audiance. With the limited resources and knowing about no one can target directly to the destination consumers, the business intelligence can be more efficient in their policy of marketing by being very selective about consumers choice they should reach out to. The proposed solution is to perform analysis on movie data present on different ott platforms. Analysis can be performed using a number of Machine Learning Algorithms. This project leverages the concepts of Multiple linear regression and Random forest algorithm to achieve the results. Furthermore, the result obtained from each of these algorithms are compared to understand their respective suitability under varied conditions.

1.5 Scope

Analyzing the Movie data is important for many applications such as OTT platforms trying to find out the response of their movies in the market, predicting popularity of the movie. Pushing movies on platforms based on the current trends in the market.

This Analysis on Movie Dataset has a number of applications:

Business: OTT platforms like Netflix, Prime Video, Hulu and Disney+ they all use this data to judge the popularity of the movie based on the current trends and thus helps them get insights and push such movies.

Recommendation: Recommendation is one of the most important and popular techniques to engage users with their content and this is what OTT platforms look for and through this insights they can easily help them recommend.

Reviews :These reviews and rating can help us understand the how the movie is and we get to decide whether to go for it or not.

Chapter 2

Review of Literature

Chapter 2: Review of Literature

Here we will elaborate the aspects like the literature survey of the project and what all projects are existing and been used in the market which the makers of this project took the inspiration from and thus decided to go ahead with the project covering with the problem statement. Literature review helps us analyze the past innovations related to the project and also help ameliorate them.

2.1 Literature Survey

[1] this paper Movies Reviews Sentiment Analysis and Classification the authors main focus is how sentimental is performed here goal of this work is to address SA by constructing an approach that can classify movie reviews and then compare the results in an inclusive study of eight wellknown classifiers. To evaluate the proposed model, IMDB reviews real dataset was utilized. Tokenization was applied on the dataset to transfer strings into word vector, then stemming was used to extract the root of the words, afterwards gain ratio was applied on the dataset as an attribute selection algorithm. Then, the data was split into training and testing datasets using the percentages 66%, 34% respectively. In order to compare the eight different classifiers, five different evaluation metrics are utilized. The results show that Random Forest outperforms the other classifiers. Furthermore, Ripper Rule Learning performed the worst on the dataset according to the results attained from the evaluation metrics.

[2] Here we followed paper based on The Performance Comparison of Multiple Linear Regression, Random Forest the paper gave us an detailed classification about the processes . For comparison of there are several data mining techniques, the power production data from a Photovoltaic Module was used in the research. In this study, the model was constituted from seven variables. The highest correlation coefficient was obtained in Artificial Neural Network architecture (R = 0.997). The study by author also showed the importance of data mining method. If this study had been evaluated by MLR then the findings of the study would have been obtained biased and non-robust. So, a study must be evaluated by robust statistical methods in order to estimate a model in a high accuracy rate. This study showed that the MLP-ANN architecture has the best performance when compared with MLR and RF

2.2 Problem Statement

Comments, reviews, and opinion of the people about the movie play an important role to determine whether a given population is satisfied with the movie or not. It helps in predicting the popularity of the movie among a wide range and a wide variety of people on a particular event of interest like the review of a movie roaming around the world. These data are essential for OTT platforms to perform analysis.

This project leverages the concepts of Multiple linear regression and Random forest algorithm to achieve the results. Furthermore, the result obtained from each of these algorithms are compared to understand their respective suitability under varied conditions.

Chapter 3

Description

Chapter 3: Description

This chapter enlightens about the theoretical explanation of the concepts used in the implementation of the project such as data preprocessing, algorithms used, and performance measure followed by the design consideration details.

3.1 Theory

Movies on OTT analysis is an interesting idea to get insights from the data and to analyse the trends around the world this trends are mostly related to the ratings , popularity of platforms the movies liked by the different age group of people these trends helps us better understand the platform and reveling meaning full insights

For data manipulation we have used the different libraries such as dplyr and for data visualization we have used libraries such as ggplot and some of the built in libraries. R and Python are widely used for analysis on dataset. Analysis of data is now much more than a college project or a certification program. A good number of Tutorials related to Analysis are available for educating students on the analysis project report and its usage with R and Python.

3.1.1 Data Pre-processing

Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Steps Involved in Data Pre-processing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a). Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

1. Removing NA value:

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a table.

2. Filling or Skipping missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value. Or we can skip such values

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
2. **Attribute Selection:**
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
3. **Discretization:**
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.
4. **Concept Hierarchy Generation:**
   Here attributes are converted from level to higher level in hierarchy. For Example-The attribute "city" can be converted to "country".

3. Data Reduction:

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we use data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

1. **Data Cube Aggregation:**
   Aggregation operation is applied to data for the construction of the data cube.
2. **Attribute Subset Selection:**
   The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute. The attribute having p-value greater than significance level can be discarded.
3. **Numerosity Reduction:**
   This enables to store the model of data instead of whole data, for example: Regression Models.

3.1.2 Multiple linear regression

Regression models are used to describe relationships between variables by fitting a line to the observed data. Regression allows you to estimate how a dependent variable changes as the independent variable(s) change.

Multiple linear regression is used to estimate the relationship between two or more independent variables and one dependent variable. You can use multiple linear regression when you want to know:

1. How strong the relationship is between two or more independent variables and one dependent variable (e.g. how rainfall, temperature, and amount of fertilizer added affect crop growth).

2. The value of the dependent variable at a certain value of the independent variables (e.g. the expected yield of a crop at certain levels of rainfall, temperature, and fertilizer addition).

Multiple linear regression formula

The formula for a multiple linear regression is:

$$y = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n + \varepsilon$$

- **y** = the predicted value of the dependent variable
- $\mathbf{B_0}$ = the y-intercept (value of y when all other parameters are set to 0)
- $\mathbf{B_1X_1}$= the regression coefficient (B_1) of the first independent variable ($\mathbf{X_1}$) (a.k.a. the effect that increasing the value of the independent variable has on the predicted **y** value)
- **…** = do the same for however many independent variables you are testing
- $\mathbf{B_nX_n}$ = the regression coefficient of the last independent variable
- **e** = model error (a.k.a. how much variation there is in our estimate of **y**)

To find the best-fit line for each independent variable, multiple linear regression calculates three things:

- The regression coefficients that lead to the smallest overall model error.
- The *t*-statistic of the overall model.
- The associated *p*-value (how likely it is that the t-statistic would have occurred by chance if the null hypothesis of no relationship between the independent and dependent variables was true).

It then calculates the *t*-statistic and *p*-value for each regression coefficient in the model.

Example

Multiple linear regression in R using heart data set:

	B	C	D
	biking	smoking	heart.disease
1	30.80125	10.89661	11.76942
2	65.12922	2.219563	2.854081
3	1.959665	17.58833	17.1778
4	44.8002	2.802559	6.816647
5	69.42845	15.9745	4.062224
6	54.40363	29.33318	9.550046
7	49.05616	9.060846	7.624507
8	4.784604	12.83502	15.85465

While it is possible to do multiple linear regression by hand, it is much more commonly done via statistical software. We are going to use R for our examples because it is free, powerful, and widely available. Download the sample dataset to try it yourself.

Load the heart.data dataset into your R environment and run the following code:

R code for multiple linear regressionheart.disease.lm<-lm(heart.disease ~ biking + smoking, data = heart.data)

This code takes the data set heart.data and calculates the effect that the independent variables biking and smoking have on the dependent variable heart disease using the equation for the linear model: lm().

Interpreting the results

To view the results of the model, you can use the summary() function:

summary(heart.disease.lm)
This function takes the most important parameters from the linear model and puts them into a table that looks like this:

```
Call:
lm(formula = heart.disease ~ biking + smoking, data = heart.data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1789 -0.4463  0.0362  0.4422  1.9331

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.984658   0.080137  186.99   <2e-16 ***
biking      -0.200133   0.001366 -146.53   <2e-16 ***
smoking      0.178334   0.003539   50.39   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.654 on 495 degrees of freedom
Multiple R-squared:  0.9796,  Adjusted R-squared:  0.9795
F-statistic: 1.19e+04 on 2 and 495 DF,  p-value: < 2.2e-16
```

The summary first prints out the formula ('Call'), then the model residuals ('Residuals'). If the residuals are roughly centered around zero and with similar spread on either side, as these do (median 0.03, and min and max around -2 and 2) then the model probably fits the assumption of heteroscedasticity.

Next are the regression coefficients of the model ('Coefficients'). Row 1 of the coefficients table is labeled (Intercept) – this is the y-intercept of the regression equation. It's helpful to know the estimated intercept in order to plug it into the regression equation and predict values of the dependent variable:

heart disease = 15 + (-0.2*biking) + (0.178*smoking) ± e
The most important things to note in this output table are the next two tables – the estimates for the independent variables.

The Estimate column is the estimated **effect**, also called the **regression coefficient** or r^2 value. The estimates in the table tell us that for every one percent increase in biking to work there is an associated 0.2 percent decrease in heart disease, and that for every one percent increase in smoking there is an associated .17 percent increase in heart disease.

The Std.error column displays the **standard error** of the estimate. This number shows how much variation there is around the estimates of the regression coefficient.

The t value column displays the **test statistic**. Unless otherwise specified, the test statistic used in linear regression is the *t*-value from a two-sided t-test. The larger the test statistic, the less likely it is that the results occurred by chance.

The Pr(> | t |) column shows the ***p*-value**. This shows how likely the calculated *t*-value would have occurred by chance if the null hypothesis of no effect of the parameter were true.

Because these values are so low ($p < 0.001$ in both cases), we can **reject the null hypothesis** and conclude that both biking to work and smoking both likely influence rates of heart disease.

Presenting the results

When reporting your results, include the estimated effect (i.e. the regression coefficient), the standard error of the estimate, and the *p*-value. You should also interpret your numbers to make it clear to your readers what the regression coefficient means.

In our survey of 500 towns, we found significant relationships between the frequency of biking to work and the frequency of heart disease and the frequency of smoking and frequency of heart disease ($p < 0.001$ for each). Specifically we found a 0.2% decrease (± 0.0014) in the frequency of heart disease for every 1% increase in biking, and a 0.178% increase (± 0.0035) in the frequency of heart disease for every 1% increase in smoking.

Visualizing the results in a graph

It can also be helpful to include a graph with your results. Multiple linear regression is somewhat more complicated than simple linear regression, because there are more parameters than will fit on a two-dimensional plot.

However, there are ways to display your results that include the effects of multiple independent variables on the dependent variable, even though only one independent variable can actually be plotted on the x-axis.

Here, we have calculated the predicted values of the dependent variable (heart disease) across the full range of observed values for the percentage of people biking to work.

To include the effect of smoking on the independent variable, we calculated these predicted values while holding smoking constant at the minimum, mean, and maximum observed rates of smoking.

3.1.3 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting.

"The spirit is willing, but the flesh is weak."
Here is the result when the sentence was translated to Russian and back to English:
"The vodka is good, but the meat is rotten."

The below diagram explains the working of the Random Forest algorithm:

3.1.4 Performance Measure (Confusion Matrix)

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.
Well, it is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve.

Let's understand TP, FP, FN, TN in terms of pregnancy analogy.

True Positive:

Interpretation: You predicted positive and it's true.

You predicted that a woman is pregnant, and she actually is.

True Negative:

Interpretation: You predicted negative and it's true.

You predicted that a man is not pregnant, and he actually is not.

False Positive: (Type 1 Error)

Interpretation: You predicted positive and it's false.

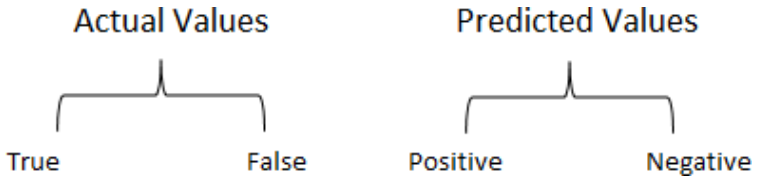
You predicted that a man is pregnant, but he actually is not.

False Negative: (Type 2 Error)

Interpretation: You predicted negative and it's false.

You predicted that a woman is not pregnant, but she actually is.

Just Remember, we describe predicted values as Positive and Negative and actual values as True and False.



Chapter 4

Implementation and results

Chapter 4: Implementation and Results

In this chapter the Implementation details of the project would be shed light upon including the lines of code executed to achieve the results.

4.1 Important Necessary libraries

```
install.packages("tidyverse")
install.packages("ggplot")
install.packages("plotly")
install.packages('tidytext')
install.packages('caret')
install.packages('caTools')
install.packages('randomForest')

library("tidyverse")
library("plotly")
library('tidytext')
library('caret')
library('caTools')
library('randomForest')
```

- **Tidyverse**: Tidyverse is a collection of essential R packages for data science. The packages under the tidyverse umbrella help us in performing and interacting with the data. There are a whole host of things you can do with your data, such as subsetting, transforming, visualizing, etc. Dplyr and ggplot are 2 of the most important packages which are inside tidyverse.

- **Plotly**: Plotly's R graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, and 3D (WebGL based) charts

- **Caret:** The **caret** package (short for Classification And REgression Training) contains functions to streamline the model training process for complex regression and classification problems. caret loads packages as needed and assumes that they are installed. If a modeling package is missing, there is a prompt to install it.

- **Tidytext**: Using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use. Much of the infrastructure needed for text mining with tidy data frames already exists in packages like dplyr, broom, tidyr and ggplot2. In this package, we provide functions and supporting data sets to allow conversion of text to and from tidy formats, and to switch seamlessly between tidy tools and existing text mining packages.

- **CaTools**: Contains several basic utility functions including: moving (rolling, running) window statistic functions, read/write for GIF and ENVI binary files, fast calculation of AUC, LogitBoost classifier, base64 encoder/decoder, round-off-error-free sum and cumsum, etc.

- **randomForest:** randomForest implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

4.1.1 Installing and importing all the necessary packages

4.2 Providing dataset and reading dataset

	ID	Title	Year	Age	IMDb	Rotten To	Netflix	Hulu	Prime Vid	Disney+	Type	Directors	Genres	Country	Language	Runtime
0	1	Inception	2010	13+	8.8	87%	1	0	0	0	0	Christoph	Action,Ad	United Sta	English,Ja	148
1	2	The Matri	1999	18+	8.7	87%	1	0	0	0	0	Lana Wach	Action,Sci	United Sta	English	136
2	3	Avengers:	2018	13+	8.5	84%	1	0	0	0	0	Anthony F	Action,Ad	United Sta	English	149
3	4	Back to th	1985	7+	8.5	96%	1	0	0	0	0	Robert Ze	Adventur	United Sta	English	116
4	5	The Good,	1966	18+	8.8	97%	1	0	1	0	0	Sergio Lec	Western	Italy, Spair	Italian	161
5	6	Spider-Mi	2018	7+	8.4	97%	1	0	0	0	0	Bob Persi	Animatior	United Sta	English,Sp	117
6	7	The Pianis	2002	18+	8.5	95%	1	0	1	0	0	Roman Po	Biography	United Kir	English,Ge	150
7	8	Django Ur	2012	18+	8.4	87%	1	0	0	0	0	Quentin T	Drama,We	United Sta	English,Ge	165
8	9	Raiders of	1981	7+	8.4	95%	1	0	0	0	0	Steven Sp	Action,Ad	United Sta	English,Ge	115
9	10	Inglourio	2009	18+	8.3	89%	1	0	0	0	0	Quentin T	Adventur	Germany,	English,Ge	153
10	11	Taxi Drive	1976	18+	8.3	95%	1	0	0	0	0	Martin Sc	Crime,Dra	United Sta	English,Sp	114
11	12	3 Idiots	2009	13+	8.4	100%	1	0	1	0	0	Rajkumar	Comedy,C	India	Hindi,Engl	170
12	13	Pan's Laby	2006	18+	8.2	95%	1	0	0	0	0	Guillermo	Drama,Fa	Mexico,Sp	Spanish	118
13	14	Room	2015	18+	8.1	93%	1	0	0	0	0	Lenny Abr	Drama,Thi	Ireland,Ca	English	118
14	15	Monty Pyt	1975	7+	8.2	97%	1	0	0	0	0	Terry Gilli	Adventur	United Kir	English,Fr	91
15	16	Once Upo	1968	13+	8.5	95%	1	0	1	0	0	Sergio Lec	Western	Italy, Unite	Italian,En	165
16	17	Indiana Jo	1989	13+	8.2	88%	1	0	0	0	0	Steven Sp	Action,Ad	United Sta	English,Ge	127
17	18	Groundho	1993	7+	8	96%	1	0	0	0	0	Harold Rai	Comedy,F	United Sta	English,Fr	101
18	19	The King's	2010	18+	8	95%	1	0	0	0	0	Tom Hoop	Biography	United Kir	English	118
19	20	Her	2013	18+	8	95%	1	0	0	0	0	Spike Jon	Drama,Ro	United Sta	English	126
20	21	There Wil	2007	18+	8.2	91%	1	0	0	0	0	Paul Thor	Drama	United Sta	English,Ar	158
21	22	The Social	2010	13+	7.7	96%	1	0	0	0	0	David Finc	Biography	United Sta	English,Fr	120

Fig 4.1 Dataset

Read Data from csv:

To read data from csv file we can use read.csv

dp1 = read.csv("E:\\ikeee\\A_Rtut Important\\D+\\MoviesOnStreamingPlatforms_updated.csv", stringsAsFactors = FALSE)

glimpse(dp1):

```
> glimpse(dp1)
Rows: 16,744
Columns: 17
$ X               <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ~
$ ID              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
$ Title           <chr> "Inception", "The Matrix", "Avengers: Infinity War", "~
$ Year            <int> 2010, 1999, 2018, 1985, 1966, 2018, 2002, 2012, 1981, ~
$ Age             <chr> "13+", "18+", "13+", "7+", "18+", "7+", "18+", "18+", ~
$ IMDb            <dbl> 8.8, 8.7, 8.5, 8.5, 8.8, 8.4, 8.5, 8.4, 8.4, 8.3, 8.3,~
$ Rotten.Tomatoes <chr> "87%", "87%", "84%", "96%", "97%", "97%", "95%", "87%"~
$ Netflix         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ Hulu            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ Prime.Video     <int> 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, ~
$ Disney.         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ Type            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ Directors       <chr> "Christopher Nolan", "Lana Wachowski,Lilly Wachowski",~
$ Genres          <chr> "Action,Adventure,Sci-Fi,Thriller", "Action,Sci-Fi", "~
$ Country         <chr> "United States,United Kingdom", "United States", "Unit~
$ Language        <chr> "English,Japanese,French", "English", "English", "Engl~
$ Runtime         <int> 148, 136, 149, 116, 161, 117, 150, 165, 115, 153, 114,~
```

summary(dp1):

```
> summary(dp1)
       X                ID            Title                Year
 Min.   :    0   Min.   :    1   Length:16744       Min.   :1902
 1st Qu.: 4186   1st Qu.: 4187   Class :character   1st Qu.:2000
 Median : 8372   Median : 8372   Mode  :character   Median :2012
 Mean   : 8372   Mean   : 8372                      Mean   :2003
 3rd Qu.:12557   3rd Qu.:12558                      3rd Qu.:2016
 Max.   :16743   Max.   :16744                      Max.   :2020

     Age                IMDb       Rotten.Tomatoes       Netflix
 Length:16744       Min.   :0.000   Length:16744       Min.   :0.0000
 Class :character   1st Qu.:5.100   Class :character   1st Qu.:0.0000
 Mode  :character   Median :6.100   Mode  :character   Median :0.0000
                    Mean   :5.903                      Mean   :0.2126
                    3rd Qu.:6.900                      3rd Qu.:0.0000
                    Max.   :9.300                      Max.   :1.0000
                    NA's   :571
      Hulu          Prime.Video        Disney.             Type
 Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0
 1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0
 Median :0.00000   Median :1.0000   Median :0.00000   Median :0
 Mean   :0.05393   Mean   :0.7378   Mean   :0.03368   Mean   :0
 3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0
 Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :0

  Directors            Genres            Country           Language
 Length:16744       Length:16744       Length:16744       Length:16744
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

Data Cleaning and removing Duplicates :

Helps us to perform data cleaning operations on the data

```
dp111 = complete.cases(dp11)
dp11_11= dp11[dp111, ]
dp1= dp11_11[!duplicated(dp11_11),]
```

4.3 Data Visualization and manipulation of data:

- **Visualizing Number of tites by each country**

```
dp_country = filter(dp1, nchar(Country)>0)
dp_country %>%
 filter(!str_detect(Country,',')) %>%
 group_by(Country) %>%
 count() %>%
 arrange(desc(n)) %>%
 head(20) %>%
 ggplot() + geom_col(aes(y = reorder(Country,n), x = n)) +
 geom_label(aes(y = reorder(Country,n), x = n, label = n)) +
 labs(title = 'Approx. Number of Titles of each Country') +
 theme_minimal()
```

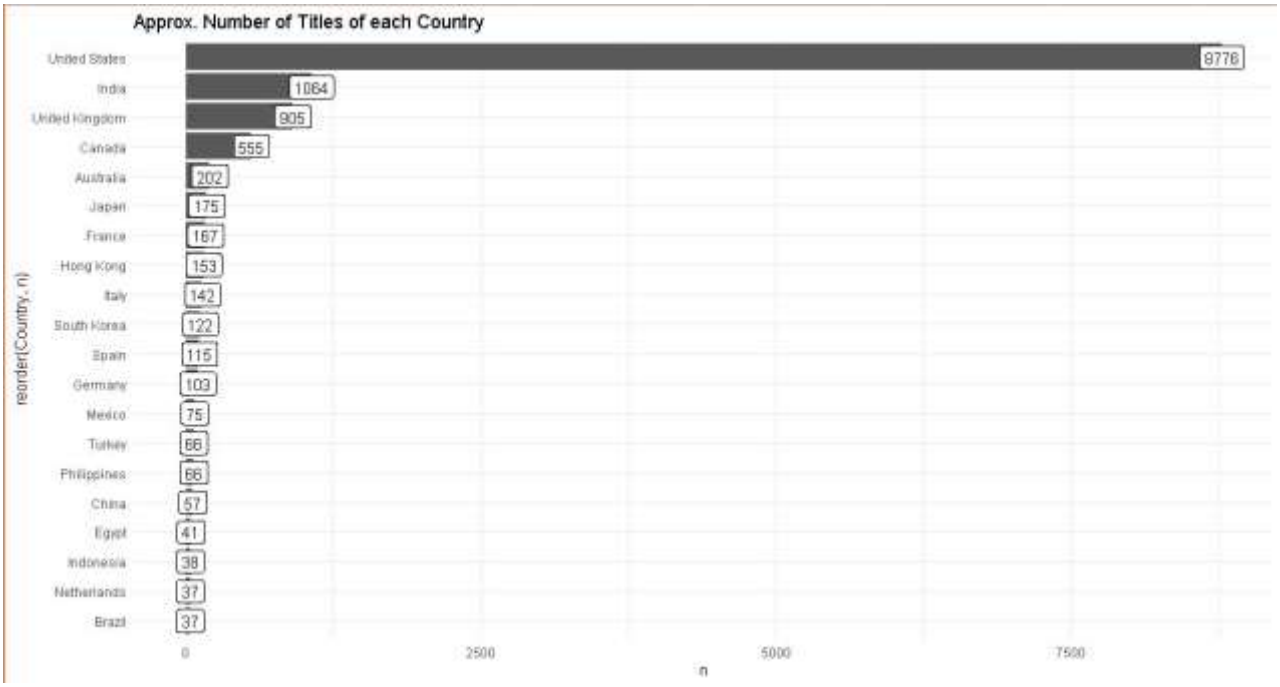


Fig 4.2 Number of Titles of each Country

- **IMDB Rating trends by year**

```
dp1 %>%
  filter(Year !=2021 && Year !='') %>%
  group_by(IMDb,Year) %>%
  count() %>%
  ggplot()+geom_line(aes(x=Year,y=n)) +
  labs(title = 'Trend of Titles every Year') +
  theme_minimal()
```

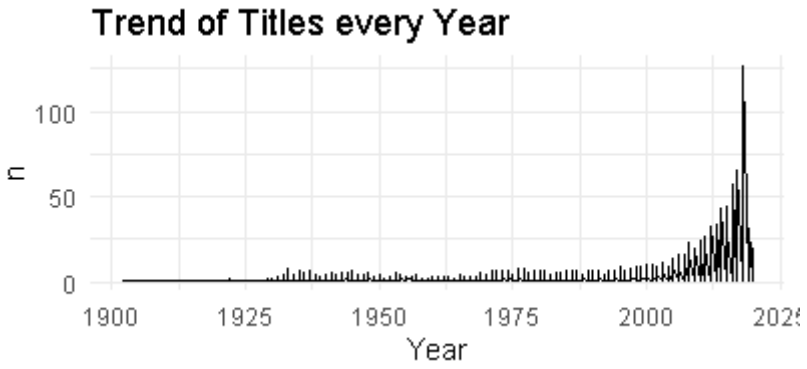


Fig 4.3 Trends of each year

- **IMDB Trends On Netflix**

```
dp_country %>%
 filter(Netflix==1) %>%
group_by(IMDb,Year) %>%
 count() %>%
 ggplot(aes(x=IMDb,y=Year))+geom_point() +
 labs(title = 'IMDB Trend of every Year on Netflix') +theme_minimal()
```

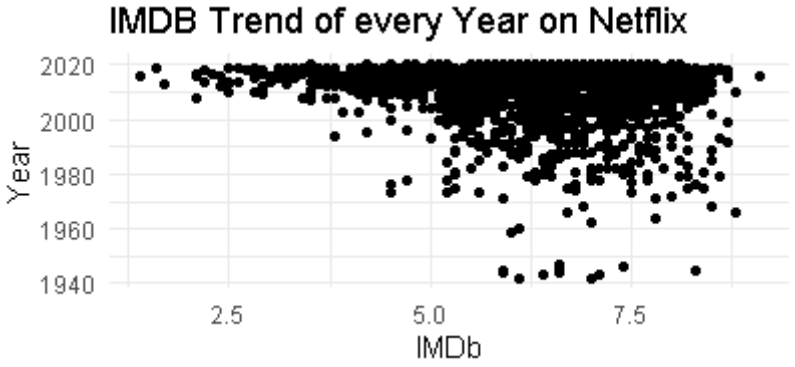


Fig 4.4 trends on Netflix

- **Number of titles based on themes / Genre of Titles**

```
dp1 %>% filter(Genres!='') %>%
  select(Genres) %>%
  mutate(Genres = str_split(Genres,',')) %>%
  unnest(Genres) %>%
  mutate(Genres= trimws(Genres, which = c("left")))%>%
  group_by(Genres) %>%
  count() %>%
  arrange(desc(n)) %>%
  head(30) %>%
  ggplot() + geom_col(aes(y = reorder(Genres,n), x = n)) +
  labs(title = 'Genres of Movies',
       x = 'Movie Titles',
;
       y = 'Genres') +
  theme_minimal()
```

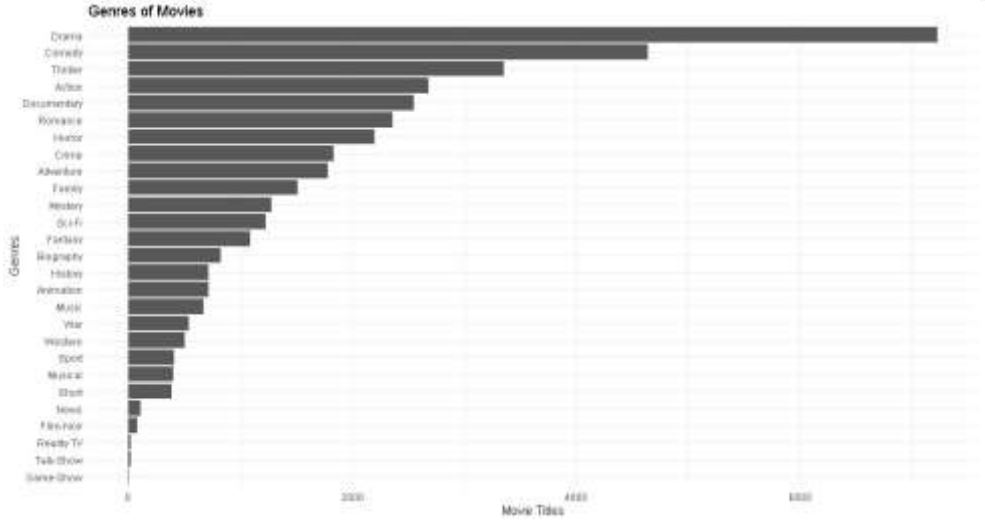


Fig 4.5 trends by geners

- **No of Titles By Directors**

```
dp1_dir <- dp1 %>% filter(Directors!='') %>%
  select(Directors) %>%
  mutate(Directors = str_split(Directors,',')) %>%
  unnest(Directors) %>%
  mutate(Directors= trimws(Directors, which = c("left")))%>%
  group_by(Directors) %>%
  count() %>%
  arrange(desc(n)) %>%
  head(30) %>%
  ggplot() + geom_col(aes(y = reorder(Directors,n), x = n)) +
  labs(title = 'No of Movies',
       x = 'Movie Titles',
       y = 'Directors') +
  theme_minimal()
dp1_dir
```

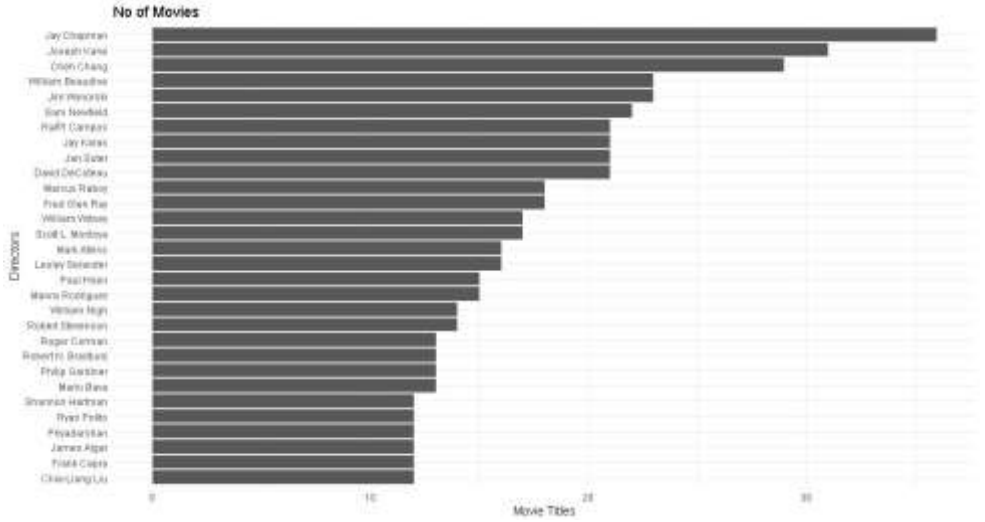


Fig 4.6 No of movies

- **Age Distribution Of Movies**

```
dp1 %>% filter(Age !='') %>% count(Age, sort = T) %>%
  mutate(prop = paste0(round(n / sum(n) * 100, 0), "%")) %>%
  ggplot(aes(x = "", y = prop, fill = Age)) +
  geom_bar(
    stat = "identity",
    width = 1,
    color = "steelblue",
    size = 1
  )+
  coord_polar("y", start = 0) +
  geom_text(
    aes(y = prop, label = prop),
    position = position_stack(vjust = 0.5),
    size = 6,
    col = "white",
    fontface = "bold"
  ) +
  scale_fill_manual (values = c('#e41a1c', '#377eb8','#6ffc76','#a765e0','#f547b2','#22d487')) +
  theme_void() +
  labs(
    title = "Age Distribution Of Movies Across All OTT Platforms",
     subtitle = "Pie Plot, Age Distribution across Movies",
    fill = ""
  )
```

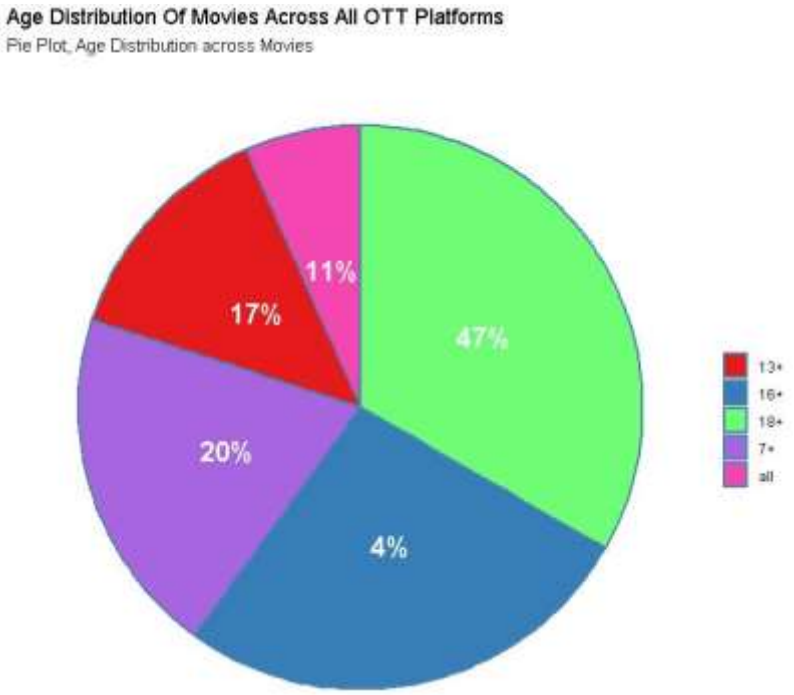
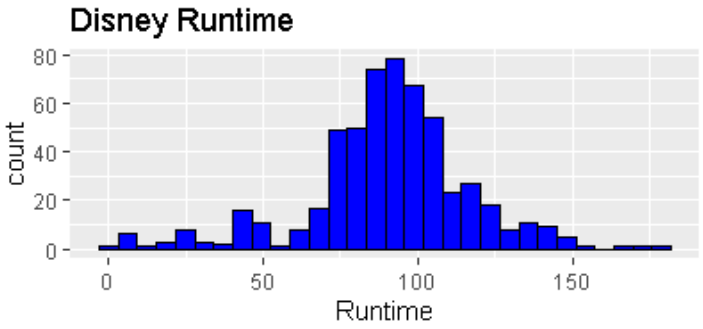
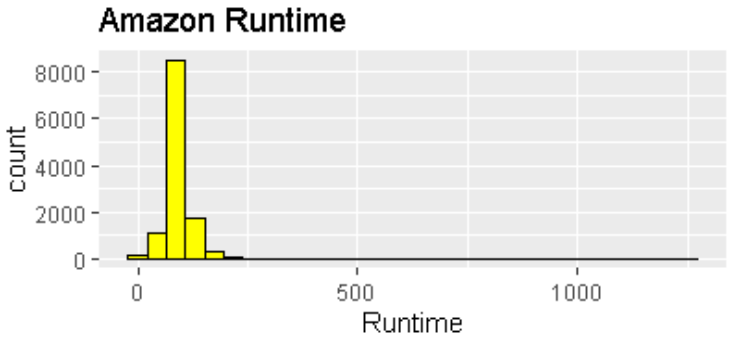


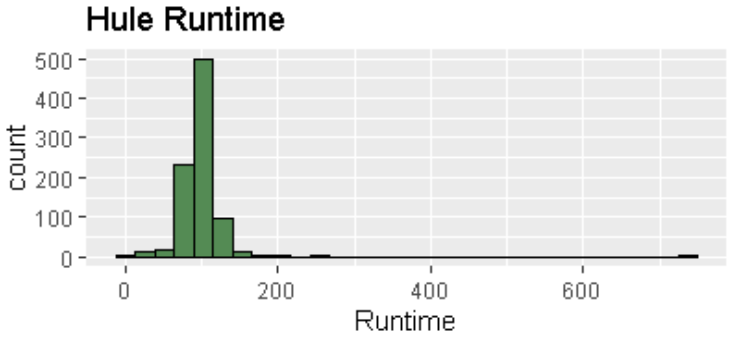
Fig 4.7 Age Distribution

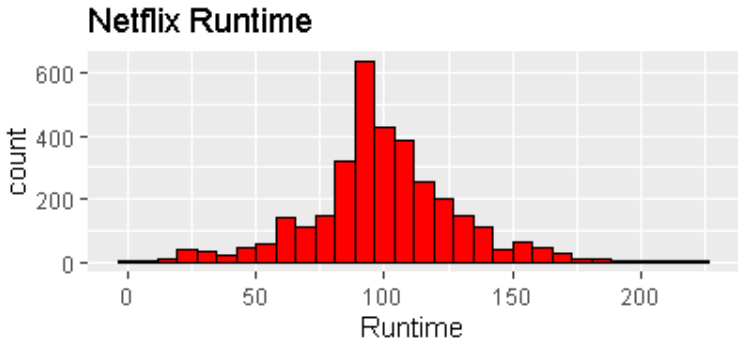
- **Runtime Analysis of platforms:**

```
ggplot(data = Netflix
,aes(x=Runtime))+geom_histogram(fill='red',col='black')+ggtitle("Netflix Runtime")
ggplot(data = Hulu
,aes(x=Runtime))+geom_histogram(fill='palegreen4',col='black')+ggtitle("Hule Runtime")
ggplot(data = Amazon
,aes(x=Runtime))+geom_histogram(fill='yellow',col='black')+ggtitle("Amazon Runtime")
ggplot(data = Disney
,aes(x=Runtime))+geom_histogram(fill='Blue',col='black')+ggtitle("Disney Runtime")
```









4.4 The Training and Testing Phase:

4.4.1 Data preprocessing before traning

```
dp3_ml = dp1 %>%  select(Runtime ,Genres , Year,IMDb,Age) %>%
  mutate(Age_i = parse_number(Age))%>%  drop_na()

dp3_ml %>% filter(is.na(Runtime)==T & is.na(Age_i)== T & Age_i!= 'all' &
is.na(Year)==T & is.na(IMDb)==T) %>% select(Runtime ,Genres , Year,IMDb,Age_i )

dp113 = complete.cases(dp3_ml)
dp11_13 = dp11[dp113, ]
dp2_ml = dp3_ml[!duplicated(dp3_ml),]
```

4.4.2 Splitting data for traning and testing training and testing set

```
set.seed(88)
split <- sample.split(dp2_ml$IMDb, SplitRatio = 0.70)

#get training and test data
train <- subset(dp2_ml, split == TRUE)
test <- subset(dp2_ml, split == FALSE)

train = train %>%
  mutate(Genres = strsplit(as.character(Genres), ",")) %>%
  unnest(Genres) %>%
  mutate(Genres = trimws(Genres, which = c("left"))) %>%
  mutate(Genres = tolower(Genres))

test = test %>%
  mutate(Genres = strsplit(as.character(Genres), ",")) %>%
  unnest(Genres) %>%
  mutate(Genres = trimws(Genres, which = c("left"))) %>%
  mutate(Genres = tolower(Genres))

#dummify data
dmy <- dummyVars("~ .", data= train)
train <- data.frame(predict(dmy, newdata = train))

dmy <- dummyVars("~ .", data= test)
test <- data.frame(predict(dmy, newdata = test))
```

4.4.3 Multiple linear regression

```
model <- lm(IMDb ~., data=train)
predicted_value <- predict(model, newdata = test)
multi_linear = as.data.frame(cbind(Actual = test$IMDb, Predicted = predicted_value))
error =  (multi_linear$Actual - multi_linear$Predicted)
multi_linear = as.data.frame(cbind(multi_linear,error))
rmse = sqrt(mean((error)^2))
head(multi_linear)
```

print(rmse)

```
> multi_linear = as.data.frame(cbind(Actual = test$IMDb, Predicted = predicted_value))
> error =  (multi_linear$Actual - multi_linear$Predicted)
> multi_linear = as.data.frame(cbind(multi_linear,error))
> rmse = sqrt(mean((error)^2))
> head(multi_linear)
  Actual Predicted    error
1    8.7  5.883517 2.816483
2    8.7  5.559584 3.140416
3    8.3  6.201172 2.098828
4    8.3  6.660270 1.639730
5    8.3  6.691015 1.608985
6    8.4  7.238405 1.161595
> print(rmse)
[1] 1.203138
```

4.4.4 Implementing Random Forest

#Random Forest
model <- randomForest(IMDb ~., data=train)
predicted_value <- predict(model, newdata = test)
random_forest = as.data.frame(cbind(Actual = test$IMDb , Predicted predicted_value))
error = (random_forest$Actual - random_forest$Predicted)
random_forest = as.data.frame(cbind(random_forest,error))
rmse = sqrt(mean((error)^2))
head(random_forest)
print(rmse)

```
> model <- randomForest(IMDb ~., data=train)
> predicted_value <- predict(model, newdata = test)
> random_forest = as.data.frame(cbind(Actual = test$IMDb , Predicted = predicted_value))
> error =  (random_forest$Actual - random_forest$Predicted)
> random_forest = as.data.frame(cbind(random_forest,error))
> rmse = sqrt(mean((error)^2))
> head(random_forest)
  Actual Predicted    error
1    8.7  6.329825 2.370175
2    8.7  6.302840 2.397160
3    8.3  6.267569 2.032431
4    8.3  6.491673 1.808327
5    8.3  6.281408 2.018592
6    8.4  6.681656 1.718344
> print(rmse)
[1] 1.186786
```

4.4.5 **Confusion Matrix:**

 # For Multiple Linear Regression:

table(test$IMDb,predicted_value > 5.0)

```
> table(test$IMDb,predicted_value > 5.0)

      FALSE TRUE
  1.7     1    2
  1.8     4    1
  2       4    9
  2.1     4   10
  2.2     2    6
  2.3     4    7
  2.4    12   12
  2.5     7   11
  2.6     4    9
  2.7     6   18
  2.8    13   23
  2.9     8   22
  3       6   21
  3.1     6   24
  3.2     9   18
  3.3    13   26
  3.4    12   33
  3.5    13   40
  3.6     9   35
  3.7    13   23
  3.8     8   48
  3.9     8   36
  4      17   36
  4.1    18   58
  4.2    15   51
  4.3     7   50
  4.4     9   56
  4.5    18   77
  4.6     7   78
  4.7    10   90
  4.8    12   95
  4.9    11   55
  5      26  103
  5.1    15  108
```

 # For Random Forest Algorithm:

table(test$IMDb,predicted_value > 6.0)

```
> table(test$IMDb,predicted_value > 6.0)

      FALSE TRUE
  1.7     3    0
  1.8     5    0
  2      13    0
  2.1    11    3
  2.2     7    1
  2.3     9    2
  2.4    19    5
  2.5    17    1
  2.6    13    0
  2.7    22    2
  2.8    32    4
  2.9    27    3
  3      27    0
  3.1    29    1
  3.2    23    4
  3.3    37    2
  3.4    43    2
  3.5    52    1
  3.6    43    1
  3.7    34    2
  3.8    49    7
  3.9    39    5
  4      48    5
  4.1    72    4
  4.2    52   14
  4.3    48    9
  4.4    44   21
  4.5    80   15
  4.6    74   11
  4.7    83   17
  4.8    88   19
  4.9    52   14
```

Chapter 5

Comparative study

Chapter 5: Comparative Study

Comparative Study helps to analyze the best the optimal solution for a problem and hence, this chapter enlightens the comparative analysis of Multiple linear regression Algorithm and Random Forest Algorithm Processing to highlight the optimal algorithm.

5.1 Comparison of Two Algorithms

The comparison of both algorithm through results can give us a better understanding about their accuracy and usefulness within the project. The Fig 5.1 which is for Multiple Linear regression gives us the actual value along with predicted and error value. The root mean square error is 1.203138 here. In Fig 5.2 which is for Random Forest it gives us the actual value along with predicted and error value. The root mean square error is 1.186786 here. Hence from here from the root mean square error we can judge that the Random forest algorithm is much more accurate and efficient as compared to the Multiple linear regression.

```
> multi_linear = as.data.frame(cbind(Actual = test$IMDb, Predicted = predicted_value))
> error =  (multi_linear$Actual - multi_linear$Predicted)
> multi_linear = as.data.frame(cbind(multi_linear,error))
> rmse = sqrt(mean((error)^2))
> head(multi_linear)
  Actual Predicted    error
1    8.7  5.883517 2.816483
2    8.7  5.559584 3.140416
3    8.3  6.201172 2.098828
4    8.3  6.660270 1.639730
5    8.3  6.691015 1.608985
6    8.4  7.238405 1.161595
> print(rmse)
[1] 1.203138
```

Fig 5.1 Multiple Linear Regression

```
> model <- randomForest(IMDb ~., data=train)
> predicted_value <- predict(model, newdata = test)
> random_forest = as.data.frame(cbind(Actual = test$IMDb , Predicted = predicted_value))
> error =  (random_forest$Actual - random_forest$Predicted)
> random_forest = as.data.frame(cbind(random_forest,error))
> rmse = sqrt(mean((error)^2))
> head(random_forest)
  Actual Predicted    error
1    8.7  6.329825 2.370175
2    8.7  6.302840 2.397160
3    8.3  6.267569 2.032431
4    8.3  6.491673 1.808327
5    8.3  6.281408 2.018592
6    8.4  6.681656 1.718344
> print(rmse)
[1] 1.186786
```

Fig 5.2 Random Forest Algorithm

Chapter 6

Conclusion

Chapter 6: Conclusion

This chapter aggregates conclusion and future scope. The conclusion emphasis on the collective summary of the project and lays the idea about how the project is planned and what is comprises of. The future scope highlights the scope of development in the proposed project with time.

6.1 Conclusion

Herein, I after cleaning and removing duplicate values from data analyzed the data with respect to its IMDb ratings which I predicted using many other independent parameters . The aim of this experiment was movies on OTT analysis wherein I predict the IMDb ratings based on various other parameters using Multiple Linear Regression Algorithm and Random Forest Algorithm. For the data we used kaggle which provided us with the data the data was around 16,744 which was later processed and manipulated to get meaning full insights .From results It is clear that the accuracy of Random Forest Algorithm with root mean square error = 1.186786 is a bit higher as compared to Multiple Linear Regression Algorithm with root mean square error = 1.203138 but as the data will increase the accuracy of the algorithms will also change. Hence the addition of more data will help us in better prediction for actual values.

6.2 Future Scope

The Existing Database is of around 16k data which is not sufficient to get meaningful insights from the data that accurately . Also, this type of database is limited for processing of structured data and has a limitation when dealing with a large amount of data. So, the use of Big Data technologies like Hadoop can be used to achieve better results. We could further improve our analysis by using various different algorithms and see which gives us better accuracy.

References

[1] Yasen, M., & Tedmori, S. (2019). *Movies Reviews Sentiment Analysis and Classification. 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT).* doi:10.1109/jeeit.2019.8717422.

[2] Kayri, M., Kayri, I., & Gencoglu, M. T. (2017). *The performance comparison of Multiple Linear Regression, Random Forest and Artificial Neural Network by using photovoltaic and atmospheric data. 2017 14th International Conference on Engineering of Modern Electric Systems (EMES).* doi:10.1109/emes.2017.7980368