BI FINAL REPORT

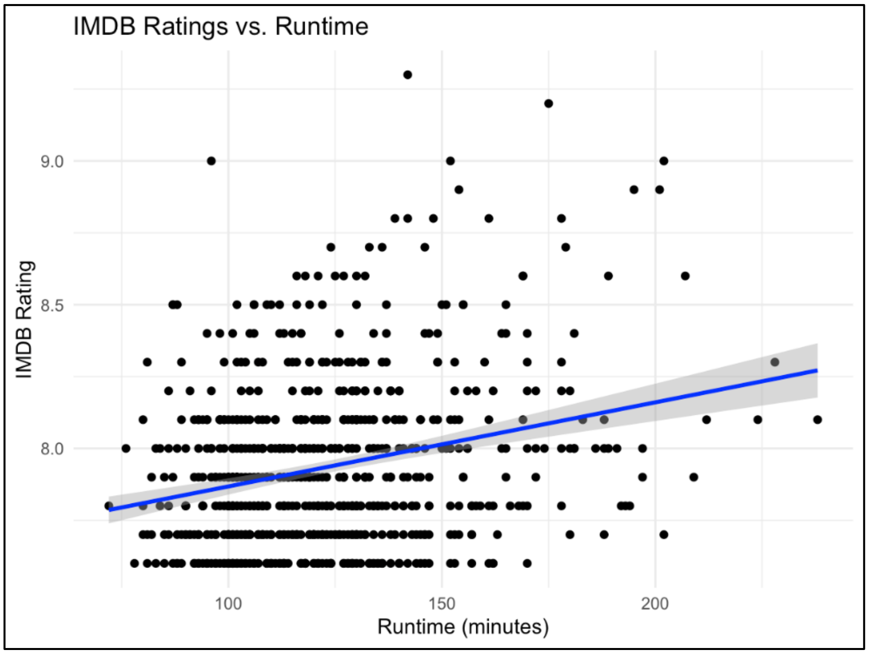
1. Research Question:

Exploring the IMDB ratings of movies and TV Shows based on various variables.

2. Codes

a. For IMDB ratings vs Runtime

```
# Load necessary libraries
library(ggplot2)
library(dplyr)
library(tidyr)
library(caret)
library(MASS)

# Convert runtime to numeric, assuming 'min' values are NA or errors for now
d <- d%>%
mutate(Runtime = gsub(" min", "", Runtime), Runtime = as.numeric(Runtime))

# Now plotting
p <- ggplot(d, aes(x=Runtime, y=IMDB_Rating)) + geom_point() +   geom_smooth(method="lm", se=TRUE,
color="blue") + labs(x="Runtime (minutes)", y="IMDB Rating", title="IMDB Ratings vs. Runtime") +
theme_minimal()

# Print the plot
print(p)
```

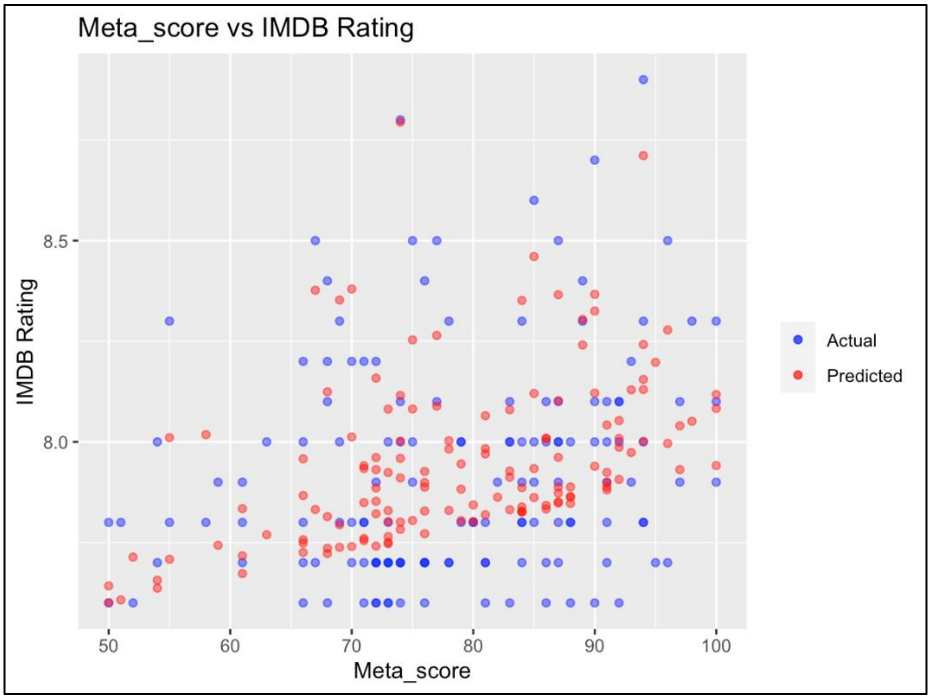


Interpretation: We noticed a positive correlation between runtime of movies (minutes) and the IMDb rating, suggesting that the movies with longer runtime generally receive higher ratings. Confidence interval indicates variability, not all movies follow this trend.

Movies with high IMDb ratings (>8.5) have diverse runtimes, which implies that there are multiple other factors other than length which impact the IMDB ratings. Results suggest viewer perception of quality in longer, comprehensive films, influenced by expectations for storytelling, character development, and production value. Genre differences in typical runtimes may contribute to higher ratings.

b. For linear regression:

```
# Load necessary libraries
library(ggplot2)
library(dplyr)
library(tidyr)
library(caret)
library(MASS)

# Load the dataset
data <- read.csv("cleaned_dataset_specific_columns.csv")

# Preparing the data
independent_vars <- data[,c("Meta_score", "No_of_Votes")]
dependent_var <- data$IMDB_Rating

# Splitting the data into training and testing sets
set.seed(42) # For reproducibility
trainIndex <- createDataPartition(dependent_var, p = .8,  list = FALSE, times = 1)
data_train <- data[trainIndex,]
data_test <- data[-trainIndex,]

# Fitting the linear regression model
model <- lm(IMDB_Rating ~ Meta_score + No_of_Votes, data=data_train)

# Predicting IMDB Ratings
predictions <- predict(model, data_test)

# Plotting

#  IMDB_Rating vs Meta_score
ggplot(data_test, aes(x=Meta_score, y=IMDB_Rating)) +
geom_point(aes(color="Actual"), alpha=0.5) +
geom_point(aes(x=Meta_score, y=predictions, color="Predicted"), alpha=0.5) +
labs(title="Meta_score vs IMDB Rating", x="Meta_score", y="IMDB Rating") +
scale_color_manual("", breaks = c("Actual", "Predicted"),values = c("blue", "red"))
```

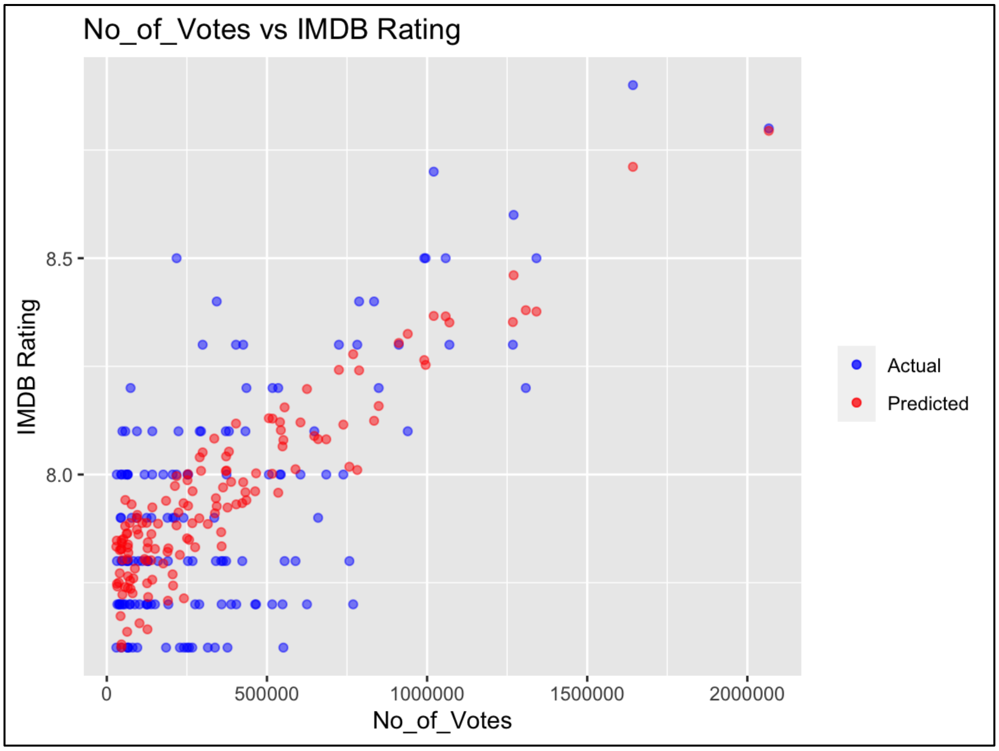


Interpretation: The scatter plot compares the IMDb ratings with the Meta scores. Blue points represent actual ratings, red points show predictions of the ratings. The RMSE value is 0.08017525, showing that the model is quite accurate.

Results indicate a quantifiable relationship between Meta score and IMDb ratings. Although the model is reasonably accurate, there are discrepancies between actual and predicted values, that suggest varied preferences amongst IMDb users and Meta score critics. Despite the low RMSE, differences in audience and critic perceptions contribute to prediction errors. Meta scores can generally predict IMDb ratings, but the inherent variability in preferences across platforms should be considered.

# IMDB_Rating vs No_of_Votes

```
ggplot(data_test, aes(x=No_of_Votes, y=IMDB_Rating)) +
geom_point(aes(color="Actual"), alpha=0.5) +geom_point(aes(x=No_of_Votes, y=predictions,
color="Predicted"), alpha=0.5) +labs(title="No_of_Votes vs IMDB Rating", x="No_of_Votes", y="IMDB
Rating") + scale_color_manual("",  breaks = c("Actual", "Predicted"), values = c("blue", "red"))
```



Interpretation: Scatter plot shows relationship between IMDb ratings vs. No. Of votes. Blue points depict actual ratings, red points show predictions. Low RMSE of 0.0926 depicts that the model is quite accurate. The connection between votes and ratings is good but not perfect and the discrepancies between the actual and predicted data tells us that there are other factors that contribute to deviations from predictions on the 1-10 IMDb scale.

# Display model summary to get coefficients

```
summary(model)

Call:
lm(formula = IMDB_Rating ~ Meta_score + No_of_Votes, data = data_train)
Residuals:
    Min      1Q   Median      3Q      Max
-0.56994 -0.15387 -0.02562  0.14825  0.76252
Coefficients:
              Estimate   Std. Error t value          Pr(>|t|)
(Intercept) 7.24322017909 0.05748416722 126.004 <0.0000000000000002 ***
Meta_score  0.00668392333 0.00073143899   9.138 <0.0000000000000002 ***
No_of_Votes 0.00000051113 0.00000002602  19.640 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.2196 on 571 degrees of freedom
Multiple R-squared:  0.4553,           Adjusted R-squared:  0.4534
F-statistic: 238.6 on 2 and 571 DF,  p-value: < 0.00000000000000022
```

Interpretation: The intercept value of IMDB is 7.24, and the p-value is significantly low, stating both these values are quite significant to predict the IMDB, however, there would be other factors as well, to accurately predict the ratings.

Information on dataset:

Link:
https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows

Context:

IMDB Dataset of top 1000 movies and tv shows.

Content

Data: -

- **Poster Link** - Link of the poster that imdb using
- **Series_Title** - Name of the movie
- **Released_Year** - Year at which that movie released
- **Certificate** - Certificate earned by that movie
- **Runtime** - Total runtime of the movie
- **Genre** - Genre of the movie
- **IMDB_Rating** - Rating of the movie at IMDB site
- **Overview** - mini story/ summary
- **Meta_score** - Score earned by the movie
- **Director** - Name of the Director
- **Star1, Star2, Star3, Star4** - Name of the Stars
- **No_of_votes** - Total number of votes
- **Gross** - Money earned by that movie

Preprocessing:

The main focus of this research was to study various variables affecting the IMDB ratings of movies and TV Shows. NA/Null values were present were negligible to the size of dataset. Assuming the missing values were MCAR type, we are dropping entries with missing values.