Sub : ML USING WATSON STUDIO.

PROJECT ;

STUDENT PERFORMANCE PREDICTION ;-

ABSTRACT:

This project predictive the student performance analysis: study hours and student scores through linear regression in R. Simulated data with 100 observations of study hours and corresponding scores is used. A scatter plot illustrates this relationship.IN R language programming we using packages like: tidyverse, caret, caTools The dataset is split into training (80%) and testing (20%) sets. A linear regression model is trained on the training set and evaluated using the test set. The Mean Absolute Error (MAE) is calculated to assess prediction accuracy. The final model is saved for future use, demonstrating its potential in educational planning and performance forecasting.

INTRODUCTION:

This project aims to explore the number of hours students study and their resulting scores. By using linear regression in R programming language, we seek to predict student scores based on their study hours. The project involves data simulation, model training, and evaluation, culminating in a saved predictive model.

ALGORITHMS :

**1.Load Libraries and Install Packages:**

* The code installs and loads the tidyverse package for data manipulation and visualization and the caTools package for data splitting.

install.packages("tidyverse")

install.packages("caret")

library(tidyverse)

library(caTools)

**2.Simulate Data:**

* The code simulates a dataset with 100 observations, containing Hours (hours studied) and Scores (scores obtained).

set.seed(123)

data <- data.frame(

Hours = runif(100, min = 1, max = 10),

Scores = runif(100, min = 50, max = 100)

)

**3.Scatter Plot:**

* The code creates a scatter plot of Hours vs Scores.

ggplot(data, aes(x = Hours, y = Scores)) +

geom\_point() +

labs(title = "Scatter plot of Hours Studied vs Scores", x = "Hours Studied", y = "Scores")

**4.Split Data:**

* The code splits the dataset into training (80%) and testing (20%) sets.

set.seed(123)

split <- sample.split(data$Scores, SplitRatio = 0.8)

train\_data <- subset(data, split == TRUE)

test\_data <- subset(data, split == FALSE)

**5.Fit Linear Regression Model:**

* The code fits a linear regression model to predict Scores based on Hours using the training data.

model <- lm(Scores ~ Hours, data = train\_data)

summary(model)

**6.Predict Scores:**

* The code makes predictions on the test set and combines actual and predicted scores.

predictions <- predict(model, test\_data)

results <- data.frame(

Actual = test\_data$Scores,

Predicted = predictions

)

**7.Plot Actual vs Predicted Scores:**

* The code creates a scatter plot of actual vs predicted scores.

ggplot(results, aes(x = Actual, y = Predicted)) +

geom\_point() +

geom\_abline(slope = 1, intercept = 0, color = "red") +

labs(title = "Actual vs Predicted Scores", x = "Actual Scores", y = "Predicted Scores")

**8.Calculate Mean Absolute Error (MAE):**

* The code calculates and prints the MAE.

mae <- mean(abs(results$Actual - results$Predicted))

cat("Mean Absolute Error:", mae, "\n")

**9.Save the Model:**

* The code saves the trained model to a file.

saveRDS(model, file = "student\_score\_model.rds")

SOURCE CODE :

install.packages("tidyverse")

install.packages("caret")

library(tidyverse)

library(caTools)

# Simulate some data

set.seed(123)

data <- data.frame(

Hours = runif(100, min = 1, max = 10),

Scores = runif(100, min = 50, max = 100)

)

# Scatter plot of the data

ggplot(data, aes(x = Hours, y = Scores)) +

geom\_point() +

labs(title = "Scatter plot of Hours Studied vs Scores", x = "Hours Studied", y = "Scores")

# Split the data into training and testing sets

set.seed(123)

split <- sample.split(data$Scores, SplitRatio = 0.8)

train\_data <- subset(data, split == TRUE)

test\_data <- subset(data, split == FALSE)

# Fit a linear regression model

model <- lm(Scores ~ Hours, data = train\_data)

summary(model)

# Predict scores on the test set

predictions <- predict(model, test\_data)

# Combine actual and predicted scores

results <- data.frame(

Actual = test\_data$Scores,

Predicted = predictions

)

# Plot actual vs predicted scores

ggplot(results, aes(x = Actual, y = Predicted)) +

geom\_point() +

geom\_abline(slope = 1, intercept = 0, color = "red") +

labs(title = "Actual vs Predicted Scores", x = "Actual Scores", y = "Predicted Scores")

# Calculate the Mean Absolute Error (MAE)

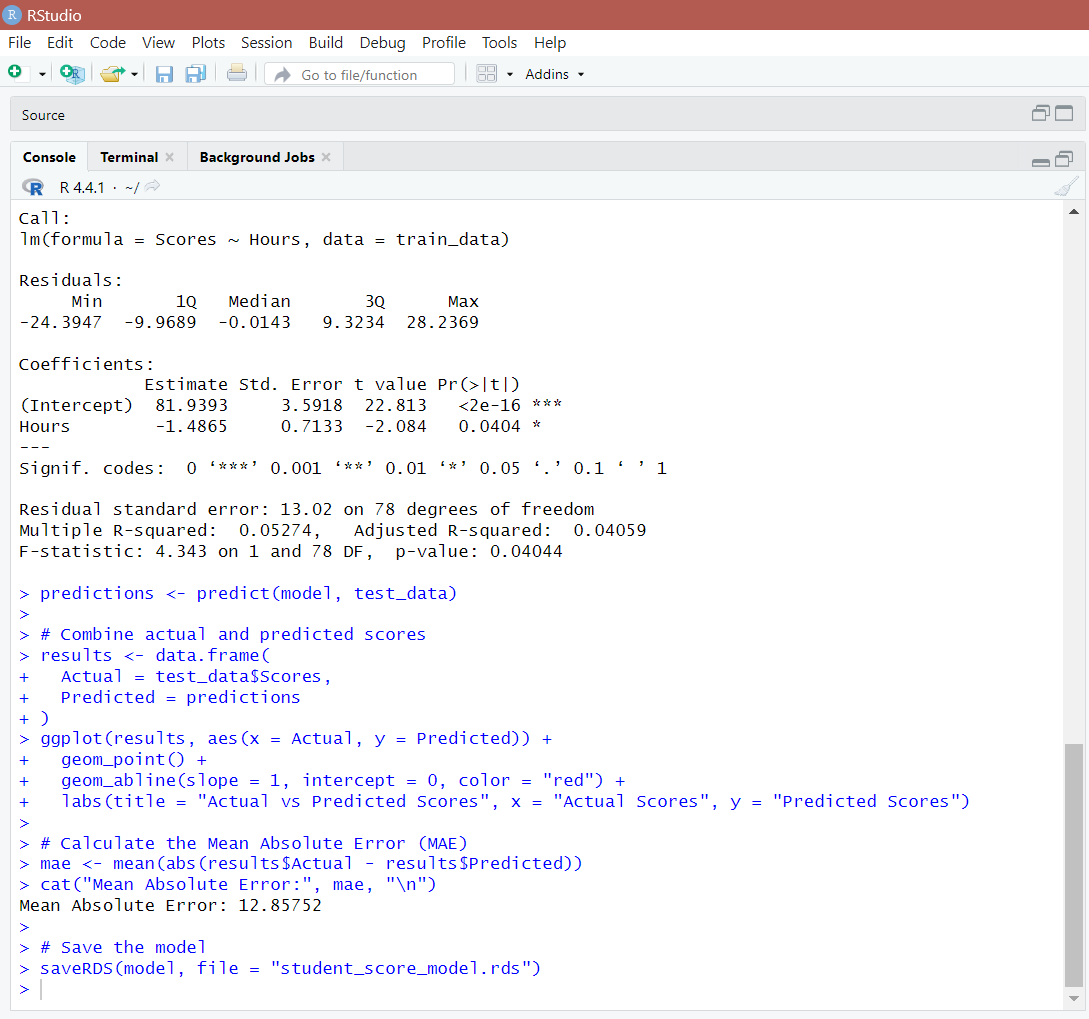
mae <- mean(abs(results$Actual - results$Predicted))

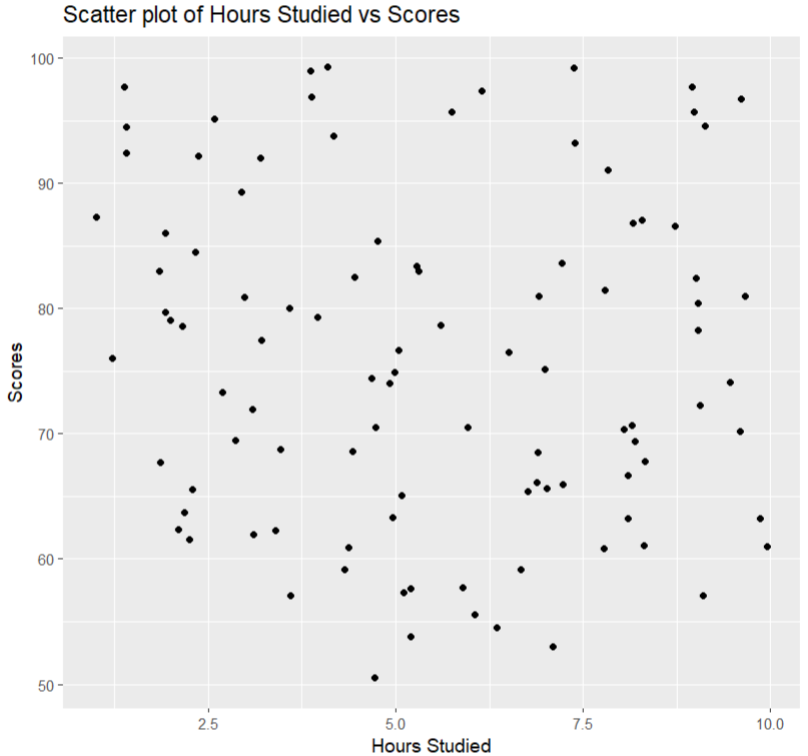
cat("Mean Absolute Error:", mae, "\n")

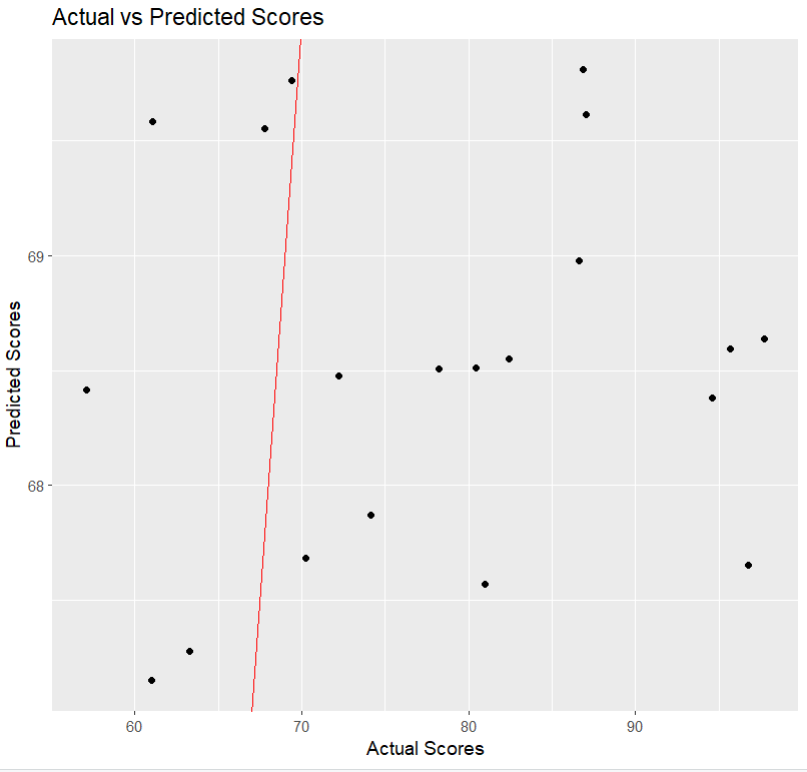
# Save the model

saveRDS(model, file = "student\_score\_model.rds")

OUTPUT :







Conclusion:

The project concludes with saving the trained model for future use. The results demonstrate the model's effectiveness in predicting student scores based on study hours, with potential applications in educational planning and student performance forecasting.

By developing this predictive model, we gain valuable insights into how study time impacts academic performance. Such a model can be a powerful tool for educators to identify students who might need additional support and for students to optimize their study habits. The simplicity and effectiveness of the linear regression model in this context demonstrate its potential application in real-world educational settings.