Predictive Analysis Report: Students Performance & Clustering

A Project Report

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

(Computer Science Engineering)

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

Prepared for

Continuous Assessment 3

Autumn 2024

SUBMITTED BY

Name of student: Yogesh

Registration Number: 12219441

Roll No.: 24

Section: K22ZM

Name of Supervisor: Anchal

Kaundal UID of Supervisor: 29612

Declaration:

I, Yogesh, student of *Program Name* under the CSE/IT discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Certificate

This is to certify that *Yogesh* bearing Registration No. ***12219441*** has completed the project titled *"**Predictive Analysis Report: Students Performance & Clustering**"* under my guidance and supervision. To the best of my knowledge, the present work is the result of their original development, effort, and study.

Acknowledgement:

I would like to express my sincere gratitude to my project guide, ***Anchal Kaundal*** , for their invaluable guidance, encouragement, and support throughout the course of this project. Special thanks to Lovely Professional University for providing the resources and environment to work on this project.

Table of Contents

Untitled11

November 17, 2024

1 Predictive Analysis of Student Performance

1.1 Overview

The goal of this project is to perform predictive analysis on the performance of students based on their test scores. We will use three machine learning models:

- **Linear Regression**: Predicts continuous values (Math score).
- **Support Vector Machine (SVM)**: Another regression technique for predicting continuous values.
- **Naive Bayes**: A classification model used to categorize students' performance based on their Math score into three categories: Low, Medium, and High.

The analysis will be conducted using a publicly available dataset of student performance in various subjects (Math, Reading, and Writing).

1.2 Dataset

The dataset used in this project is the `StudentsPerformance.csv` file, which contains the following features:

- **gender**: The gender of the student.
- **race**: The race/ethnicity group of the student.
- **parental_education**: The level of education of the student's parents.
- **lunch**: Whether the student receives standard or free/reduced lunch.
- **test_preparation**: Whether the student completed test preparation or not.
- **math_score**: The student's score in the Math test.
- **reading_score**: The student's score in the Reading test.
- **writing_score**: The student's score in the Writing test.

1.3 Data Preprocessing

The first step in any machine learning task is to preprocess the data. This includes:

- Renaming columns to make them more readable.
- Converting categorical variables (gender, race, parental education, lunch, test preparation) to factors.
- Checking for missing values to ensure the integrity of the data.

1.4 Code:

```
[2]: # Load the dataset
dataset <- read.csv("C:\\Users\\asus\\OneDrive\\Desktop\\K22ZM\\INT234 Machine␣
 ↪learning\\Project\\archive\\StudentsPerformance.csv")
# Preprocess the data by renaming columns
colnames(dataset) <- c("gender", "race", "parental_education",
                       "lunch", "test_preparation",
                       "math_score", "reading_score", "writing_score")

# Convert categorical variables to factors
dataset$gender <- as.factor(dataset$gender)
dataset$race <- as.factor(dataset$race)
dataset$parental_education <- as.factor(dataset$parental_education)
dataset$lunch <- as.factor(dataset$lunch)
dataset$test_preparation <- as.factor(dataset$test_preparation)
```

2 Exploratory Data Analysis (EDA)

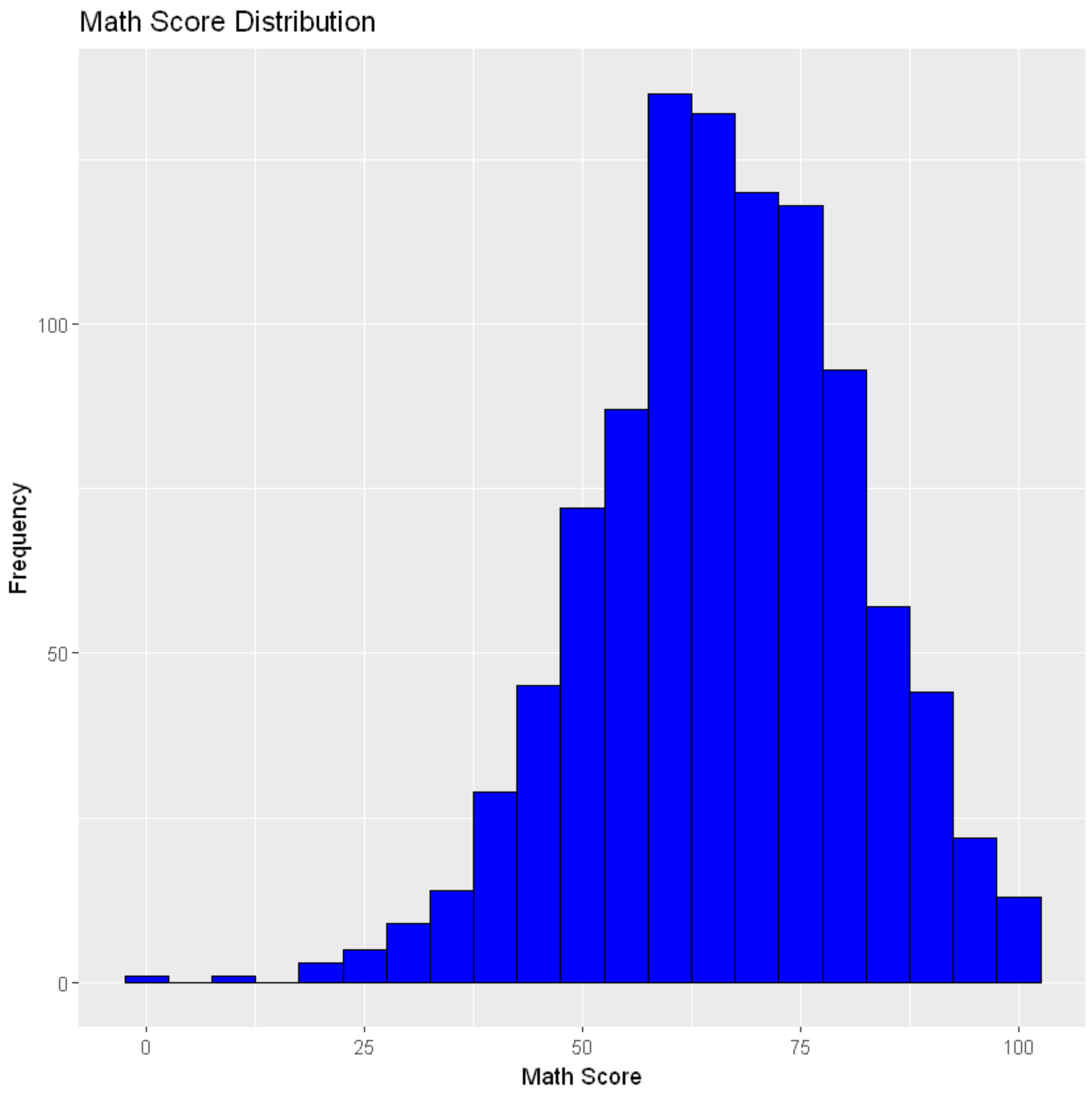
Before building the models, it is important to understand the dataset using Exploratory Data Analysis (EDA). This involves:

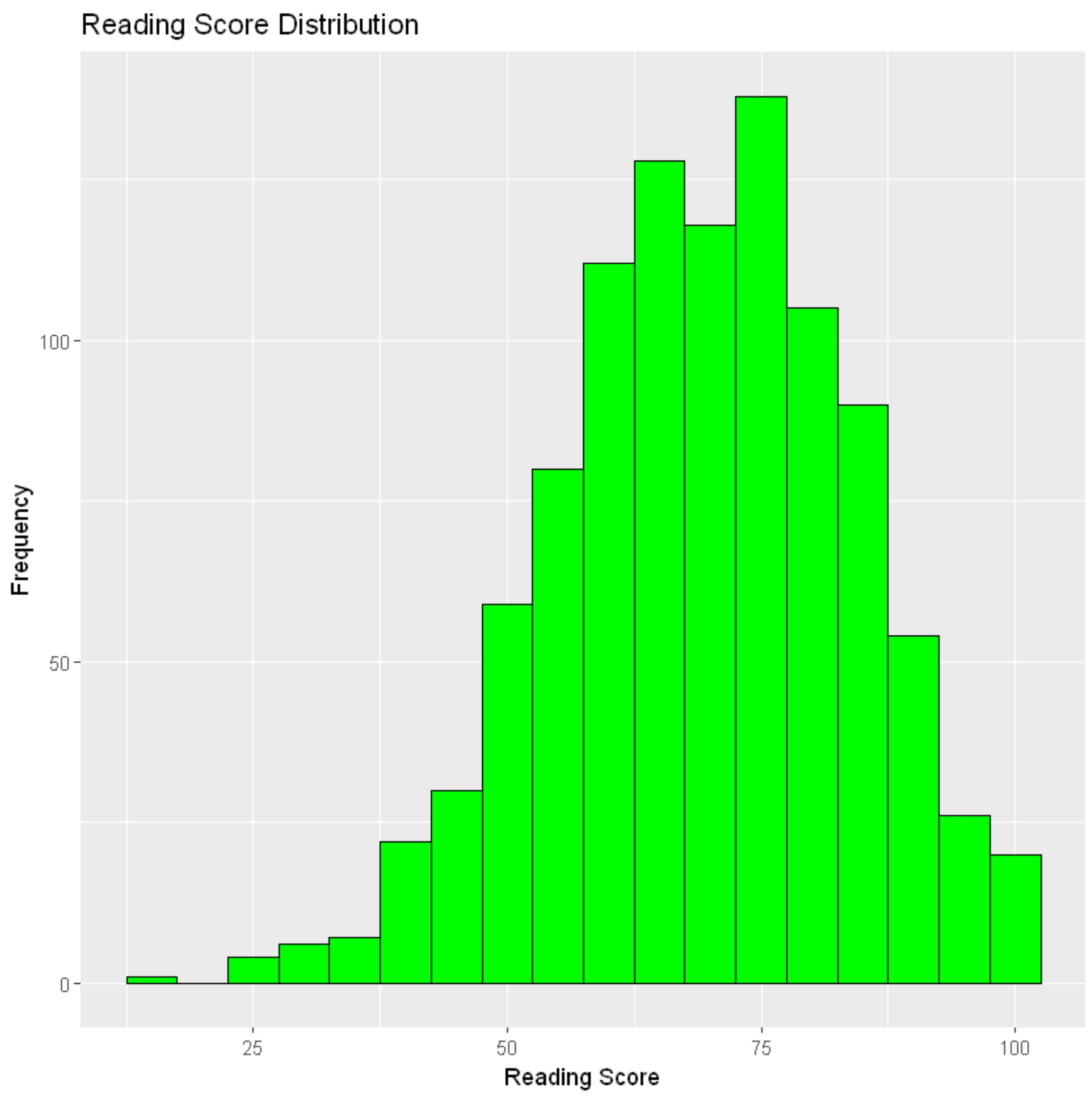
- **Summarizing the data**: Displaying the summary statistics and structure of the dataset.
- **Visualizing the distribution** of the numeric variables (Math, Reading, Writing scores).
- **Visualizing correlations** between scores to check for relationships between variables.
- **Examining categorical variables** (Gender, Race, Parental Education, etc.) and their distribution.

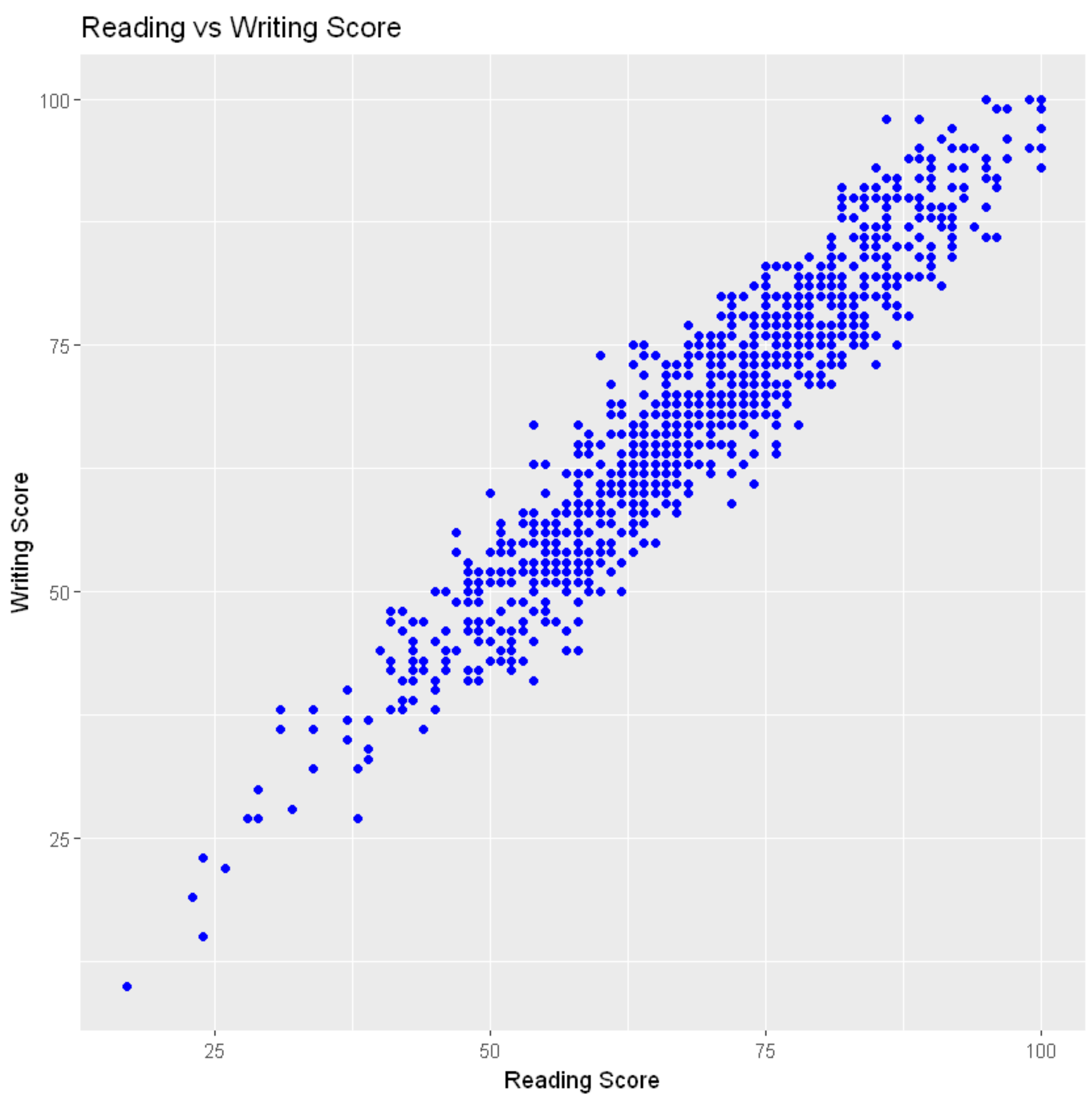
2.1 Code:

```
[3]: # Summarize the data
summary(dataset)

# Visualize the distribution of scores
ggplot(dataset, aes(x = math_score)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  labs(title = "Math Score Distribution", x = "Math Score", y = "Frequency")

ggplot(dataset, aes(x = reading_score)) +
  geom_histogram(binwidth = 5, fill = "green", color = "black") +
  labs(title = "Reading Score Distribution", x = "Reading Score", y =␣
 ↪"Frequency")

# Visualize correlation between scores
ggplot(dataset, aes(x = reading_score, y = writing_score)) +
  geom_point(color = "blue") +
  labs(title = "Reading vs Writing Score", x = "Reading Score", y = "Writing␣
 ↪Score")
```

```
   gender         race               parental_education          lunch
female:518   group A: 89   associate's degree:222     free/reduced:355
male  :482   group B:190   bachelor's degree :118     standard    :645
             group C:319   high school       :196
             group D:262   master's degree   : 59
             group E:140   some college      :226
                           some high school  :179
 test_preparation   math_score     reading_score    writing_score
completed:358     Min.   :  0.00   Min.   : 17.00   Min.   : 10.00
none     :642     1st Qu.: 57.00   1st Qu.: 59.00   1st Qu.: 57.75
                  Median : 66.00   Median : 70.00   Median : 69.00
                  Mean   : 66.09   Mean   : 69.17   Mean   : 68.05
                  3rd Qu.: 77.00   3rd Qu.: 79.00   3rd Qu.: 79.00
                  Max.   :100.00   Max.   :100.00   Max.   :100.00
```




Reading Score Distribution
Frequency
Reading Score
0
50
100
25
50
75
100



3 Model Building

Three different machine learning models were implemented to predict student performance:

3.1 1. Linear Regression

Linear regression is used to predict the continuous Math score based on the other two scores: Reading and Writing.

3.2 2. Support Vector Machine (SVM)

SVM is used as a regression technique (with a linear kernel) to predict the Math score.

3.3 3. Naive Bayes Classification

In the Naive Bayes model, the Math score is categorized into three performance levels: Low, Medium, and High. This is a classification task based on features such as Gender, Reading Score, and Writing Score.

3.4 Model Training and Prediction:

[4]:
```
# Train Linear Regression Model
lm_model <- lm(math_score ~ reading_score + writing_score, data = train_data)

# Train Support Vector Machine (SVM) Model
svm_model <- svm(math_score ~ reading_score + writing_score, data = train_data,␣
  ↪kernel = "linear")

# Train Naive Bayes Model
train_data$performance <- cut(train_data$math_score, breaks = c(-Inf, 60, 80,␣
  ↪Inf), labels = c("Low", "Medium", "High"))
nb_model <- naiveBayes(performance ~ gender + reading_score + writing_score,␣
  ↪data = train_data)
```

4 Model Evaluation

4.1 1. Linear Regression Evaluation

Linear regression is evaluated using **Mean Squared Error (MSE)**.

[5]:
```
lm_predictions <- predict(lm_model, test_data)
lm_mse <- mean((lm_predictions - test_data$math_score)^2)  # Mean Squared Error
```

4.2 2. SVM Evaluation

SVM performance is also evaluated using **Mean Squared Error (MSE)**.

[6]:
```
svm_predictions <- predict(svm_model, test_data)
svm_mse <- mean((svm_predictions - test_data$math_score)^2)  # Mean Squared␣
  ↪Error
```

4.3 3. Naive Bayes Evaluation

Naive Bayes is evaluated using **Accuracy**.

[7]:
```
nb_predictions <- predict(nb_model, test_data)
nb_cm <- confusionMatrix(nb_predictions, test_data$performance)
```

5 Results

The results from the models are displayed below:

- **Linear Regression**: The Mean Squared Error (MSE) of the linear regression model is computed.
- **SVM**: The MSE of the Support Vector Machine (SVM) model is computed.
- **Naive Bayes**: The accuracy of the Naive Bayes classification model is computed.

5.1 Code for Displaying Results:

[8]:
```
cat("Linear Regression MSE:", lm_mse, "\n")
cat("SVM MSE:", svm_mse, "\n")
cat("Naive Bayes Accuracy:", nb_cm$overall['Accuracy'] * 100, "%\n")
```

```
Linear Regression MSE: 75.44235
SVM MSE: 74.53416
Naive Bayes Accuracy: 69.69697 %
```

6 Shiny App

The Shiny app was created to provide an interactive dashboard for users to compare the three models visually. It allows users to select the model of interest and view the corresponding model performance metrics and visualizations.

[ ]:
```
# Shiny UI
library(shiny)
ui <- fluidPage(
  titlePanel("Predictive Analysis Dashboard: Students Performance"),
  sidebarLayout(
    sidebarPanel(
      h4("Model Performance"),
      p("Compare the performance of three predictive models: Linear Regression,␣
 ↪SVM, and Naive Bayes."),
      selectInput("model_type", "Choose a Model:",
                  choices = c("Linear Regression", "SVM", "Naive Bayes"),
                  selected = "Linear Regression")
    ),
    mainPanel(
      h3("Model Metrics"),
      textOutput("model_metrics"),
      plotOutput("true_vs_predicted"),
      plotOutput("performance_distribution")
    )
  )
)

# Shiny Server
server <- function(input, output) {

  # Display Model Metrics
  output$model_metrics <- renderText({
    if (input$model_type == "Linear Regression") {
      paste("Linear Regression MSE:", round(lm_mse, 2))
    } else if (input$model_type == "SVM") {
      paste("SVM MSE:", round(svm_mse, 2))
    } else {paste("Naive Bayes Accuracy:", round(nb_cm$overall['Accuracy'] *␣
↪100, 2), "%")
    }
  })

  # Plot True vs Predicted for Regression Models
  output$true_vs_predicted <- renderPlot({
    if (input$model_type == "Linear Regression") {
      ggplot() +
        geom_point(aes(x = test_data$math_score, y = lm_predictions), color =␣
↪"blue") +
        labs(title = "Linear Regression: True vs Predicted",
             x = "True Math Score", y = "Predicted Math Score") +
        geom_abline(color = "red", linetype = "dashed")
    } else if (input$model_type == "SVM") {
      ggplot() +
        geom_point(aes(x = test_data$math_score, y = svm_predictions), color =␣
↪"green") +
        labs(title = "SVM: True vs Predicted",
             x = "True Math Score", y = "Predicted Math Score") +
        geom_abline(color = "red", linetype = "dashed")
    }
  }) # Plot Performance Distribution for Naive Bayes
  output$performance_distribution <- renderPlot({
    if (input$model_type == "Naive Bayes") {
      ggplot(data.frame(Actual = test_data$performance, Predicted =␣
↪nb_predictions),
             aes(x = Actual, fill = Predicted)) +
        geom_bar(position = "dodge") +
        labs(title = "Naive Bayes: Performance Distribution",
             x = "Actual Performance", y = "Count", fill = "Predicted") +
        scale_fill_manual(values = c("Low" = "red", "Medium" = "orange", "High"␣
↪= "green"))
    }
  })
}

# Run Shiny App
shinyApp(ui = ui, server = server)
```

```
Listening on http://127.0.0.1:5146
```

7 Step-by-Step Guide: Predictive Analysis Dashboard

This guide provides a step-by-step walkthrough of the **Predictive Analysis Dashboard** built with Shiny. The app allows users to compare predictive models and visualize K-Means clustering results using the **Mall Customer Segmentation** dataset.

7.1 Step 1: Data Loading and Preprocessing

7.1.1 1.1 Load Libraries

The following libraries are required for the app:

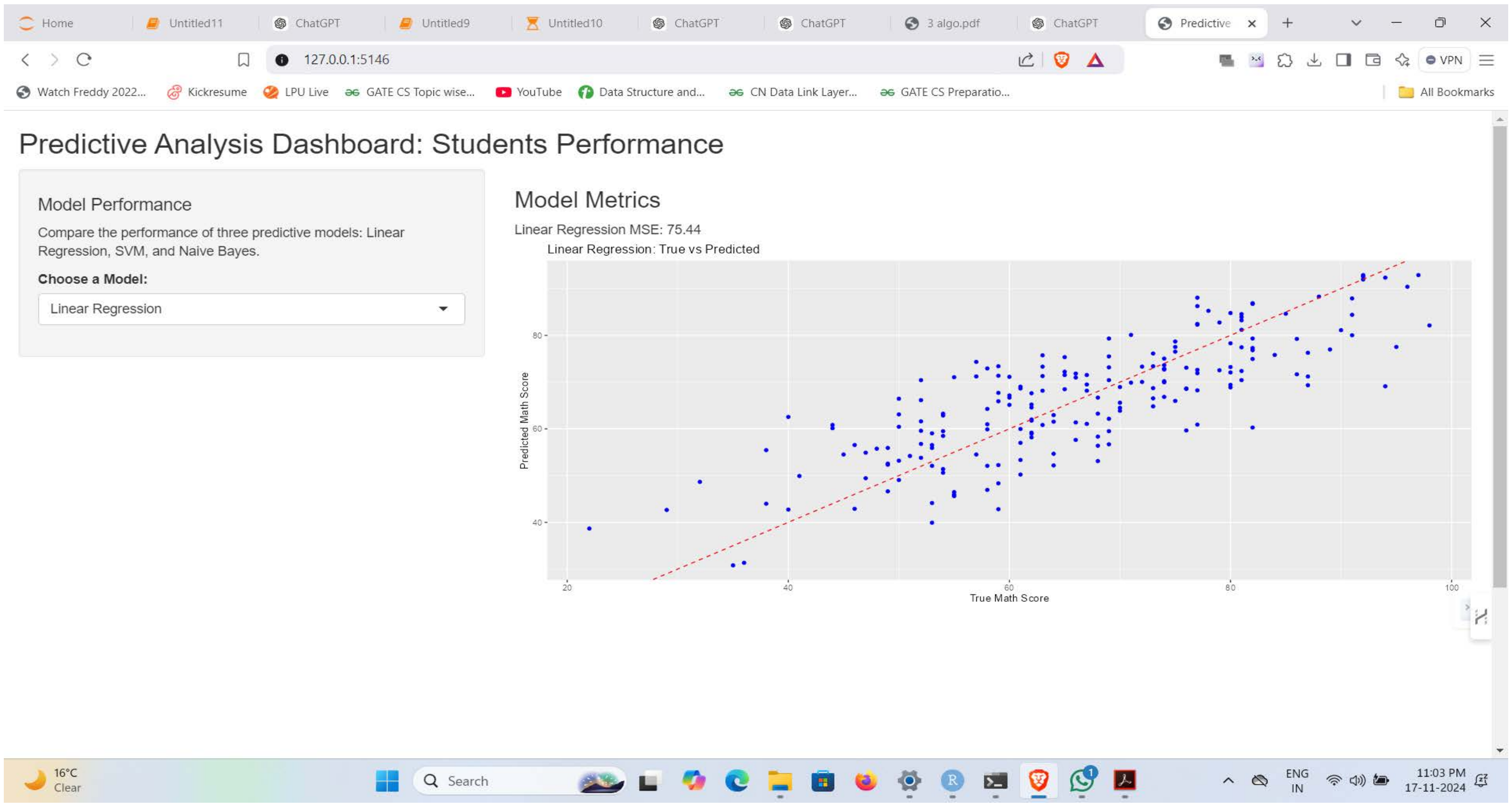
"'r library(shiny) library(tidyverse) # For data manipulation and visualization library(cluster) # For KMeans library(factoextra) # For visualizing clusters library(scales) # For scaling data

[ ]:
```
# Load necessary libraries
library(tidyverse)
library(cluster)
library(factoextra)

# Step 1: Load the Dataset
# Ensure the dataset is saved in your working directory
mall_data <- read.csv("C:\\Users\\asus\\OneDrive\\Desktop\\K22ZM\\INT234␣
 ↪Machine learning\\Project\\archive (2)ty\\Mall_Customers.csv")  # Replace␣
 ↪with the correct file path

head(mall_data)  # View the first few rows of the dataset
str(mall_data)   # Check the structure

# Step 2: Data Cleaning and Preprocessing
# Remove non-numeric columns like CustomerID and Gender
numeric_data <- mall_data %>% select(-CustomerID, -Gender)

# Check for missing values
sum(is.na(numeric_data))

# Standardize the data
scaled_data <- scale(numeric_data)

# Step 3: Determine Optimal Number of Clusters
# Use the Elbow Method to determine the ideal number of clusters
fviz_nbclust(scaled_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal Clusters")

# Step 4: Apply K-Means Clustering
# Set the number of clusters (choose based on the Elbow plot)
```

```
k <- 3  # Replace with the desired number of clusters
set.seed(123)  # For reproducibility
kmeans_result <- kmeans(scaled_data, centers = k, nstart = 25)

# Add cluster labels to the original dataset
mall_data$Cluster <- as.factor(kmeans_result$cluster)

# Step 5: Visualize Clusters
# PCA for dimensionality reduction and 2D visualization
pca <- prcomp(scaled_data)
pca_data <- as.data.frame(pca$x[, 1:2])  # Use the first two principal␣
↪components
pca_data$Cluster <- mall_data$Cluster

# Scatter plot of the clusters
ggplot(pca_data, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(size = 3, alpha = 0.8) +
  theme_minimal() +
  labs(title = "K-Means Clustering Results (PCA)", x = "Principal Component 1",␣
↪y = "Principal Component 2")

# Visualize cluster centers
fviz_cluster(kmeans_result, data = scaled_data) +
  labs(title = "Cluster Visualization")

# Step 6: Analyze Clusters
# Calculate the average values for each cluster
cluster_summary <- mall_data %>%
  group_by(Cluster) %>%
  summarise(across(where(is.numeric), mean))
print(cluster_summary)

# Step 7: Save the Results
# Save the dataset with cluster assignments
write.csv(mall_data, "Mall_Customers_with_Clusters.csv", row.names = FALSE)
```

[ ]:

DASHBOARD :



Predictive Analysis Dashboard: Students Performance

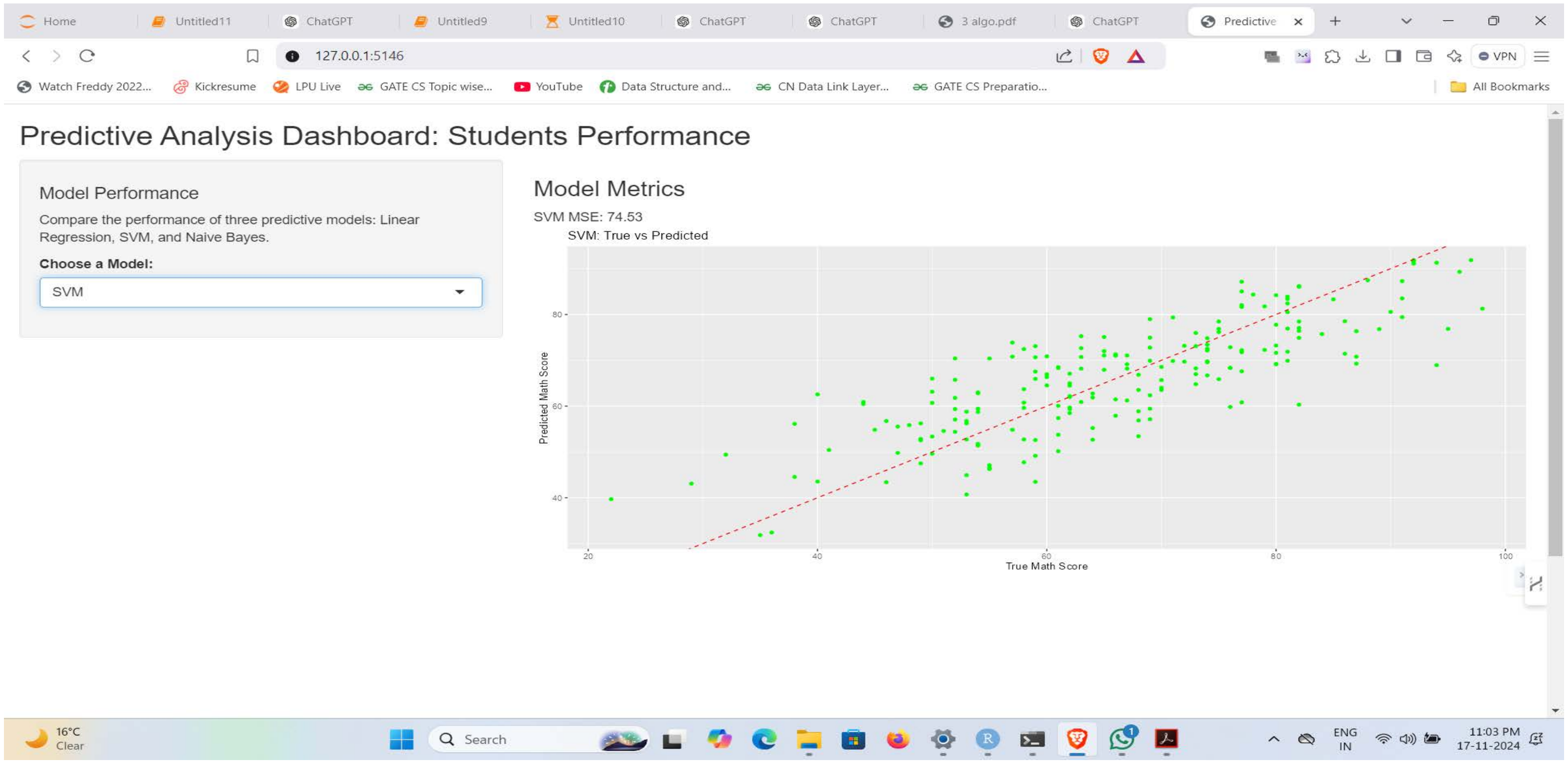
Model Performance

Compare the performance of three predictive models: Linear Regression, SVM, and Naive Bayes.

Choose a Model: Linear Regression

Model Metrics

Linear Regression MSE: 75.44



Predictive Analysis Dashboard: Students Performance

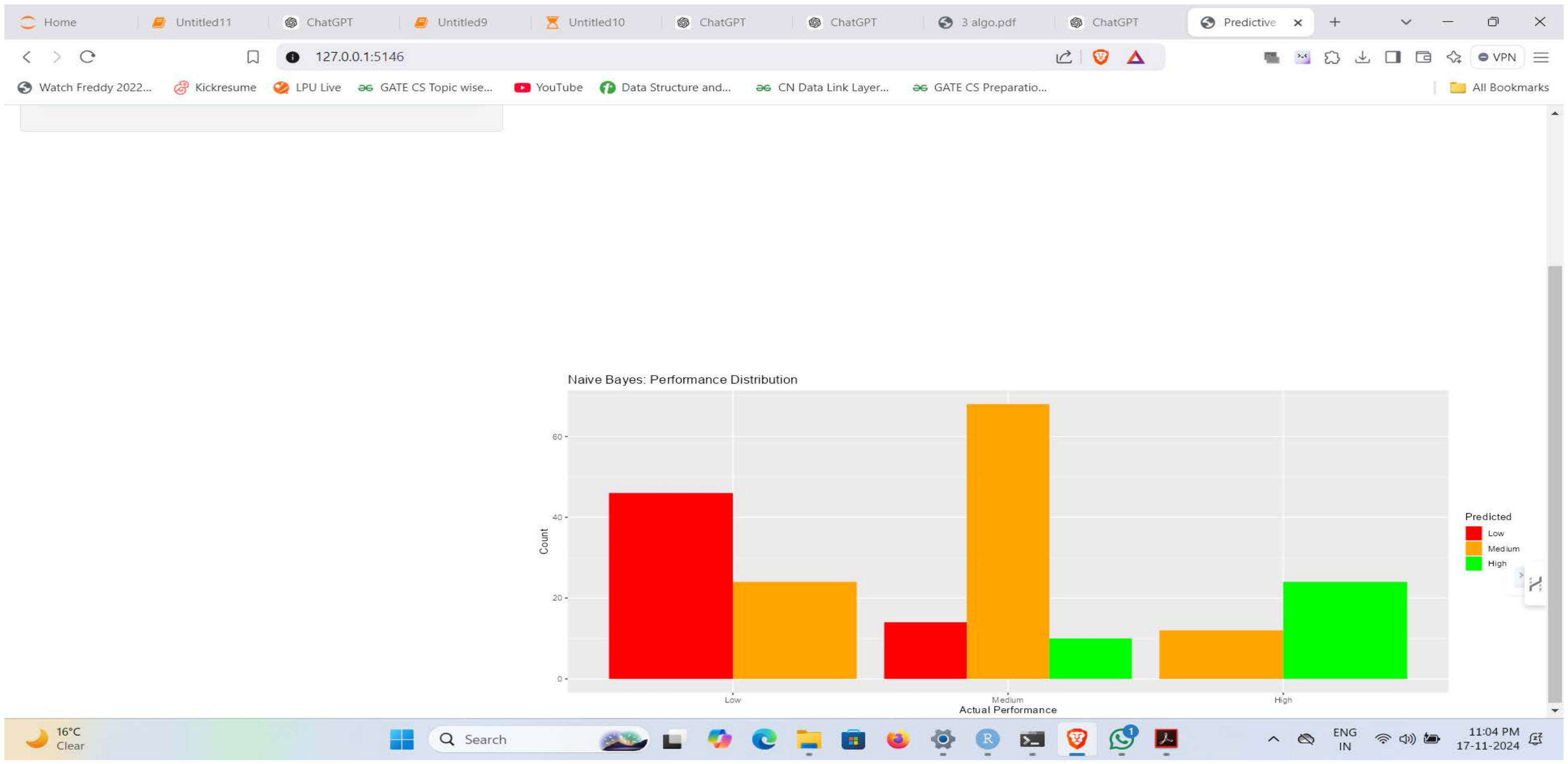
Model Performance

Compare the performance of three predictive models: Linear Regression, SVM, and Naive Bayes.

Choose a Model: SVM

Model Metrics

SVM MSE: 74.53


Naive Bayes: Performance Distribution
Count
0
20
40
60
Low
Medium
High
Actual Performance
Predicted
🟥 Low
🟧 Medium
🟩 High