**PREDICTIVE ANALYSIS PROJECT REPORT**

(Project Semester August-December 2024)

***Predictive Analysis on Student’s Performance Using Advanced Machine Learning Techniques***

Submitted by

Name: Chundru Rishith Sai Chowdary

Registration No: 12212330

Programme: B.Tech. (Computer Science and Engineering)

Course Code: INT234

Under the Guidance of

**Name: Madhu Bala**

**UID: 31770**

**Discipline of CSE/IT**

**Lovely School of Computer Science and Engineering**

**Lovely Professional University, Phagwara**

**DECLARATION**

I, **Chundru Rishith Sai Chowdary**, student of **B.Tech** under 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.

Date: 19-11-2024 Signature

Registration No. 12212330 Chundru Rishith Sai Chowdary

**ACKNOWLEDGMENT**

I would like to extend my gratitude to **Madhu Bala** Madam, for their guidance, support, and encouragement throughout this project. This report is the result of collaborative efforts and insights gained from my faculty member.

**TABLE OF CONTENTS**

1. **INTRODUCTION**
2. **SCOPE OF ANALYSIS**
3. **OBJECTIVES**
4. **METHODOLOGY**
5. **EXISTING SYSTEM**
6. **SOURCE OF DATASET**
7. **ETL PROCESS**
8. **ANALYSIS ON DATASET**
9. **LIST OF ANALYSIS WITH RESULTS**
10. **FUTURE SCOPE**
11. **CONCLUSION**
12. **REFERENCES**
13. **BIBLIOGRAPHY**

**INTRODUCTION**

This project analyses student performance data to predict participation in a test preparation course based on academic scores in math, reading, and writing. Using a variety of machine learning models, we aim to identify patterns in student achievements that correlate with course completion. This analysis includes both classification and clustering methods, which enable us to understand the underlying structure in the data and assess model effectiveness.

The models applied are:

1. **Naive Bayes Classification** – A probabilistic model that leverages conditional probabilities to make predictions, useful for understanding the likelihood of students completing the course based on their scores.
2. **Decision Tree Classification** – A tree-based model that splits data based on feature values, providing a visual and interpretable prediction path for student course completion.
3. **Support Vector Machine (SVM)** – A powerful classifier that maximizes the margin between data classes, allowing us to separate students based on performance features.
4. **Neural Network** – A predictive model with a hidden layer to capture complex patterns and interactions among features in the student performance data.
5. **K-Means Clustering** – An unsupervised learning method that groups students into clusters based on their scores, helping identify similar student profiles regardless of course completion.

Each model is evaluated on its prediction accuracy, and results are visualized in confusion matrices and heatmaps to illustrate the model's performance. Finally, a bar plot compares the accuracy of all models, providing insights into which methods best predict course participation based on student scores. This analysis can support educational strategies, guiding interventions to improve student performance and readiness.

**SCOPE OF ANALYSIS**

The scope of this analysis is to explore and evaluate various machine learning models to predict students’ participation in a test preparation course based on their academic performance in math, reading, and writing. Specifically, this analysis aims to:

1. **Identify Patterns in Academic Performance**: By analysing the correlation between scores and course participation, we aim to uncover insights into the characteristics of students who are likely to engage in preparation courses and those who may benefit from them.
2. **Compare Classification Model Performance**: Several supervised classification models (Naive Bayes, Decision Tree, SVM, and Neural Networks) are applied to assess their accuracy and suitability in predicting course participation. This comparison helps identify the most effective model for future use on similar datasets.
3. **Explore Student Grouping Using Clustering**: With K-Means clustering, the analysis seeks to find natural groupings within the data, allowing us to observe whether students with similar score patterns align with course participation or other characteristics.
4. **Evaluate and Visualize Model Accuracy**: The analysis includes visual tools such as heatmaps and bar plots to present model performance clearly. These visualizations help in understanding the effectiveness of each model and provide a means to communicate findings more intuitively.
5. **Support Educational Insights and Interventions**: The findings from this analysis can be applied to educational settings, helping educators and administrators identify which students may benefit from additional support or test preparation programs based on their performance patterns.

By encompassing both supervised and unsupervised learning techniques, this analysis not only aims to predict course participation but also to derive actionable insights that can support targeted educational interventions and improve student performance outcomes.

**OBJECTIVES**

The primary objectives of this analysis are:

1. **Predict Test Preparation Participation:** To build and evaluate machine learning models that accurately predict whether students completed a test preparation course based on their scores in math, reading, and writing.
2. **Compare Model Performance:** To assess and compare the accuracy and effectiveness of various machine learning models (Naive Bayes, Decision Tree, SVM, Neural Networks, and K-Means Clustering) in predicting course participation.
3. **Identify Key Performance Indicators:** To determine which academic factors (such as math, reading, or writing scores) most significantly influence students’ likelihood of participating in test preparation courses.
4. **Uncover Patterns in Student Performance:** To use clustering analysis to identify natural groupings within the data, which may reveal distinct student profiles based on score similarities and test preparation participation.
5. **Visualize Findings for Insights:** To create visualizations (e.g., heatmaps, confusion matrices, and bar plots) that effectively communicate model results, accuracy comparisons, and patterns in the data, providing insights that are easily interpretable for educators and stakeholders.
6. **Support Educational Decision-Making**: To provide data-driven insights that could aid educational institutions in designing or promoting test preparation programs and in identifying students who may benefit from additional support.

These objectives collectively aim to leverage machine learning to improve educational outcomes by better understanding student performance and preparation course participation patterns.

**METHODOLOGY**

1. **Naive Bayes Classification:**

* **Data Preparation**: Convert the test\_preparation\_course column to a factor to make it suitable for classification.
* **Data Splitting**: Split the dataset into 70% training data and 30% testing data to validate the model.
* **Model Training**: Use the naive Bayes function to create a Naive Bayes model that predicts test\_preparation\_course based on math\_score, reading\_score, and writing\_score.
* **Prediction**: Apply the trained model to the testing data.
* **Evaluation**: Create a confusion matrix to evaluate the model’s accuracy by comparing predicted vs. actual values. A heatmap visualizes the results.

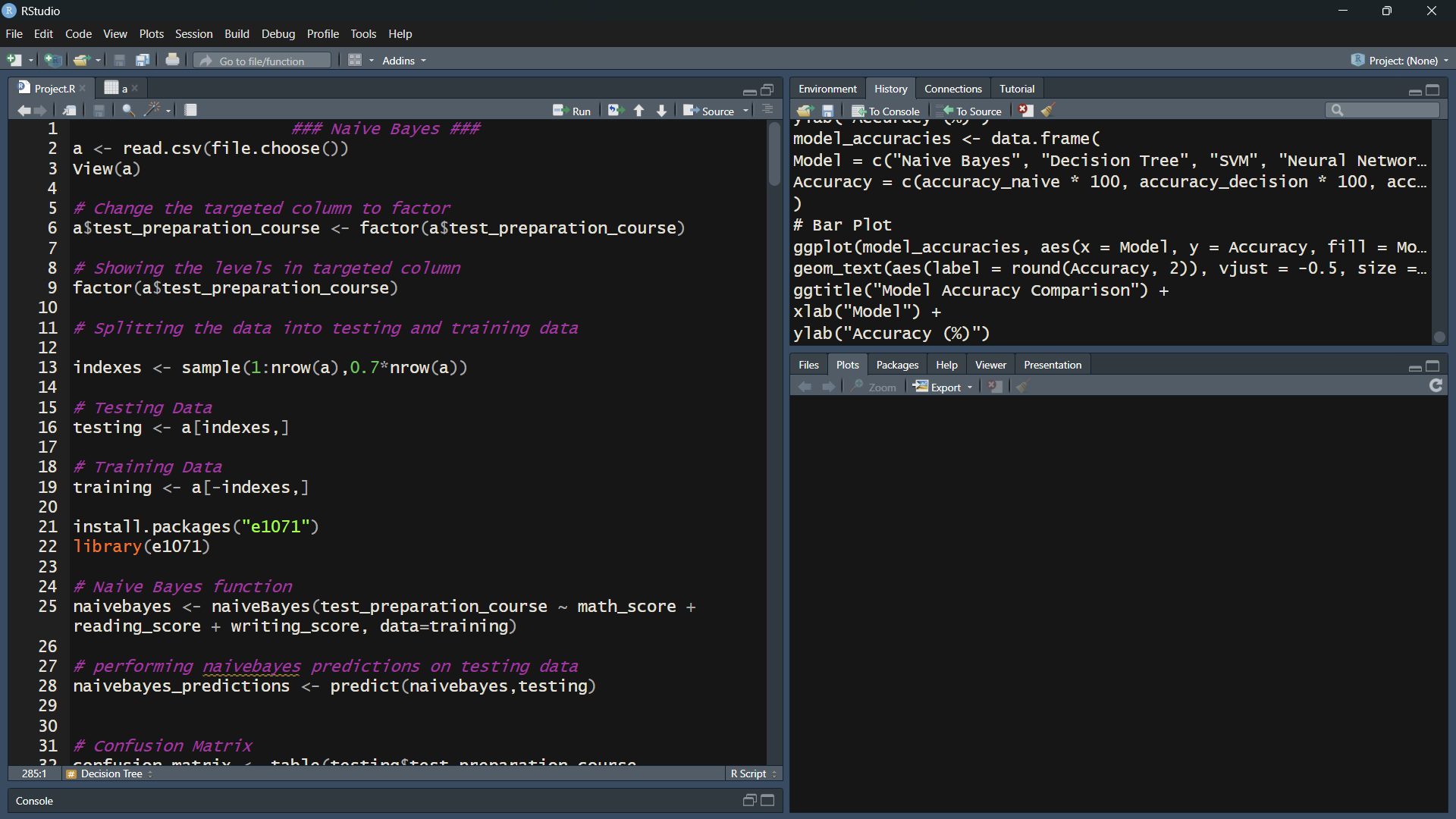


Fig 4.1: Naive Bayes Code(part-1)

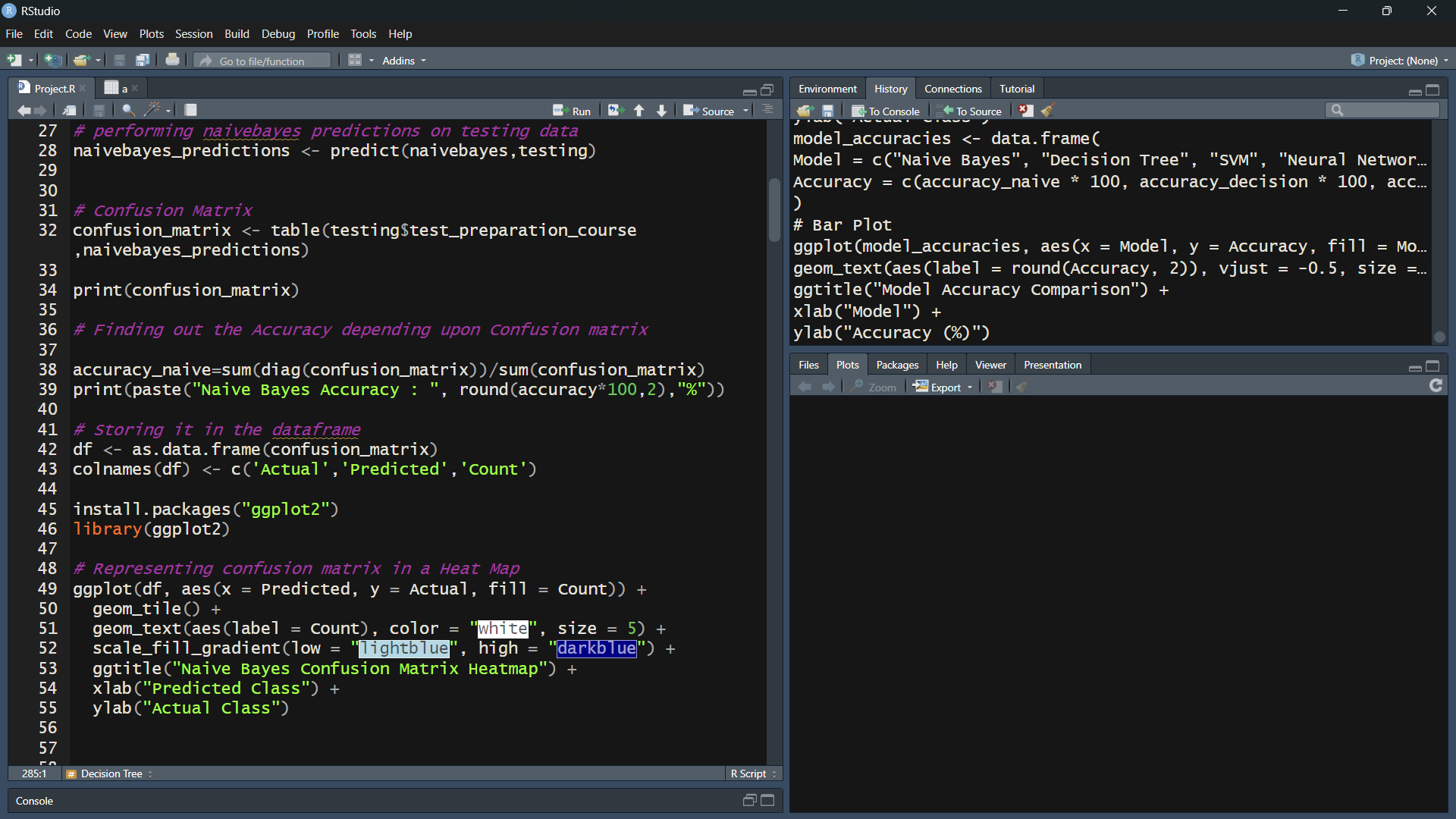


Fig 4.2: Naive Bayes Code(part-2)

1. **Decision Tree Classification:**

* **Data Splitting**: Use the same training and testing data as in Naive Bayes.
* **Model Training**: Build a decision tree classifier using the rpart function, which

identifies splits based on math, reading, and writing scores.

* **Prediction**: Predict test\_preparation\_course for the testing data.
* **Evaluation**: Calculate accuracy using a confusion matrix, and visualize it with a heatmap. The decision tree structure is visualized using the rpart.Plot library to understand the decision-making process.

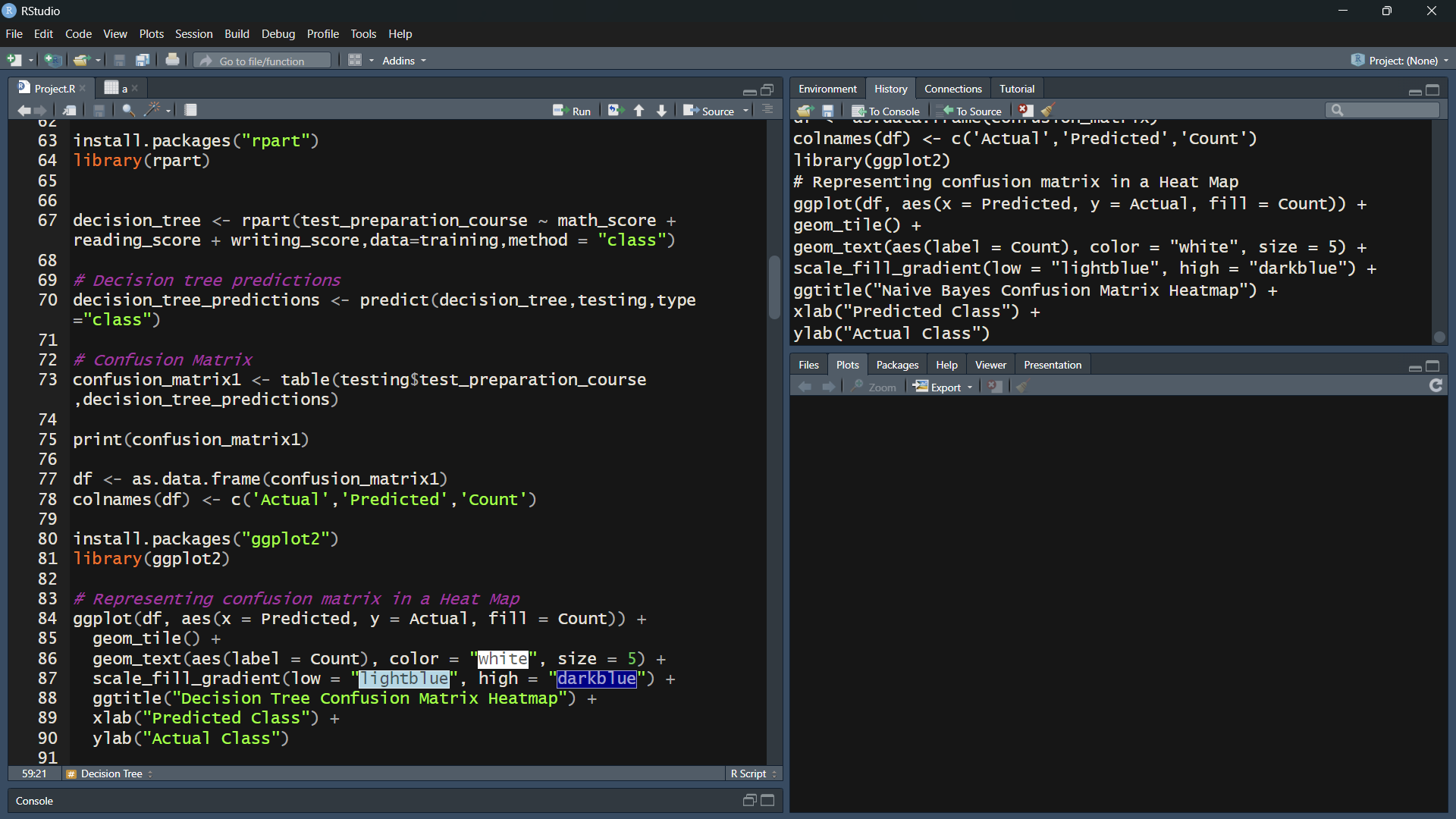


Fig 4.3: Decision Tree code(part-1)

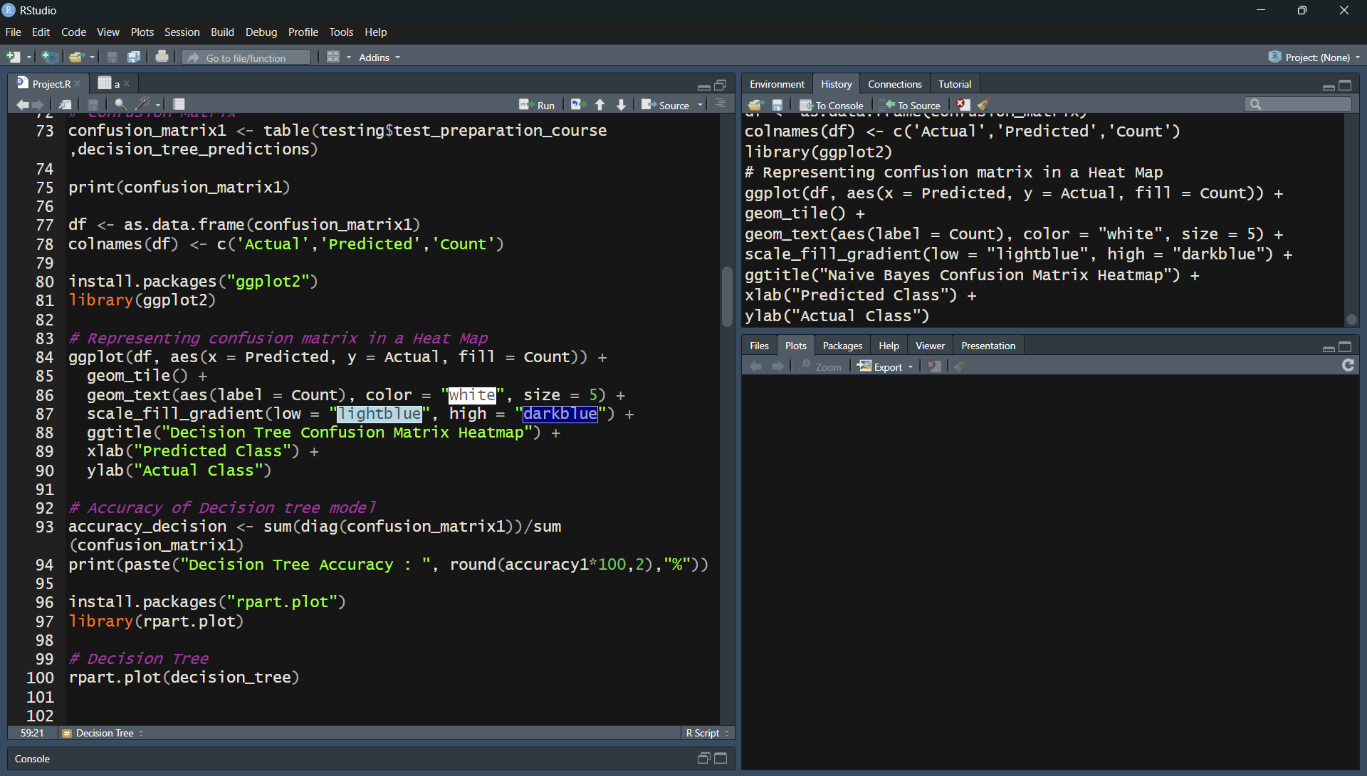


Fig 4.4: Decision Tree Code(part-2)

1. **Support Vector Machine (SVM):**

* **Data Preprocessing**: Apply feature scaling to numeric columns in both the training and testing sets to improve SVM performance.
* **Data Splitting**: Randomly split the data into training and testing sets, similar to previous models, with a 70/30 ratio.
* **Model Training**: Train an SVM classifier with a linear kernel using the svm function to separate students based on score features.
* **Prediction**: Use the trained SVM model to predict the test preparation status for the testing set.
* **Evaluation**: A confusion matrix and heatmap visualize the model’s performance. Calculate the overall accuracy.

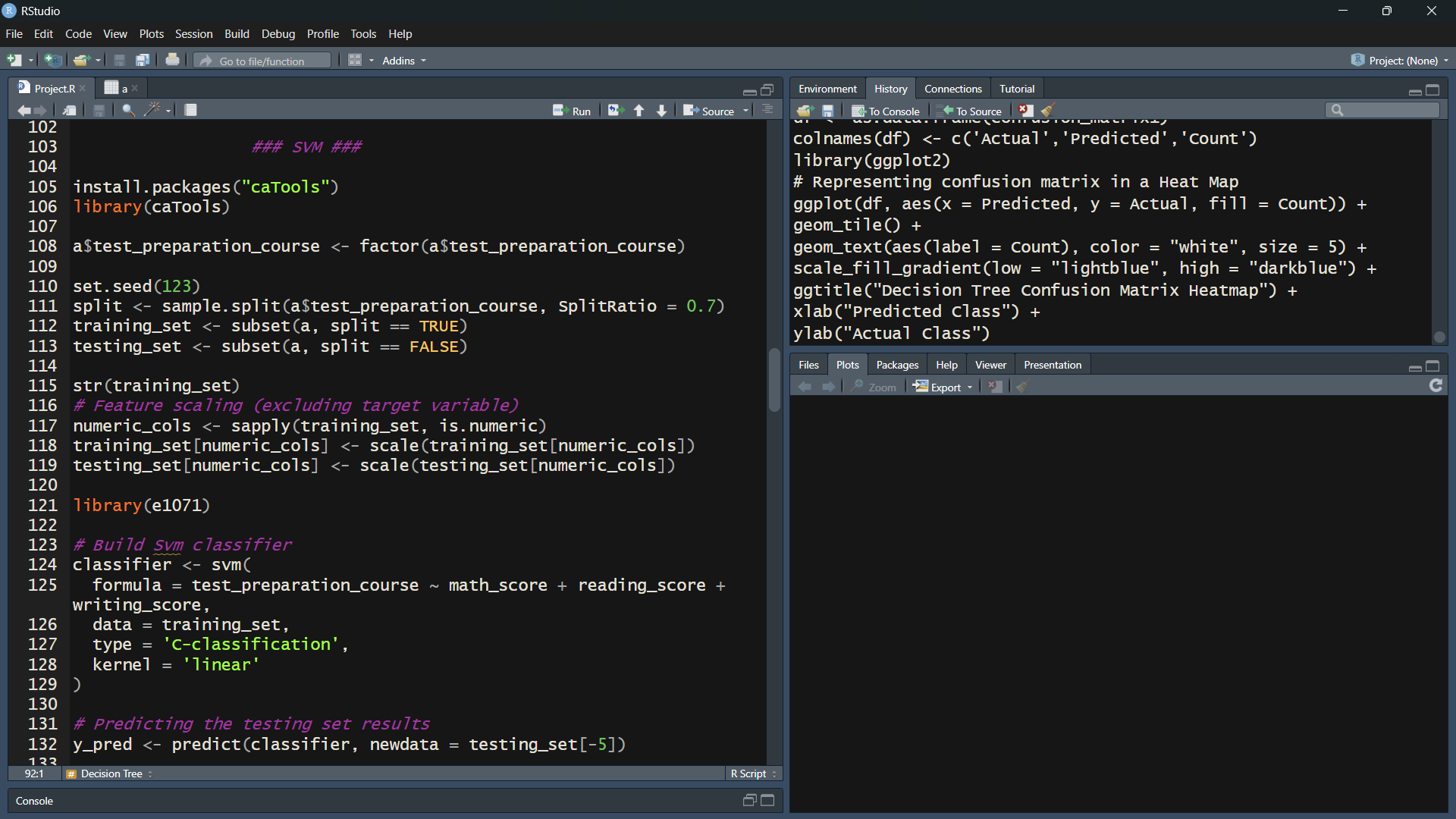


Fig 4.5: Support Vector Machine Code(part-1)

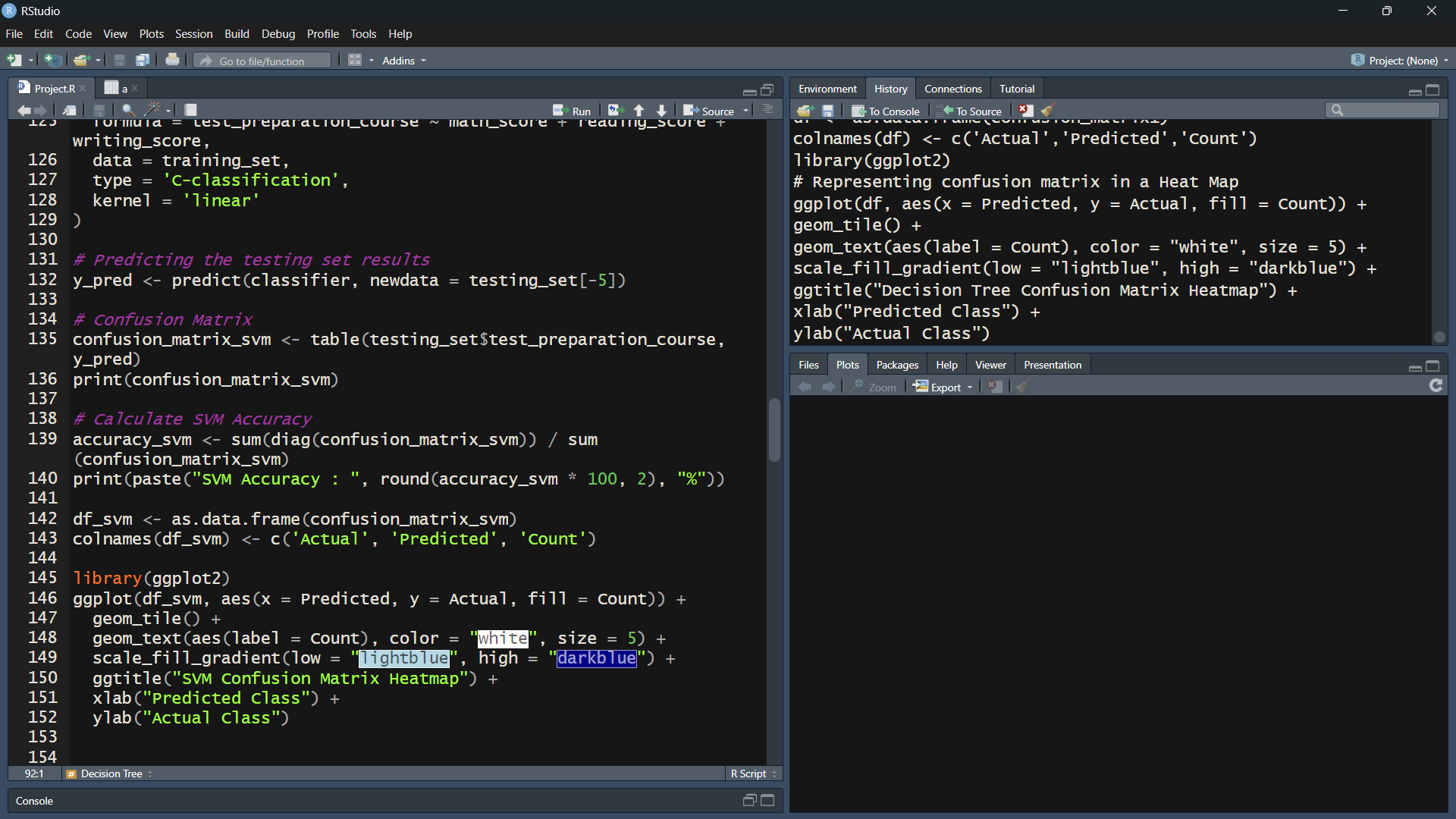


Fig 4.6: Support Vector Machine Code(part-2)

1. **Neural Network**

* **Data Normalization:** Normalize all numeric features to ensure that the neural network learns efficiently without being biased by scale differences.
* **Data Splitting:** Divide the normalized data into 70% training and 30% testing sets.
* Model Training: Build a neural network using the neuralnet function, specifying one hidden layer with 5 neurons.
* **Prediction:** Generate predictions on the test data.
* **Evaluation:** A confusion matrix is used to evaluate predictions, and accuracy is calculated. A heatmap visualizes the results, and the neural network structure is plotted.

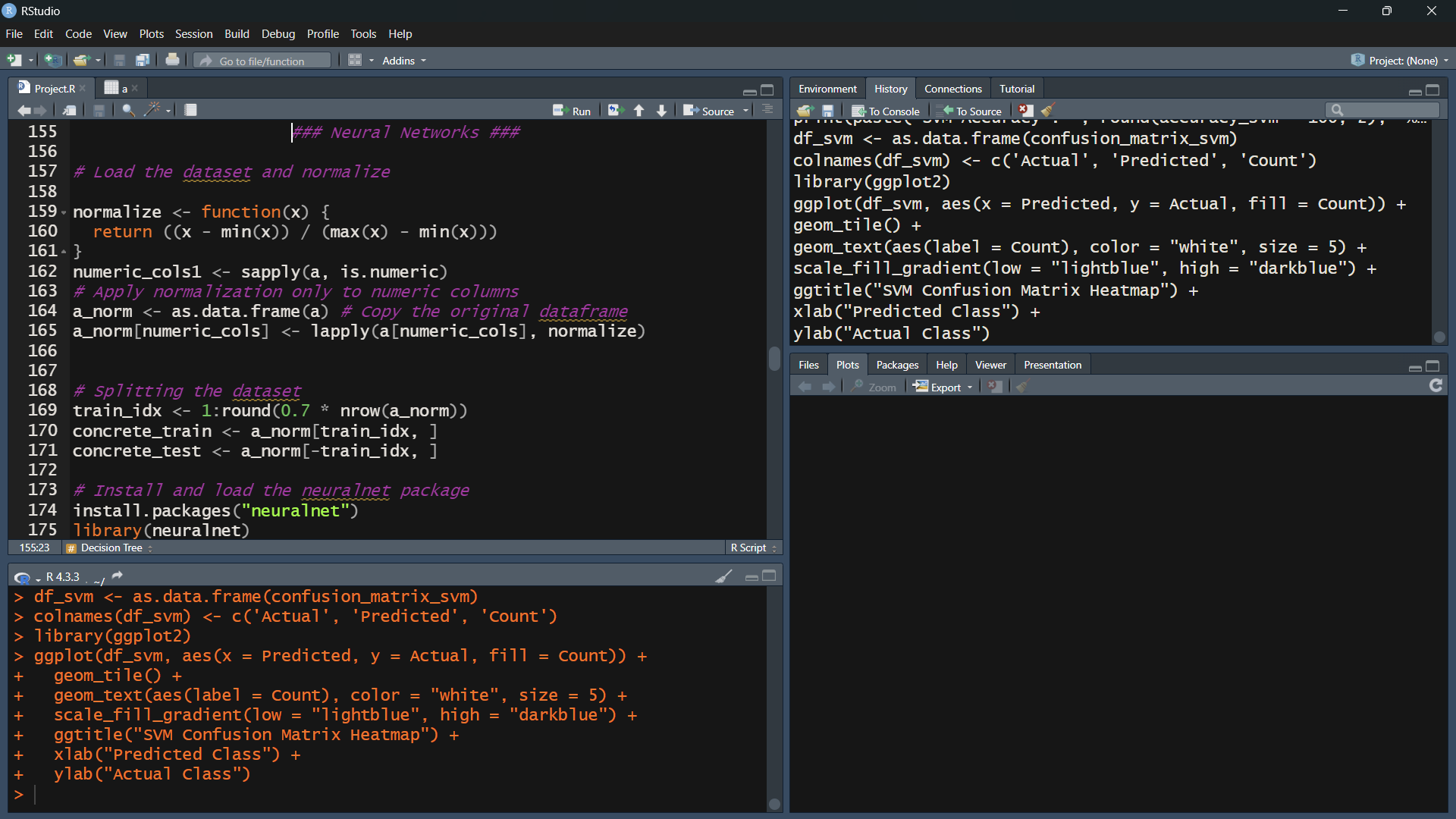


Fig 4.7: Neural Networks Code(part-1)

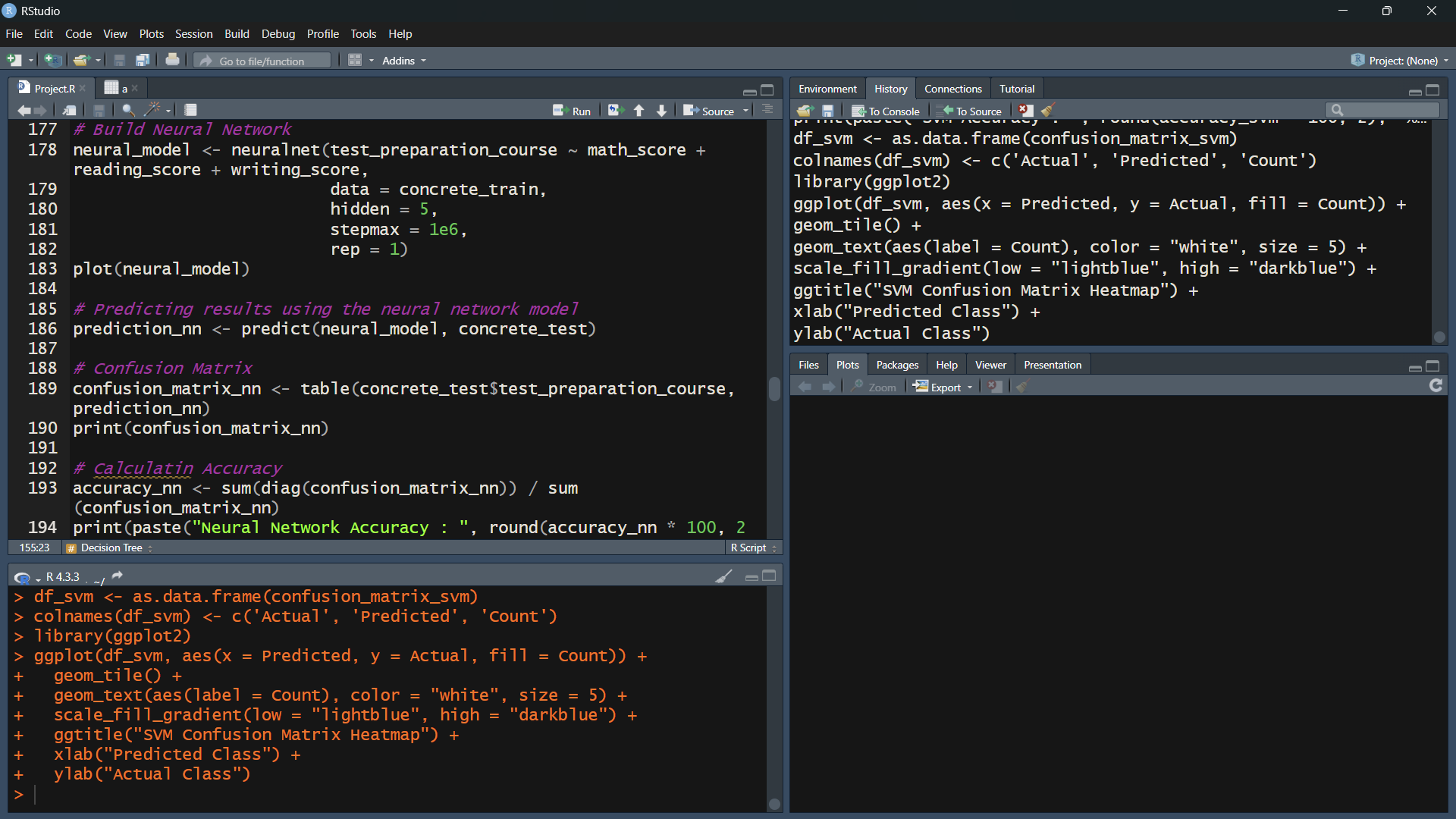


Fig 4.8: Neural Networks Code(part-2)

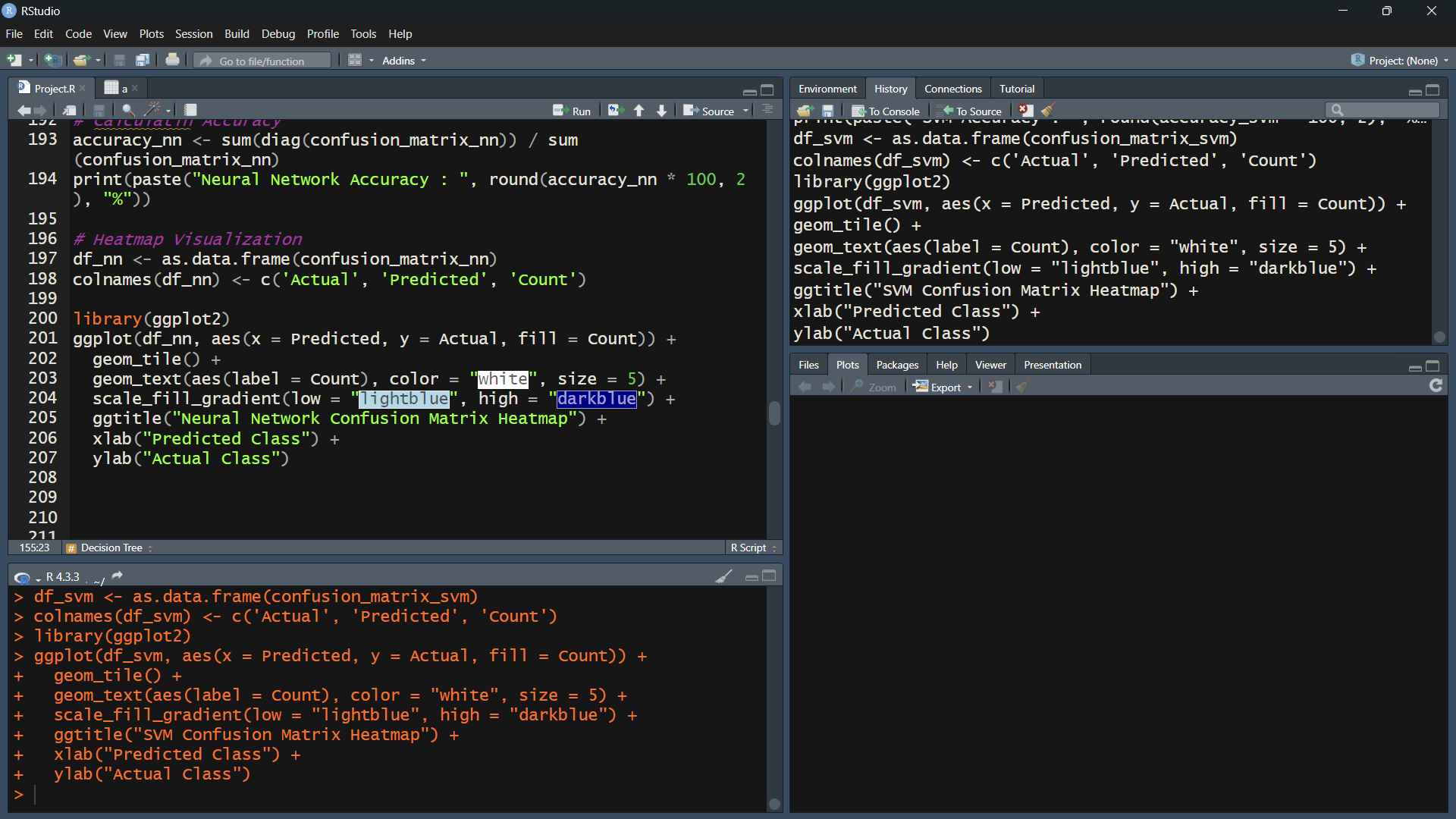


Fig 4.9: Neural Networks Code(part-3)

1. **K-Means Clustering:**

* **Data Preparation**: Remove the target column test\_preparation\_course since K-means clustering is unsupervised. Use only math\_score, reading\_score, and writing\_score.
* **Model Training**: Apply K-means clustering with 3 clusters, using a random seed for reproducibility.
* **Evaluation**: Visualize the clusters by plotting score distributions. Compare cluster assignments to actual test\_preparation\_course labels using a confusion matrix to assess whether clusters align with course participation. Visualize the confusion matrix with a heatmap.
* **Interpretation**: The clustering plot and confusion matrix help identify any meaningful grouping in the data, though accuracy is less relevant in this unsupervised context.

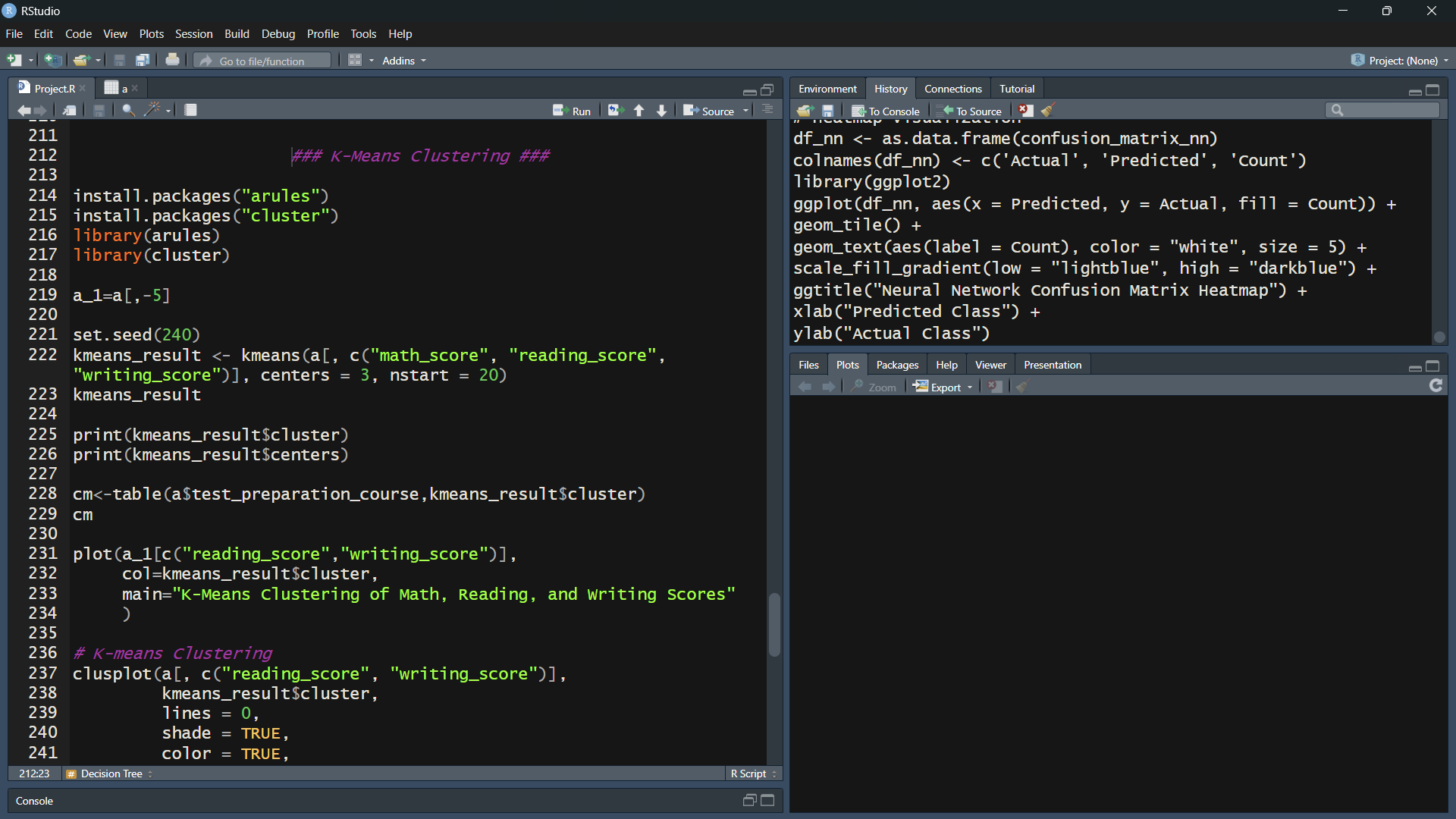


Fig 4.10: K-Means Clustering Code(part-1)

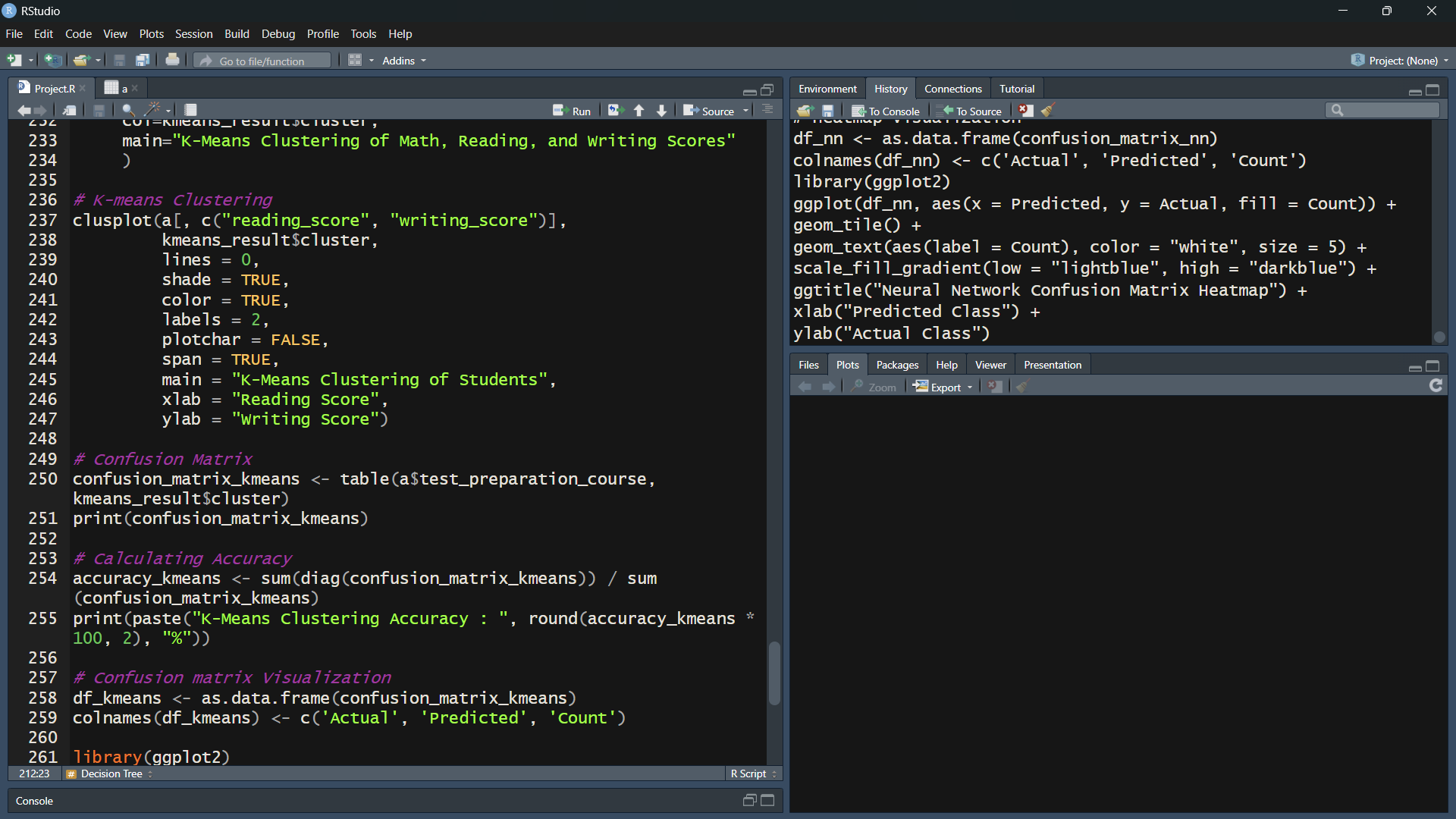


Fig 4.11: K-Means Clustering Code(part-2)

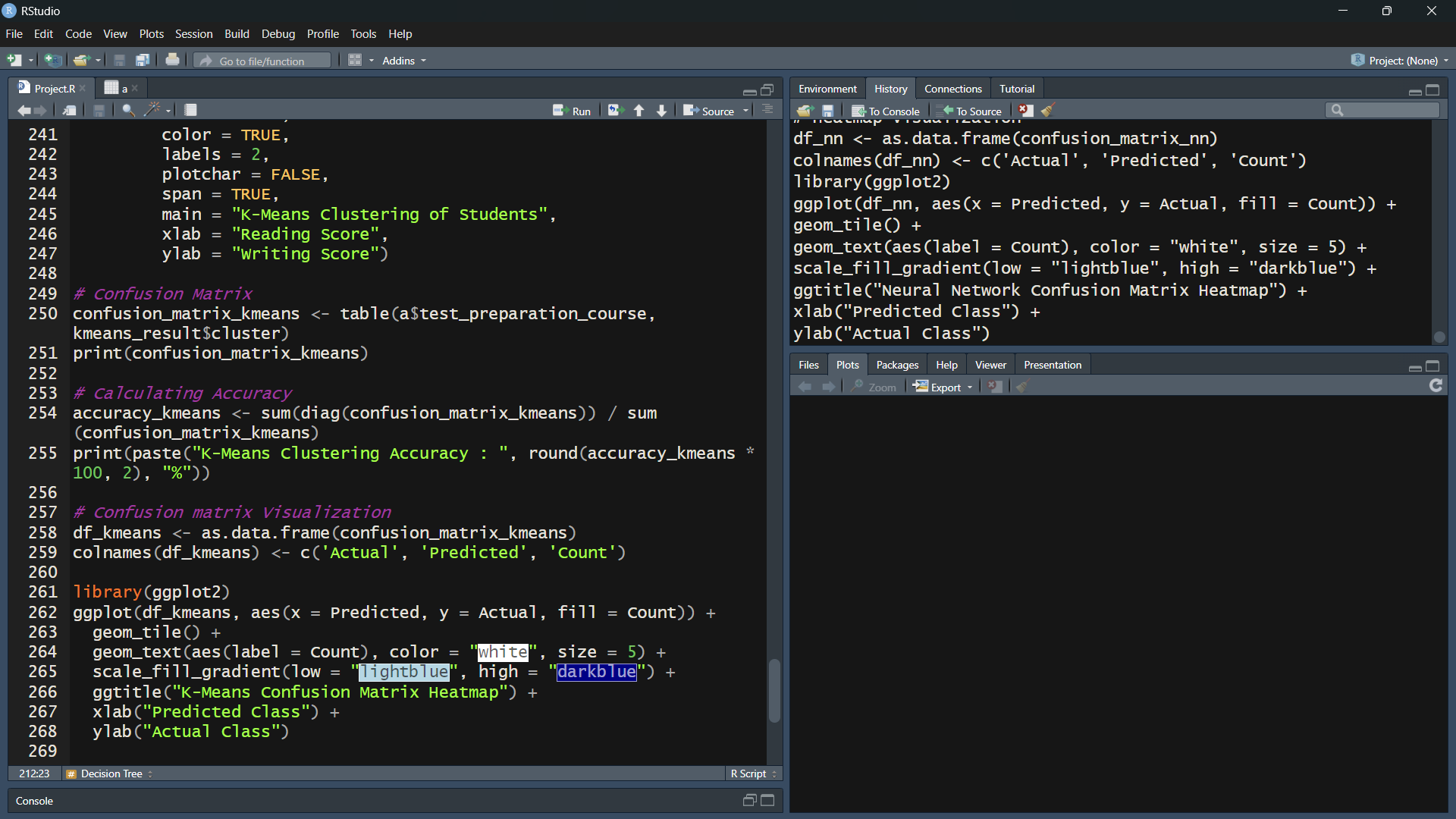


Fig 4.12: K-Means Clustering Code(part-3)

1. **Accuracy Comparison of All Models:**

* **Data Aggregation**: Collect accuracy scores from each model (Naive Bayes, Decision Tree, SVM, Neural Network, and K-Means).
* **Visualization**: Create a bar plot to display and compare the accuracy of each model, helping identify the most effective predictive model.

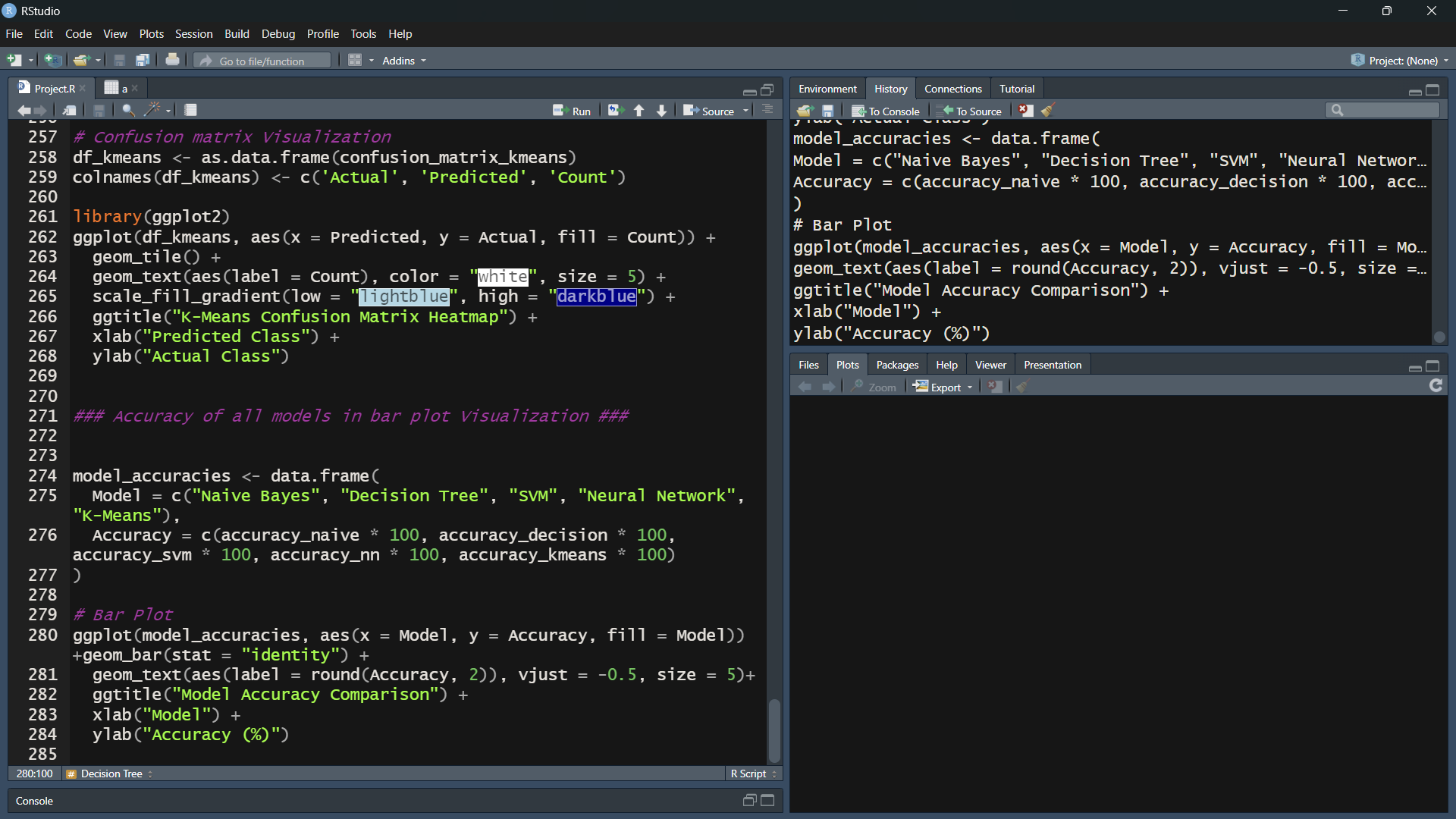


Fig 4.13: Models Accuracies

**EXISTING SYSTEM**

1. **Limited Predictive Accuracy**:

* Traditional methods, such as manual analysis or simple statistics, may fail to capture complex patterns in the data that could predict student behaviour effectively.
* Without advanced algorithms, predictions may be inaccurate, leading to students who need support being overlooked.

1. **Lack of Data-Driven Decision-Making**:

* Educational institutions often lack automated, data-driven insights into student performance and course preparation needs.
* Decisions based on intuition or basic metrics may not fully reflect each student's unique strengths and challenges.

1. **Inability to Identify Complex Patterns:**

* Simple statistical analyses may identify only basic trends but lack the ability to uncover deeper relationships between academic scores and test preparation.
* Machine learning can reveal correlations that are not immediately obvious through standard methods, such as how specific combinations of scores influence course completion.

**SOURCE OF DATASET**

* This Dataset is extracted from Kaggle.
* Dataset Link: https://www.kaggle.com/datasets/muhammadroshaanriaz/students-performance-dataset-cleaned

**ETL PROCESS**

The ETL process ensures that student performance data is accurately prepared and transformed to be ready for model training and evaluation. Here’s how the ETL steps are applied to this project:

**1. Extract:**

* Use R’s read.csv () function to load the data. Here, file.choose() is used to prompt for file selection in the provided code.
* After loading, the data is previewed with View() to inspect its structure, identify columns, and ensure all necessary data is present.

**2. Transform:**

* Convert test\_preparation\_course into a categorical (factor) variable, as it is the target variable for classification. This step is crucial for ensuring compatibility with machine learning algorithms in R.
* Convert test\_preparation\_course into a categorical (factor) variable, as it is the target variable for classification. This step is crucial for ensuring compatibility with machine learning algorithms in R.
* Use scale() to standardize scores for the SVM and Neural Network models.

**3. Load:**

* The transformed data is now stored in two main subsets: Training data and Testing data
* Each model’s outputs, including accuracy scores and confusion matrices, are stored in data frames. These data frames are used to create visualizations and calculate performance metrics.

**ANALYSIS ON DATASET**

**1. Naive Bayes Analysis:**

1. Introduction: The Naive Bayes model is used to predict the likelihood of a student’s test preparation course participation based on scores in math, reading, and writing. This probabilistic model is chosen for its simplicity and efficiency in handling categorical target variables.
2. General Description: Naive Bayes assumes independence between predictors (math, reading, and writing scores) and calculates the probability of a student participating in test preparation based on these scores.
3. Specific Requirements, Functions, and Formulas:

* Requirements: e1071 library for Naive Bayes model.
* Formula: Uses Bayes' theorem to calculate P (test\_preparation\_course | math\_score, reading\_score, writing\_score).
* Function: naive bayes (test\_preparation\_course ~ math\_score + reading\_score + writing\_score, data=training)

1. Analysis Results:

* The model generates a confusion matrix comparing predicted vs. actual participation status. Accuracy is calculated as the sum of true positive and true negative predictions divided by the total predictions.

1. Visualization:

* A heat map of the confusion matrix shows the model’s predictive accuracy.

**2. Decision Tree Analysis:**

1. Introduction: Decision Tree analysis is used to classify students’ likelihood of test preparation participation based on decision rules formed from score thresholds.
2. General Description: The decision tree algorithm splits data based on score thresholds to create “branches,” which make it easy to interpret the model’s decision-making process.
3. Specific Requirements, Functions, and Formulas:

* Requirements: rpart library for decision tree modeling, rpart.plot for visualization.
* Function: rpart (test\_preparation\_course ~ math\_score + reading\_score + writing\_score, data=training, method = "class")

1. Analysis Results:

* The decision tree produces a confusion matrix and an accuracy score based on the proportion of correct predictions.

1. Visualization:

* Tree visualization (rpart. plot) to show decision paths.
* Heatmap of confusion matrix to display the model’s accuracy.

**3. Support Vector Machine (SVM) Analysis:**

1. Introduction: Support Vector Machine (SVM) is employed to classify student participation by finding the optimal boundary (hyperplane) that separates students who participated in the test preparation course from those who did not.
2. General Description: SVM with a linear kernel is used to create a decision boundary in the score feature space, maximizing the margin between different classes.
3. Specific Requirements, Functions, and Formulas:

* Requirements: e1071 library for SVM implementation.
* Function: svm(formula = test\_preparation\_course ~ math\_score + reading\_score + writing\_score, data = training\_set, type = 'C-classification', kernel = 'linear')

1. Analysis Results: The SVM model produces a confusion matrix, showing the model’s performance, and an accuracy score based on correct classifications.
2. Visualization: Heatmap of confusion matrix to display the accuracy and distribution of predictions.

**4. Neural Network Analysis:**

1. Introduction: Neural Networks are applied to create a complex, layered model that can capture intricate patterns between score features and test preparation course participation.
2. General Description: The neural network consists of one hidden layer with five neurons, using a backpropagation algorithm to iteratively adjust weights and improve accuracy.
3. Specific Requirements, Functions, and Formulas:

* Requirements: neuralnet library for neural network model training.
* Normalization: Applied to math\_score, reading\_score, and writing\_score.
* Function: neuralnet(test\_preparation\_course ~ math\_score + reading\_score + writing\_score, data = concrete\_train, hidden = 5, stepmax = 1e6)

1. Analysis Results: The model yields a confusion matrix showing predicted vs. actual participation, along with an overall accuracy score based on the matrix’s diagonal elements.
2. Visualization: Neural network structure is visualized with a plot of the network’s layers and connections.

**5. K-Means Clustering Analysis:**

1. Introduction: K-means clustering groups students based on their performance across math, reading, and writing scores to identify similar student clusters, without using the test\_preparation\_course label.
2. General Description: This unsupervised learning technique groups students into three clusters, potentially revealing distinct performance patterns across score metrics.
3. Specific Requirements, Functions, and Formulas:

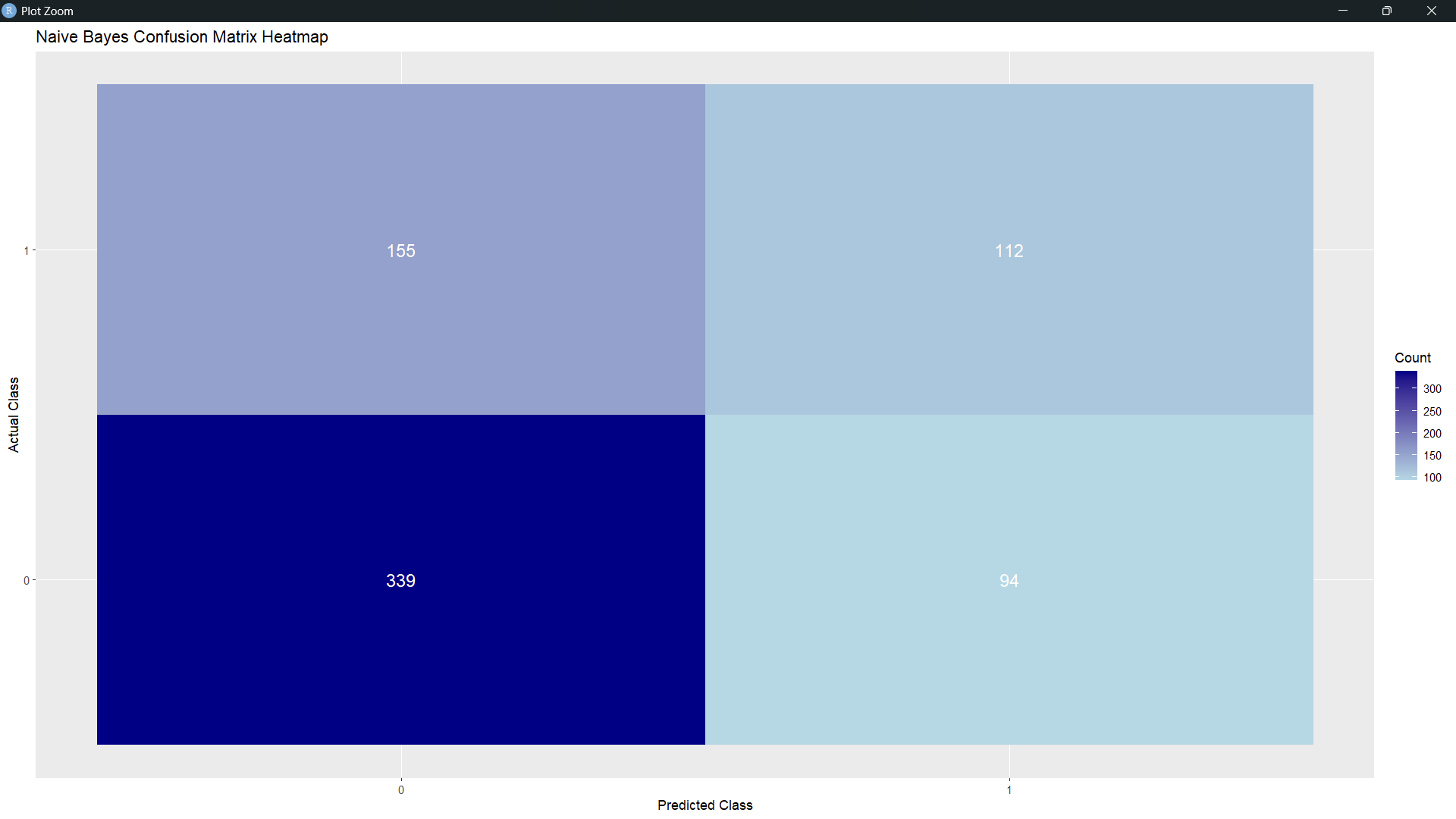
* Requirements: cluster library for clustering analysis.
* Function: kmeans(a[, c("math\_score", "reading\_score", "writing\_score")], centers = 3, nstart = 20)

1. Analysis Results: The clustering results in distinct groups that can be compared to the actual participation labels. A confusion matrix is used to assess how well clusters align with test preparation status.
2. Visualization: Scatter plots and cluster plots display clusters based on reading\_score and writing\_score.

**LIST OF ANALYSIS WITH RESULTS**

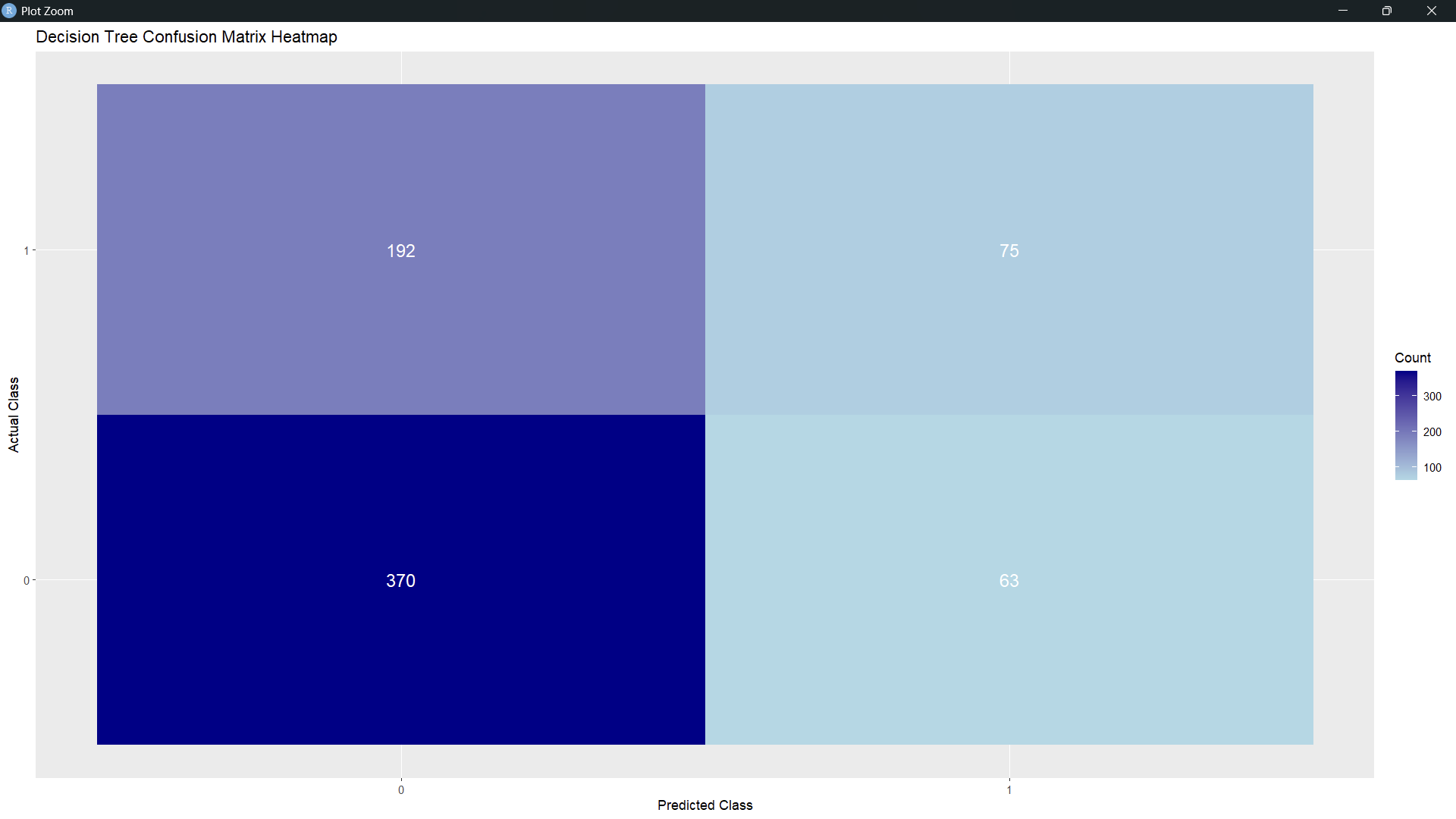
1. **Naive Bayes Classification:**

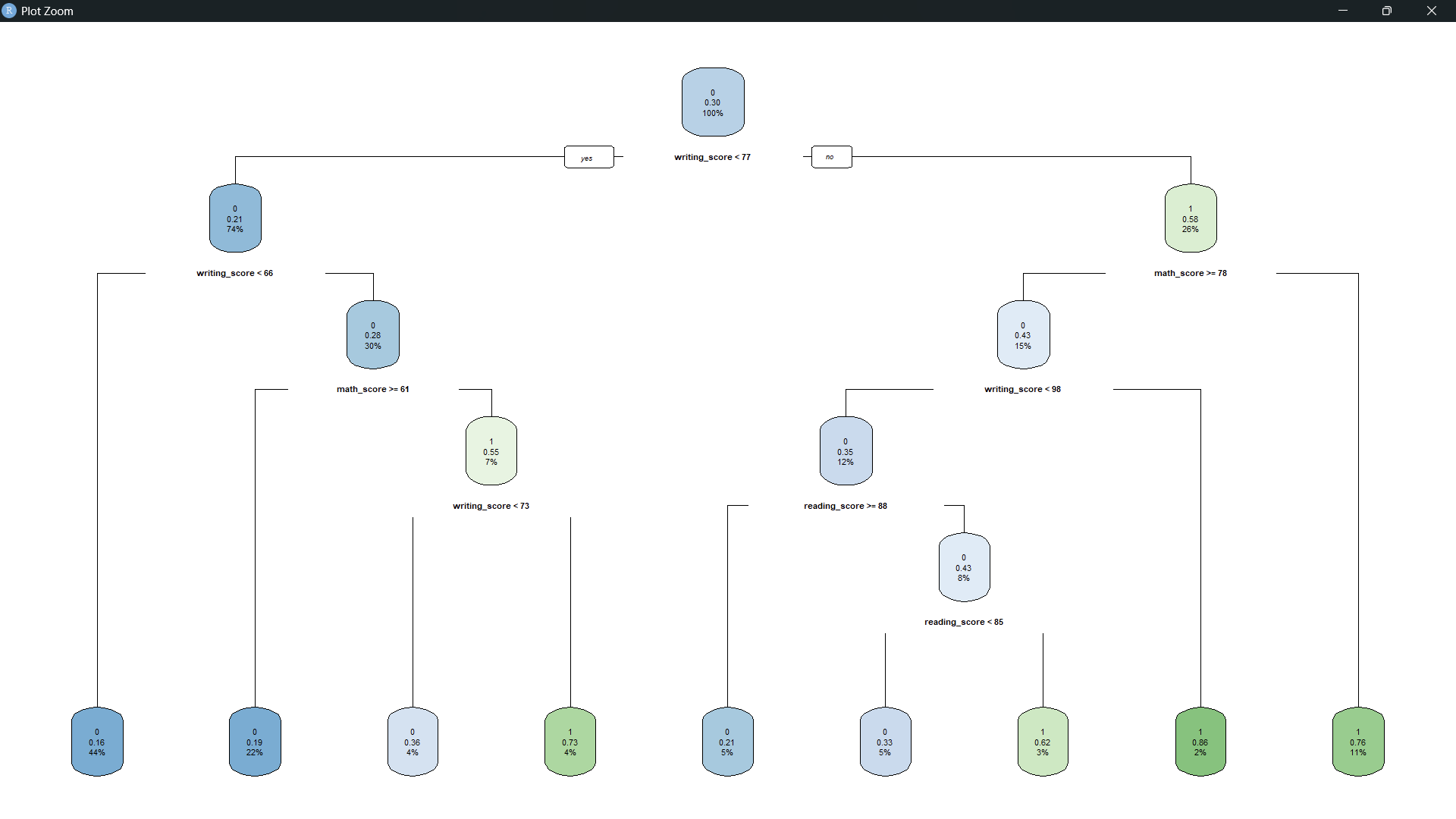
* Analysis: Naive Bayes, a probabilistic classifier, is used to predict a student's likelihood of participating in the test preparation course based on their math, reading, and writing scores. This method assumes independence among the predictors (scores) and uses Bayes’ theorem for classification.
* Results: The Naive Bayes model achieves an accuracy score of approximately 64.43%, showing its capability to predict participation status reasonably well, though it may struggle if predictors are not truly independent.
* Visualization: A heatmap of the confusion matrix displays the performance, highlighting correctly and incorrectly classified instances.



1. **Decision Tree Classification:**

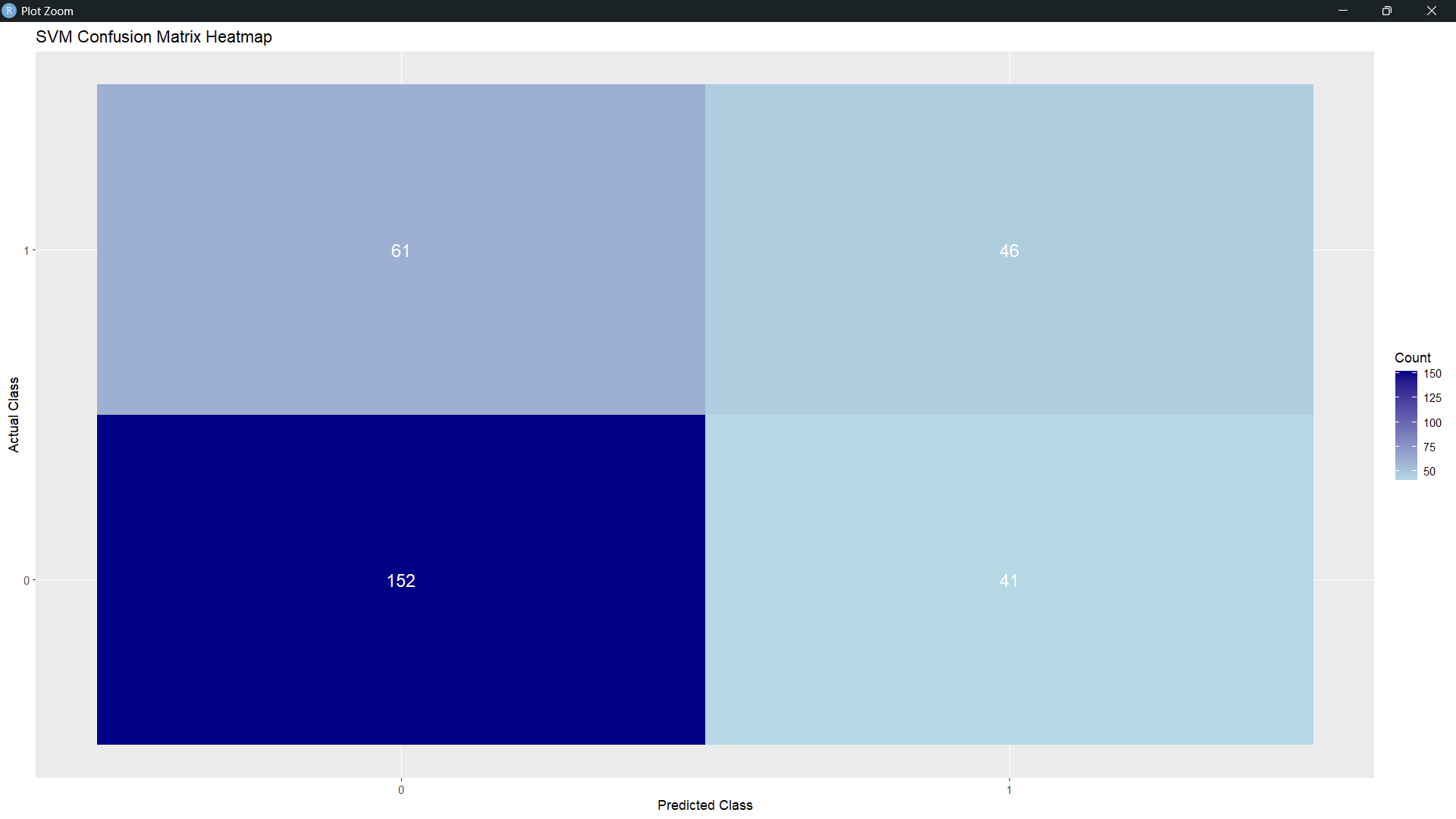
* Analysis: The Decision Tree model classifies students by splitting data at score thresholds, allowing for interpretable rules that show how score ranges relate to test preparation participation.
* Results: This model achieves an accuracy of about 63.7%, showing how well different score, ranges predict participation. The decision tree structure also reveals important thresholds, indicating which scores most influence participation decisions.
* Visualization: The tree structure is visualized, displaying decision paths and classification outcomes. Additionally, a heatmap of the confusion matrix summarizes accuracy.





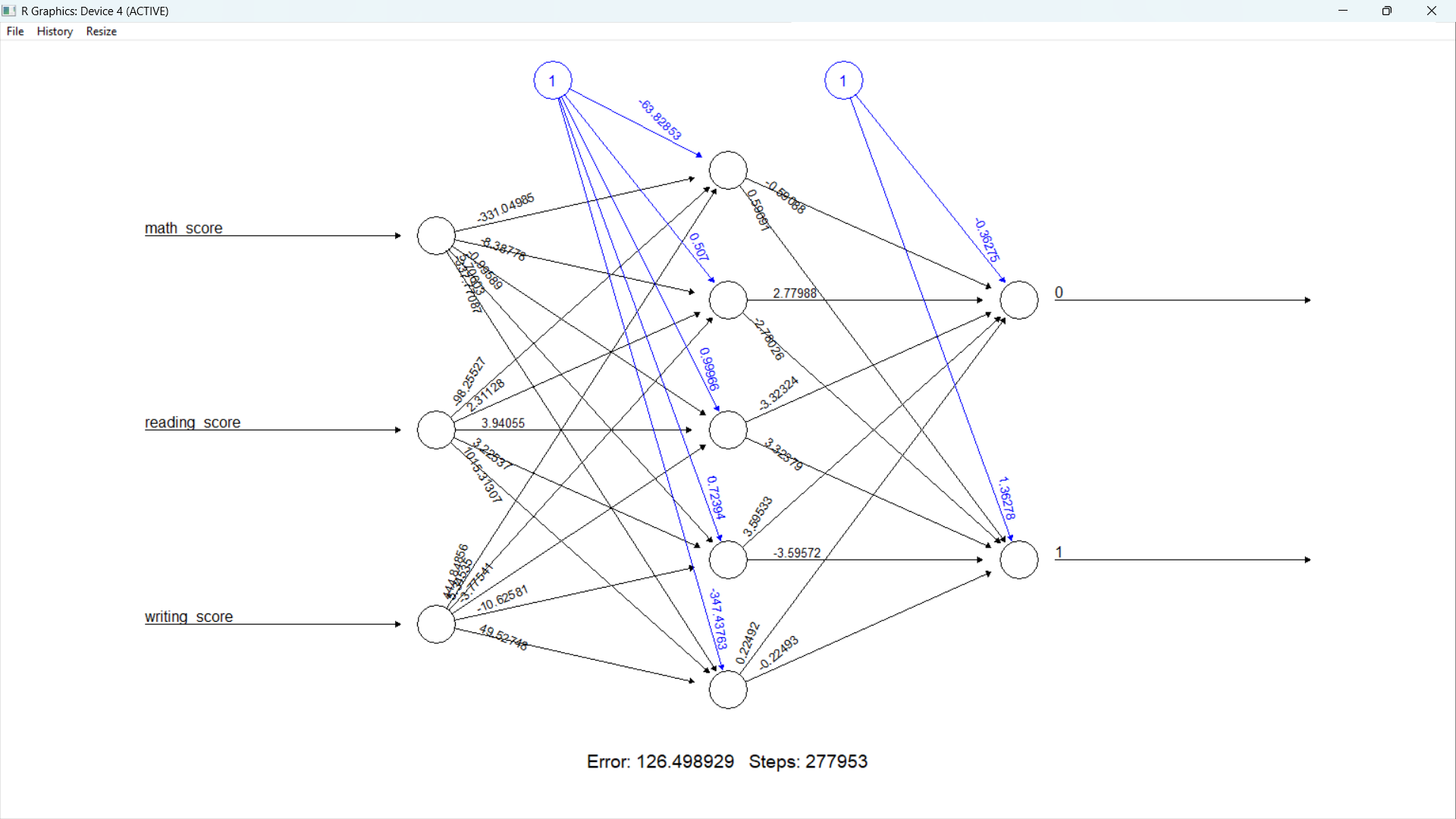
1. **Support Vector Machine (SVM) Classification:**

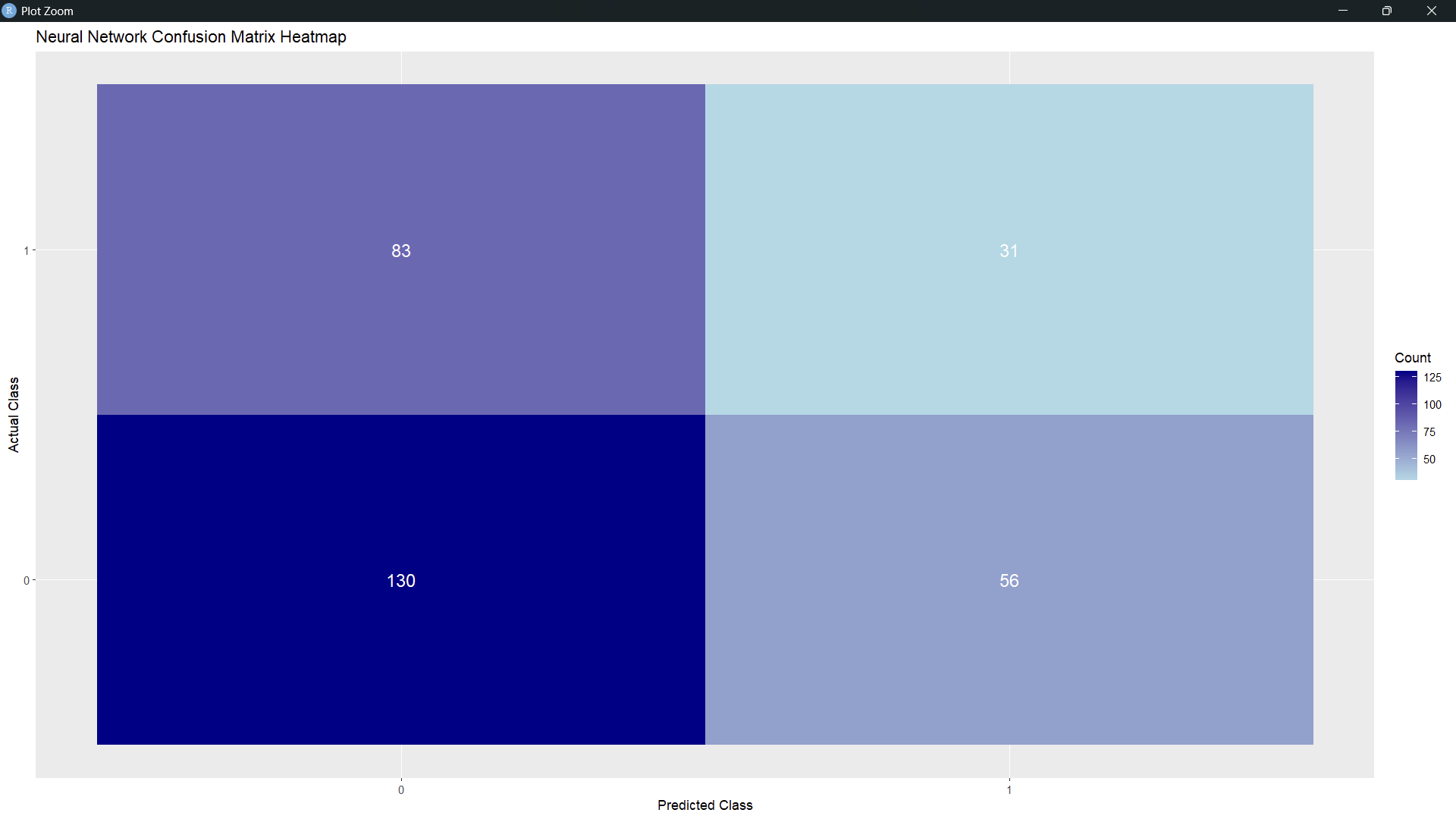
* Analysis: SVM is used to draw an optimal boundary between students who participated and those who did not, based on a linear kernel for simplicity. This model is sensitive to scaled features, so scores are standardized.
* Results: The SVM model achieves an accuracy of approximately 66%, performing well at distinguishing between classes but possibly limited by the linearity assumption.
* Visualization: A confusion matrix heatmap is created to visualize true and false classifications, emphasizing the model's overall effectiveness.



1. **Neural Network Classification:**

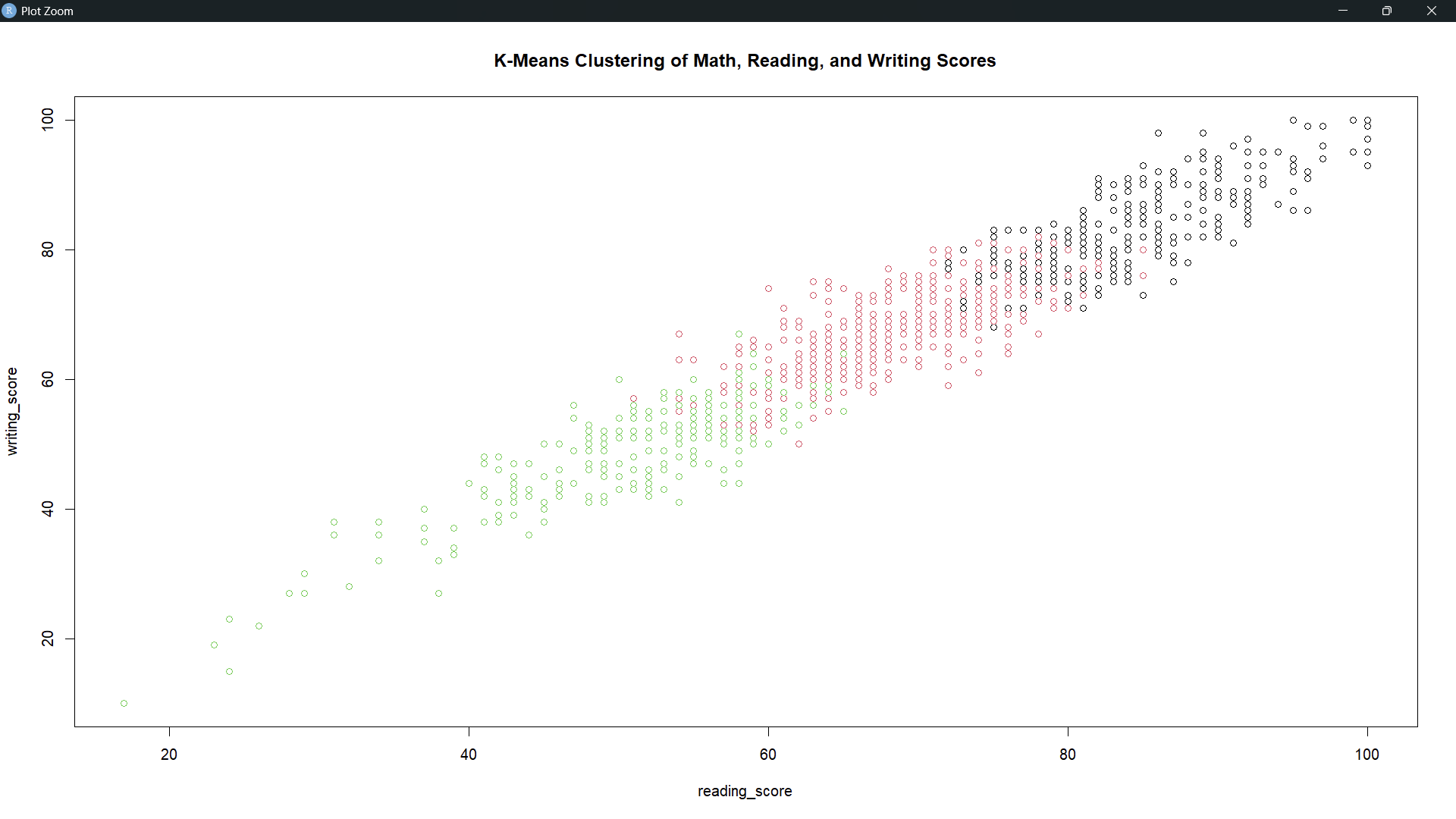
* Analysis: A neural network with one hidden layer (5 neurons) is applied to capture complex, non-linear relationships between scores and test preparation participation. Numeric scores are normalized to improve the model's learning stability.
* Results: The neural network model reaches an accuracy of 53.67%, performing well for complex relationships but possibly overfitting slightly due to its capacity for capturing intricate patterns.
* Visualization: The neural network structure is plotted to visualize layers and neurons, and a heatmap of the confusion matrix provides a clear view of classification performance.

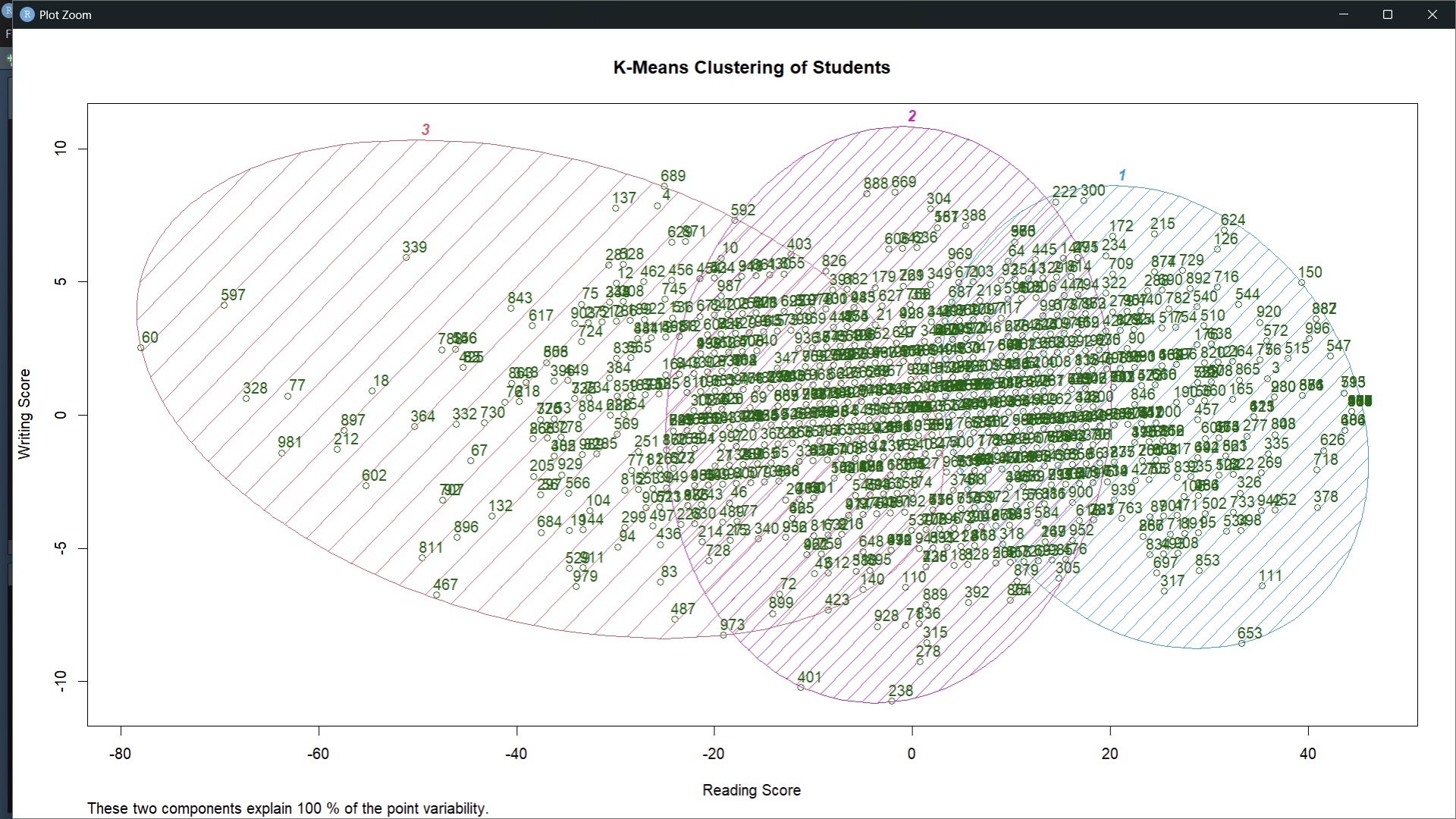


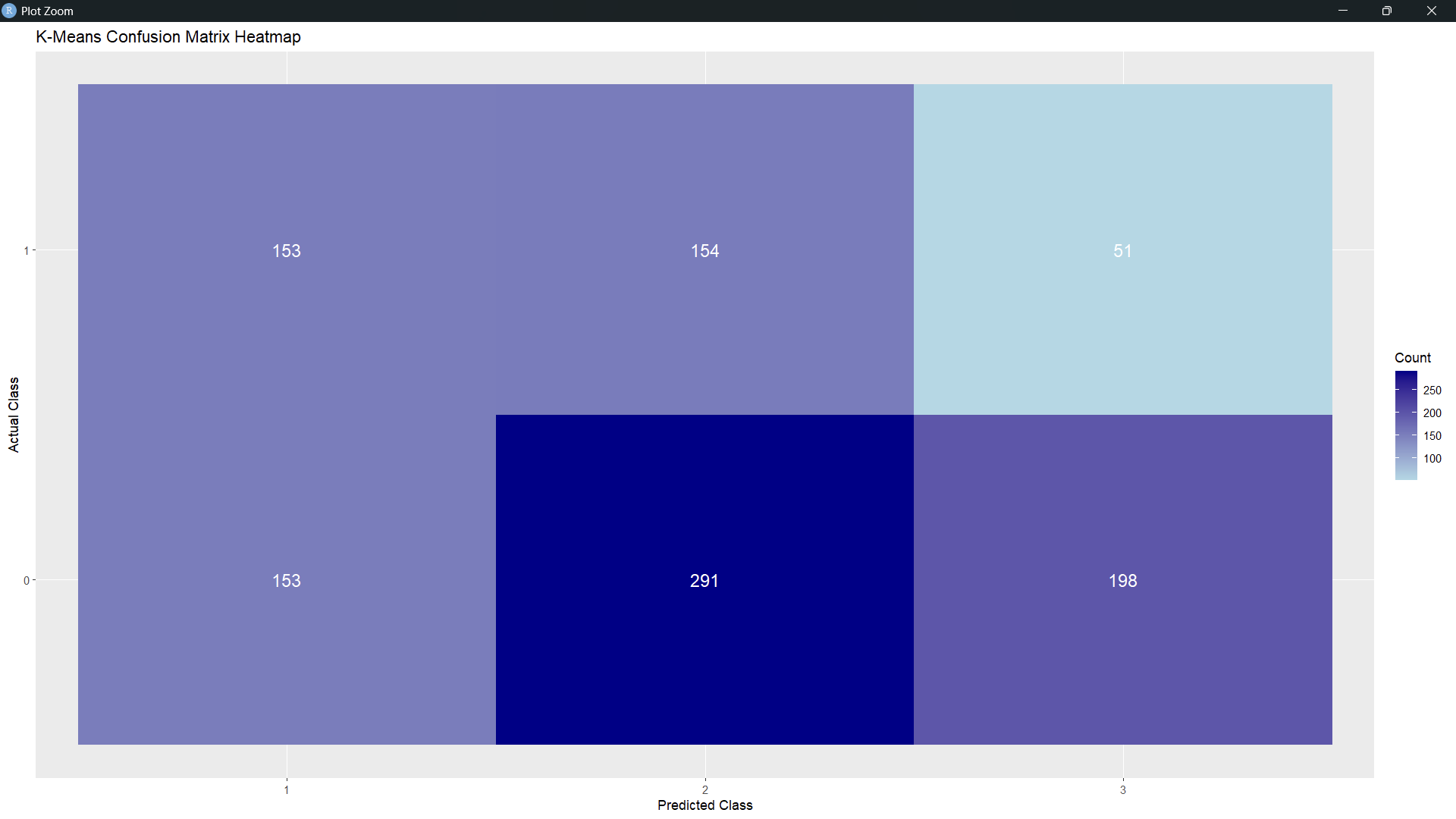


1. **K-Means Clustering:**

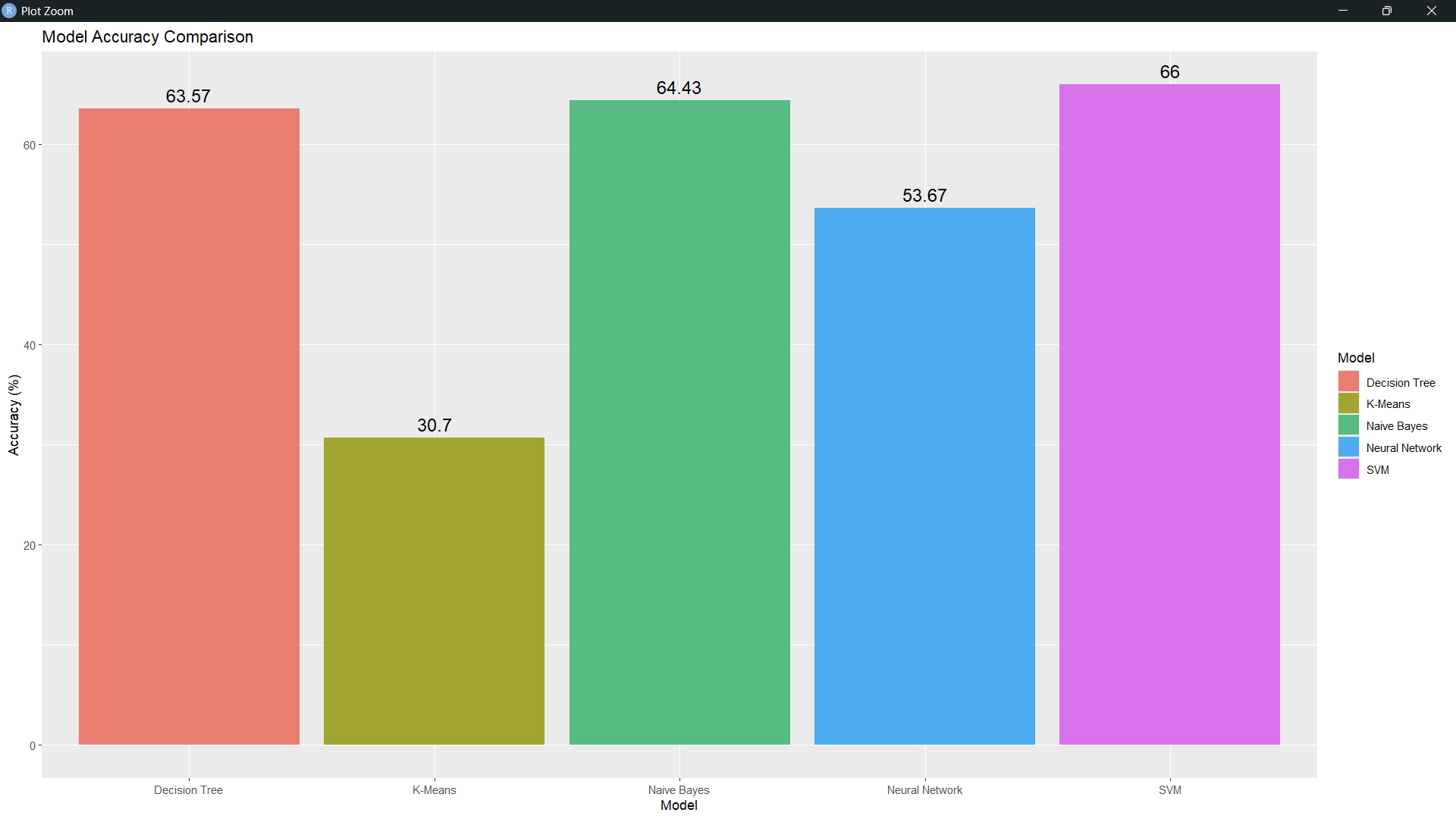
* Analysis: K-Means Clustering groups students into three clusters based on score similarities. This unsupervised approach helps identify natural groupings and compares them with test preparation participation status to understand different performance levels.
* Results: The clustering aligns with participation to a certain extent, as shown in the confusion matrix, but naturally has lower accuracy due to the lack of a supervised approach. This analysis reveals underlying score-based groupings rather than direct classification.
* Visualization: Scatter plots and cluster plots show the distribution of clusters in score space, and a confusion matrix heatmap highlights cluster overlap with actual participation.







1. **Model Comparison Analysis:**
   * Analysis: Accuracy results from each model (Naive Bayes, Decision Tree, SVM, Neural Network, and K-Means) are compiled for comparison to determine which method best predicts student participation in test preparation.
   * Results: The highest-performing model (fill with name and accuracy) shows the best predictive accuracy, while other models provide insights into different aspects of score relationships.
   * Visualization: A bar plot displays accuracy scores for each model, providing a visual comparison that highlights the most effective model for this dataset.



**FUTURE SCOPE**

* The current project explores Naive Bayes, Decision Tree, SVM, Neural Network, and K-Means Clustering. Future work could focus on enhancing model accuracy by fine-tuning hyperparameters for each model (e.g., adjusting tree depth in Decision Trees, kernel type in SVM, and hidden layers in Neural Networks).
* The Neural Network used here is simple, with a single hidden layer. Expanding the network architecture to include additional layers or using Convolutional or Recurrent Neural Networks (if time-series data or sequence data is available) could help capture more intricate relationships between the scores.
* The current analysis uses math, reading, and writing scores as predictors. Adding new features, such as demographic information, previous academic history, or socio-economic status, may improve model accuracy and provide deeper insights.
* Each model produces a heatmap of the confusion matrix and other static visualizations and could make it easier for educators and administrators to interpret predictions and make informed decisions.
* Adding a voting ensemble that combines predictions from Naive Bayes, Decision Trees, SVM, and Neural Networks could increase prediction robustness. By averaging or taking the majority vote across models, this ensemble approach may yield higher accuracy, balancing each model’s strengths and weaknesses.

**CONCLUSION**

This project demonstrates a comprehensive approach to predicting student test preparation needs and understanding factors affecting academic performance. By leveraging machine learning models—including Naive Bayes, Decision Trees, Support Vector Machines, Neural Networks, and K-Means Clustering. the analysis provides valuable insights into students’ performance in math, reading, and writing.

The findings from each model reveal patterns in the data, with classification models identifying key factors associated with students who participate in test preparation. Visualizations, including heatmaps and accuracy comparisons, provide an intuitive understanding of model performance and predictive accuracy, allowing for comparisons between different approaches. Despite varying levels of accuracy, each model contributes unique insights and confirms the potential of data-driven strategies to support educational interventions.

While the current project is a solid foundation for student performance prediction, it also highlights areas for improvement and further exploration. The future scope includes enhancing model accuracy, incorporating advanced techniques, adding new features, and making the system more interactive and scalable. By pursuing these improvements, this project could become an essential tool for educators and policymakers, helping them proactively identify and support students who may benefit from additional academic resources.

In summary, this project illustrates the power of predictive analytics in education, providing a framework for future developments aimed at enhancing educational outcomes through data-driven insights.

**REFERENCES**

[1] Han, J. 2012. Micheline Kamber, & Jian Pei.”Data Mining: Concepts and Techniques”

[2] Aziz, A. A., Ismail, N. H., & Ahmad, F. 2013.” Mining Stu-dents’ Academic Performance” inJournal of Theoretical & Applied Information Technology,53(3)

[3] Ahmad, F., Ismail, N. H., & Aziz, A. A. 2015. ”The Pre-diction of Students’ Academic Performance Using Classiﬁca-tion Data Mining Techniques.” inApplied Mathematical Sci-ences,9(129), 6415-6426

[4] El-Halees, A. 2009. Mining Student’s Data to Analyze E-Learning Behavior: A Case Study

[5] Tair, M. M. A., & El-Halees, A. M. 2012.” Mining Educa-tional Data to Improve Students’ Performance: A Case Study” in International Journal of Information,2(2), 140-146

[6] Garc´ıa, E. P. I., & Mora, P. M. 2011.” Model Prediction of Academic Performance for First Year Students”. InArtiﬁcial Intelligence (Micai), 2011 10th Mexican International Con-ference (pp. 169-174). IEEE

[7] Bhardwaj, B. K., & Pal, S.2012.” Data Mining: A Predictionfor Performance Improvement Using Classiﬁcation”.Arxiv Preprint Arxiv:1201.3418

[8] Al-Radaideh, Q. A., Al-Shawakfa, E. M., & Al-Najjar, M.I. 2006.” Mining Student Data Using Decision Trees.” In-International Arab Conference on Information Technology, Yarmouk University, Jordan.

[9] Lakshmi, D., Arundathi, S., & Jagadeesh, D. 2014.” Data Mining: A Prediction for Student’s Performance Using De-cision Tree Id3 Method”

[10] Ali, M. M. 2013. ”Role of Data Mining in Education Sec-tor” in International Journal of Computer Science and Mobile Computing,2(4), 374-383

**Github Link**: <https://github.com/RishithChundru/Predictive-Analysis-Project>

**LinkedIn Post**: <https://www.linkedin.com/feed/update/urn:li:ugcPost:7263589037339230208/>

**BIBLIOGRAPHY**

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.
2. Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer.
3. Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media.
4. Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer.
5. Teetor, P. (2011). *R Cookbook: Proven Recipes for Data Analysis, Statistics, and Graphics*. O'Reilly Media.
6. Kabacoff, R. I. (2011). *R in Action: Data Analysis and Graphics with R*. Manning Publications.
7. Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
8. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
9. Quinlan, J. R. (1986). *Induction of Decision Trees*. Machine Learning, 1(1), 81–106.
10. Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.