PREDICTIVE ANALYSIS PROJECT REPORT
(Project Semester July-December 2024)

Name: P.Saimadhav Section:

K21BG

Roll no: RK21BGB63

Subject: PredictiveAnalytics

Task No: CA-03

Submitted to: Mr. Pardeep Kumar

Under the Guidance of:

MR.PARDEEP KUMAR

Discipline of CSE/IT
Lovely School of Computer Science & Engineering Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that PERUMANDLA SAIMADHAV bearing Registration no. 12111663 has completed INT234 project titled, “CUSTOMER CHURN DETECTION USING MACHINE LEARNING ALGORITHMS” under
my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor: Designation of the Supervisor:
School of Computer Science & Engineering Lovely Professional University Phagwara, Punjab.

Date: 10/11/2024

DECLARATION

I, PERUMANDLA SAIMADHAV, student of DATA SCIENCE DOMAIN 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: 10/11/2024

Registration Number. 12111544

Name of the Student:Perumandla Saimadhav

ACKNOWLEDGEMENT

We take this opportunity to present our votes of thanks to all those who guidepost really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

We are really grateful to Mr.Pardeep Kumar for providing us with an opportunity to undertake this project and providing us with all facilities. I am highly thankful to sir for his active support, valuable time and advice, whole- hearted guidance, sincere cooperation and pains taking involvement during the study and in completing the assignment and preparing the assigned project within the time stipulated.
Lastly, I am thankful to those, particularly the various friends, who have been instrumental in creating proper, healthy and conductive environment and including new and innovative ideas for us during the project, without their help, it would have been extremely difficult for us to prepare the project within the time.

Introduction:

This code performs a comprehensive analysis of customer churn data, employing various machine learning and statistical techniques to explore, classify, and predict customer churn outcomes. It begins by loading essential libraries, such as caret, randomForest, ggplot2, and reshape2, which support data manipulation, visualization, and modeling tasks.

The code leverages classification techniques to categorize customers as churned or retained based on factors like demographics, subscription type, and satisfaction scores. Regression analysis is applied to understand the impact of variables such as monthly charges or subscription length on churn probability. Additionally, clustering methods group customers with similar churn risks, providing insights for targeted retention strategies.

This code demonstrates a robust approach to exploring and modeling churn data, combining classification, regression, and clustering techniques to gain a holistic understanding of the factors driving customer churn.

Dataset Used in the Project: Customer Churn Prediction

About the Dataset:

This dataset is designed to help businesses understand and predict customer churn—when a customer decides to discontinue their subscription or service. Churn prediction is critical for subscription-based businesses like telecommunications, streaming services, and SaaS companies, where customer retention is a primary goal to maintain recurring revenue and long-term customer lifetime value.

Key Reasons for Using this Dataset :

Identifying At-Risk Customers: By examining factors that contribute to customer dissatisfaction or switching behavior, companies can identify high-risk customers and develop targeted interventions to improve satisfaction and prevent churn.

1. **Improving Retention Strategies**: The dataset can reveal trends in customer demographics, usage, and satisfaction, informing companies about common reasons for churn. This allows businesses to tailor their retention strategies (e.g., offering discounts or better support) to keep customers engaged.

2. **Enhancing Customer Experience:** Insights from this dataset can pinpoint where the customer experience may be lacking, allowing companies to make informed changes that align with customer preferences, reducing the likelihood of churn in the future.

Attributes of the Dataset

1. **CustomerID:** A unique identifier for each customer, which allows tracking individual churn

history.

2. **Gender:** The customer's gender, which can be used to assess if churn rates differ across gender groups.

3. **Age:** The customer's age, a demographic feature that may indicate churn patterns among different age groups.

4. **Subscription_Length_Months:** Number of months the customer has been subscribed. Longer subscriptions may correlate with customer loyalty.

5. **Monthly_Charges_USD:** The monthly fee paid by the customer, which could influence churn if customers perceive fees as too high relative to service value.

6. **Contract_Type:** Type of subscription contract, such as Monthly or Yearly, which often affects customer retention, as longer contracts can reduce churn.

7. **Internet_Service:** Type of internet service used (e.g., DSL, Fiber), which can impact satisfaction based on service quality.

8. **Customer_Support_Calls:** Number of calls made to customer support, which might correlate with dissatisfaction, as higher calls can indicate unresolved issues.

9. **Satisfaction_Score:** Rating given by customers on a scale of 1 to 10, where lower satisfaction scores are often linked with higher churn rates.

10. **Churn:** The target variable, indicating whether the customer has churned (1) or stayed (0). This is the primary variable we aim to predict, using the other features

Algorithm's used in the project:

The project is implemented by variety of machine learning and statistical algorithms, each serving a distinct purpose in analysing and modelling the data.

1.K-Nearest Neighbors (KNN)

- **Instance-Based Learning**: KNN is a non-parametric, instance-based algorithm that classifies data points based on their proximity to other points in the dataset.
- **Distance Metric**: KNN uses distance metrics (e.g., Euclidean, Manhattan) to find the 'k' closest data points to a new instance, and the majority class among these neighbors determines the classification.
- **K-Value Choice**: The choice of 'k' is crucial; too small may make it sensitive to noise, while too large might dilute the influence of closer points.
- **Easy to Understand**: KNN is conceptually simple and easy to implement, making it a popular choice for beginners in classification tasks.
- **No Training Phase**: KNN has no explicit training phase; it stores the entire dataset and performs calculations only at prediction time, which can be computationally heavy.
- **Sensitive to Data Scaling**: KNN requires normalized or scaled data for optimal performance, as different scales can disproportionately impact distance calculations.

2. Support Vector Machine (SVM)

- **Hyperplane-Based Classification**: SVM is a supervised learning algorithm that finds the optimal hyperplane to separate data points of different classes with maximum margin.
- **Kernel Trick**: SVM can use different kernels (e.g., linear, polynomial, RBF) to transform data into higher dimensions, allowing complex, non-linear boundaries between classes.
- **Effective for High-Dimensional Spaces**: SVM performs well with high-dimensional data and is effective when there's a clear margin of separation between classes.

- **Robust to Overfitting**: SVM focuses on maximizing the margin, which can reduce the risk of overfitting, especially in higher-dimensional spaces.
- **Memory-Intensive**: SVM requires substantial memory and processing power, particularly for large datasets, as it involves complex optimization processes.
- **Flexible for Classification and Regression**: SVM can be adapted for both classification and regression tasks, making it versatile across applications.

3. Random Forest

- **Ensemble of Decision Trees**: Random Forest is an ensemble method that creates a large number of decision trees and aggregates their predictions for classification or regression.
- **Bootstrap Aggregation (Bagging)**: It uses bagging, where multiple trees are trained on random subsets of data, reducing variance and increasing robustness.
- **Feature Randomness**: Random Forest selects a random subset of features for each tree, helping reduce overfitting and improving generalization.
- **Highly Accurate and Scalable**: Random Forest is known for its accuracy and scalability, making it effective for large datasets.
- **Less Prone to Overfitting**: The ensemble nature of Random Forest reduces overfitting compared to individual decision trees, improving model performance.
- **Variable Importance**: It provides feature importance scores, helping identify which features contribute most to the prediction, which can be useful for understanding model behavior.

4. Naive Bayes

- **Probabilistic Classifier**: Naive Bayes is based on Bayes' theorem, calculating the probability of a class given the features, assuming independence among features.
- **Strong Independence Assumption**: Despite assuming that features are independent (which is often not true in practice), Naive Bayes can still perform well in many scenarios.
- **Fast and Efficient**: Naive Bayes is computationally efficient, with a low training cost, making it suitable for real-time applications and large datasets.
- **Works Well with Categorical Data**: Naive Bayes is particularly effective with categorical data, such as text classification, where feature independence assumptions may hold.
- **Handles High-Dimensional Data**: Naive Bayes can handle high-dimensional data well, as it does not require complex computations for feature interactions.
- **Sensitive to Data Distribution**: Its performance depends on the distribution of data; if features are highly correlated, Naive Bayes may not perform as well due to the independence assumption.

Performance comparison:

1. **K-Nearest Neighbors (KNN):** Achieved 75.67% accuracy by classifying customers based on the proximity to other customers with similar attributes. This model performed slightly better compared to the other models.
2. **Support Vector Machine (SVM):** SVM achieved an accuracy of 75.20% using a linear kernel, indicating it can reasonably separate the churn and non-churn classes but did not outperform KNN significantly.
3. **Random Forest:** The model uses an ensemble approach and produced an accuracy of 75.20%,

similar to SVM, showing robust

4. **Naive Bayes:** This model, based on Bayes' theorem with an assumption of feature independence, achieved an accuracy of 75.33%. Despite its simplicity, Naive Bayes performed comparably with more complex models like K-Nearest Neighbors and Random Forest, showcasing its effectiveness with datasets where probabilistic assumptions hold. This model's robustness and efficiency make it suitable for quick predictions with minimal computation, though it may be limited by its independence assumption in more intricate data pattern

Comparison of Algorithms

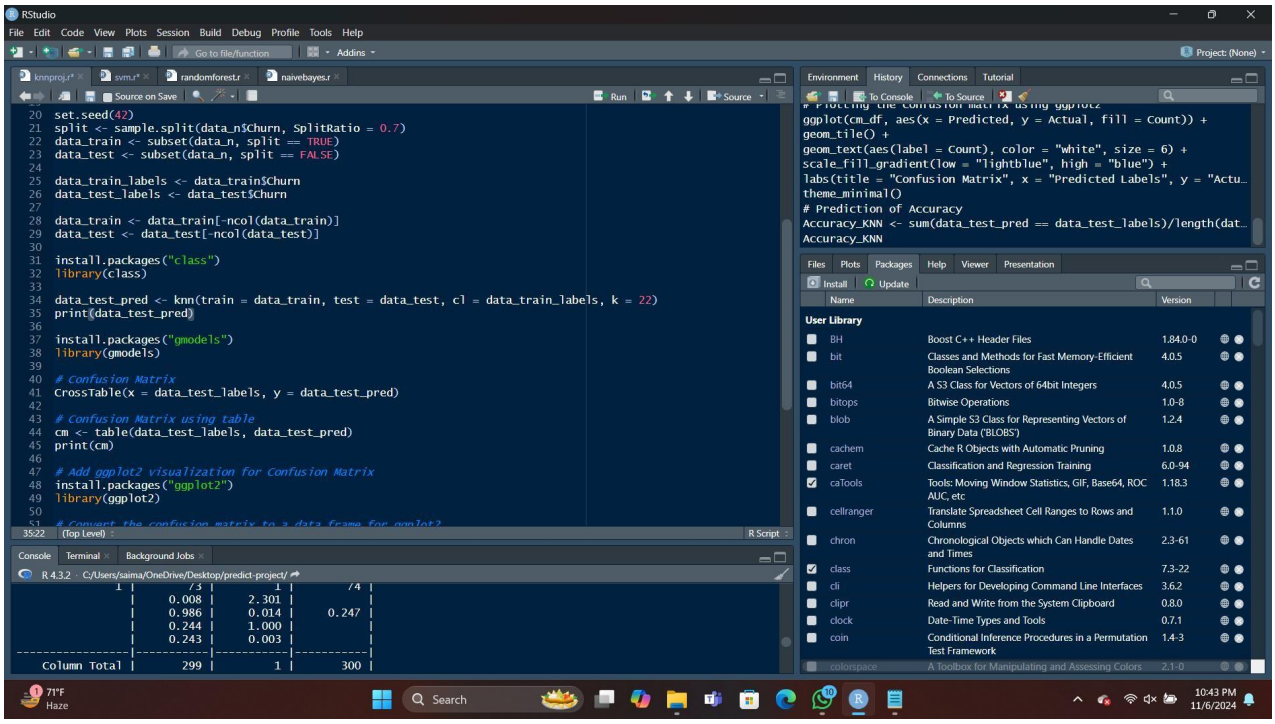
Model	Description	Accuracy (%)
K-Nearest Neighbors	Classification based on nearest data points	75.67
Support Vector Machine	Finds an optimal separating hyperplane.	75.20
Random Forest	Ensemble method using multiple decision trees.	75.20
Naive Bayes	Based on Bayes' theorem with independence assumptions	75.20

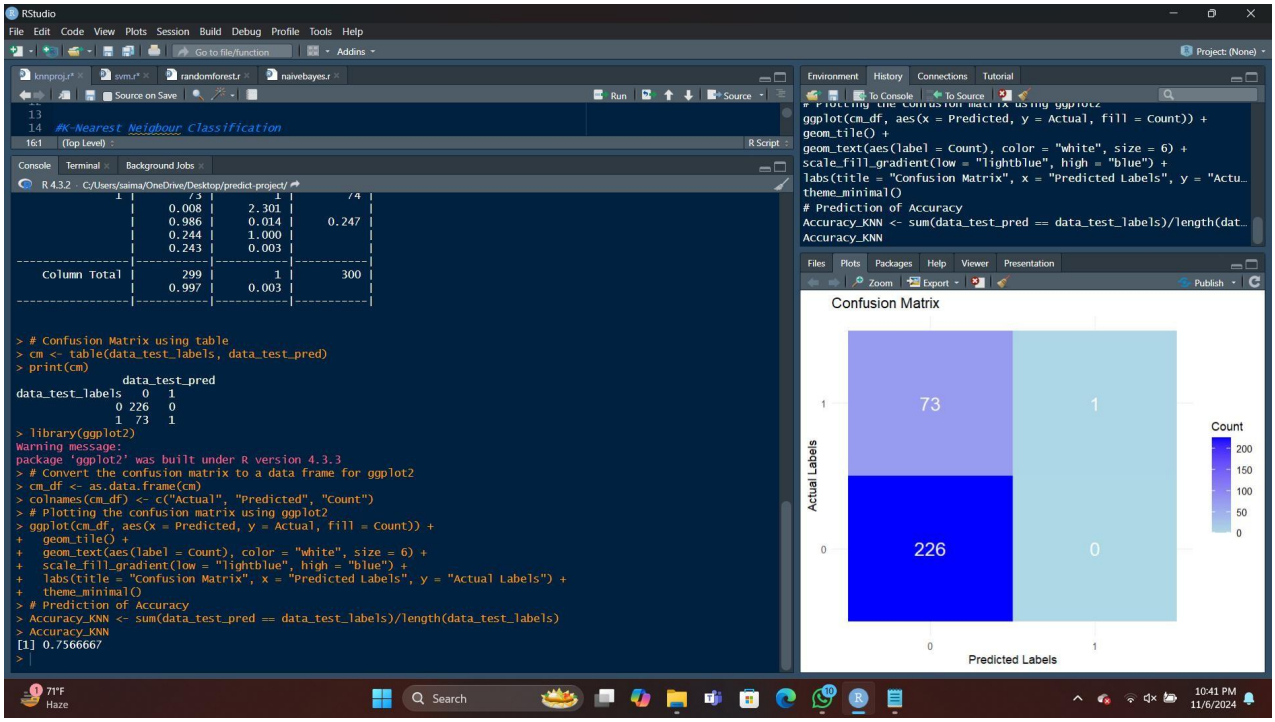
Implementation:

Problem Statement:

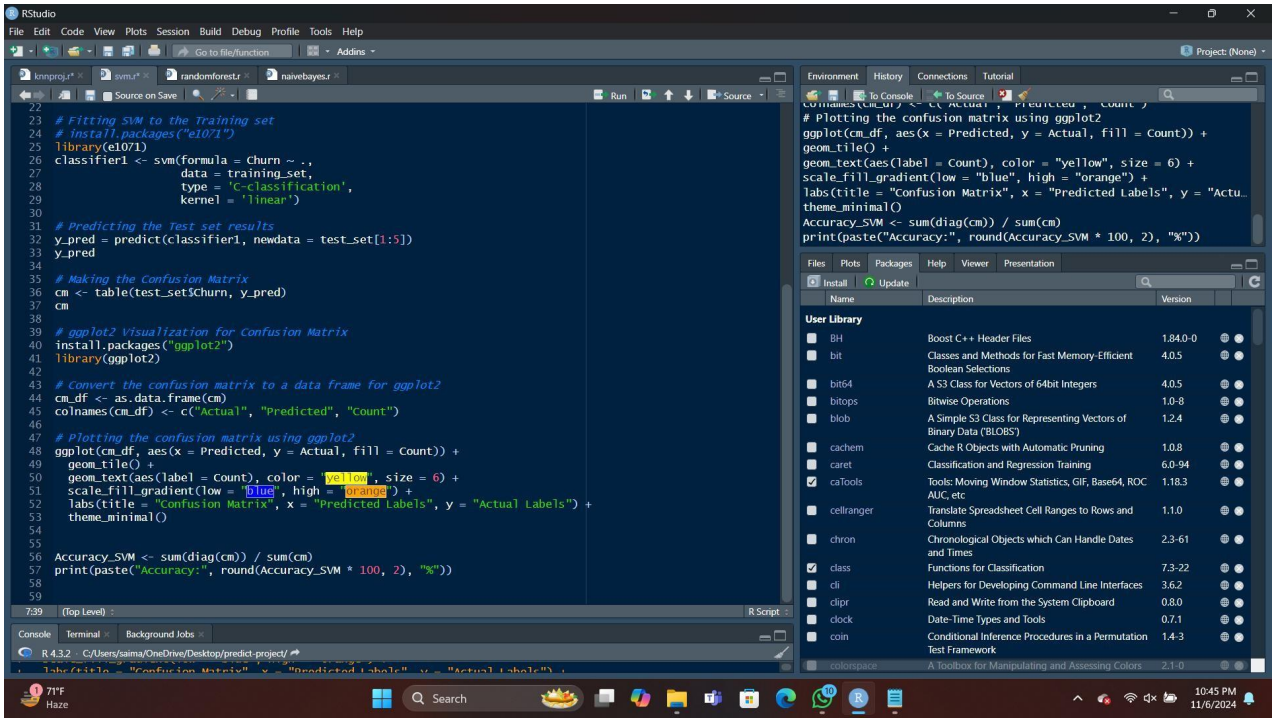
Customer churn is a critical issue impacting revenue and customer lifetime value in subscription-based businesses. This project leverages machine learning to predict customer churn, enabling targeted retention actions for at-risk customers, with a focus on demographics, subscription details, and satisfaction levels. Early detection of churn-prone customers allows companies to prioritize retention efforts, aiming for enhanced customer loyalty, reduced churn rates, and sustainable growth

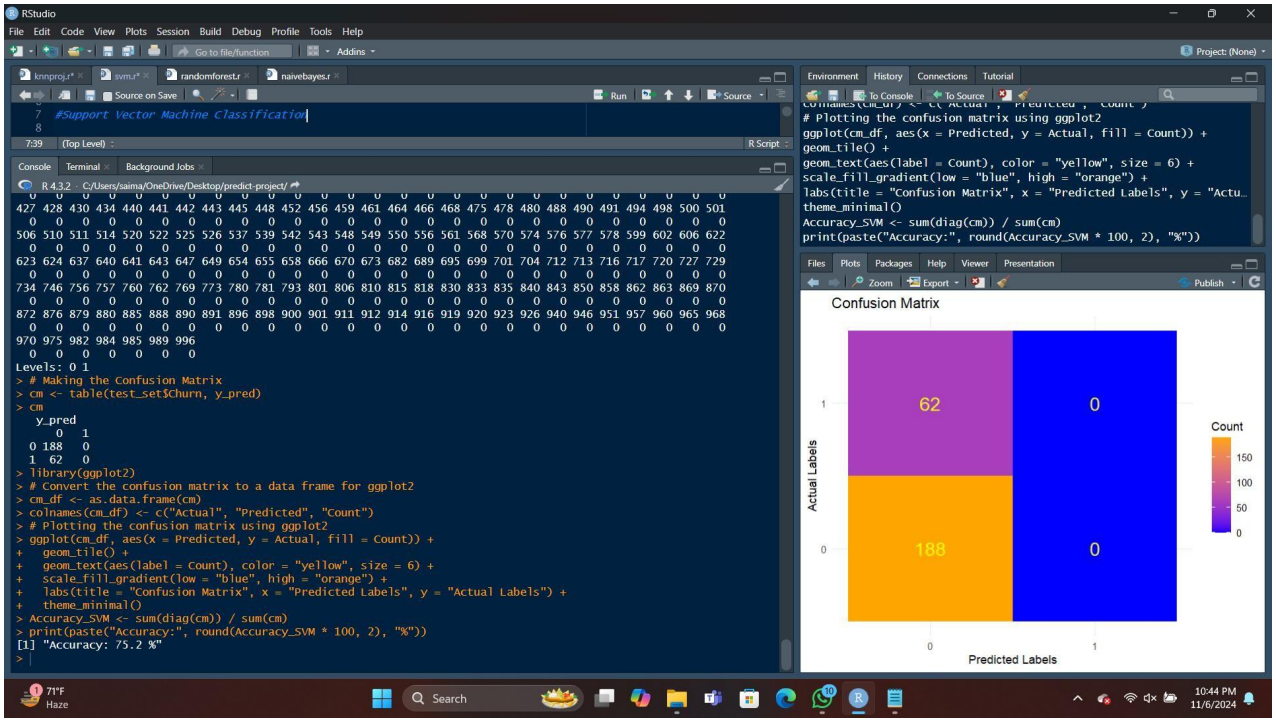
1.K-Nearest Neighbor:



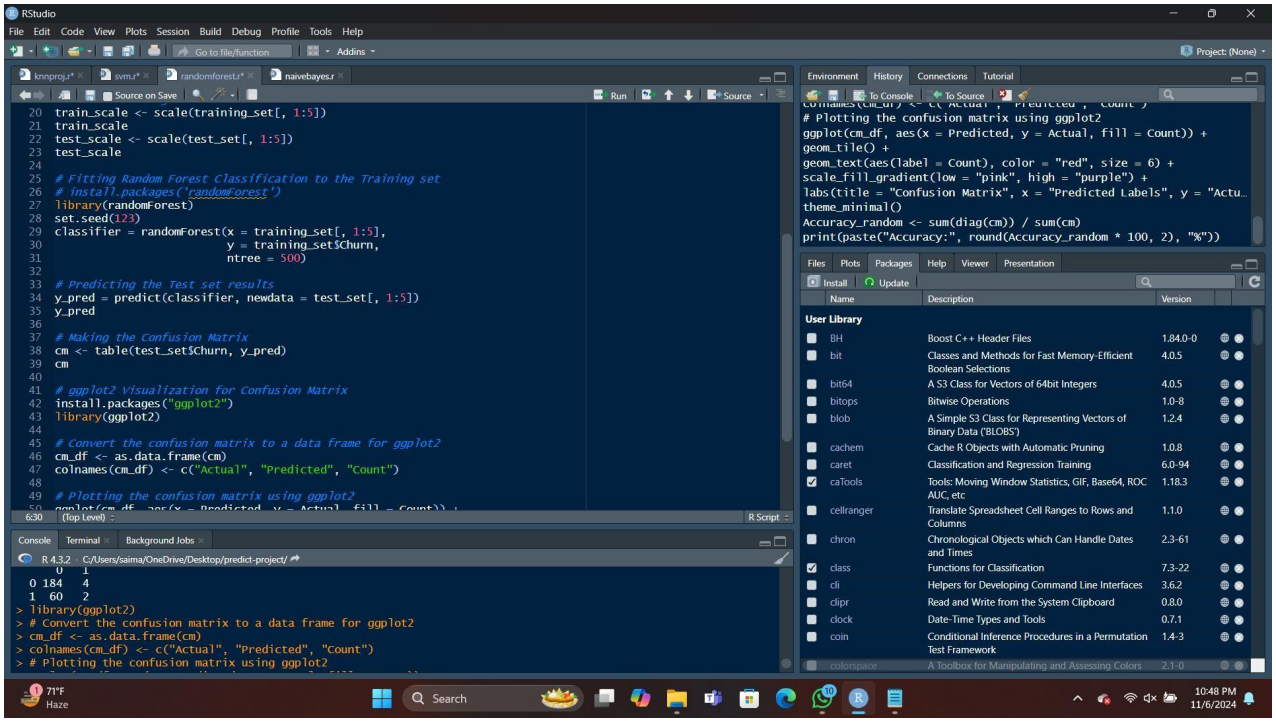


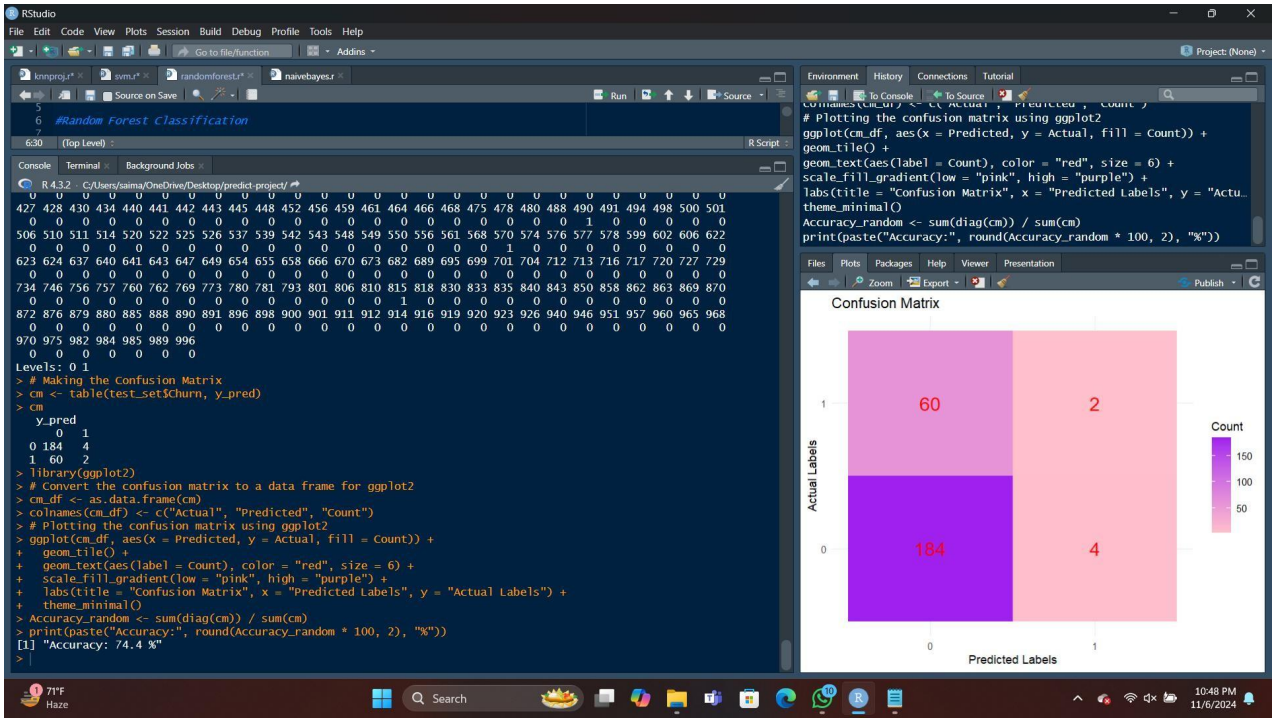
2.Support Vector Machine(SVM):



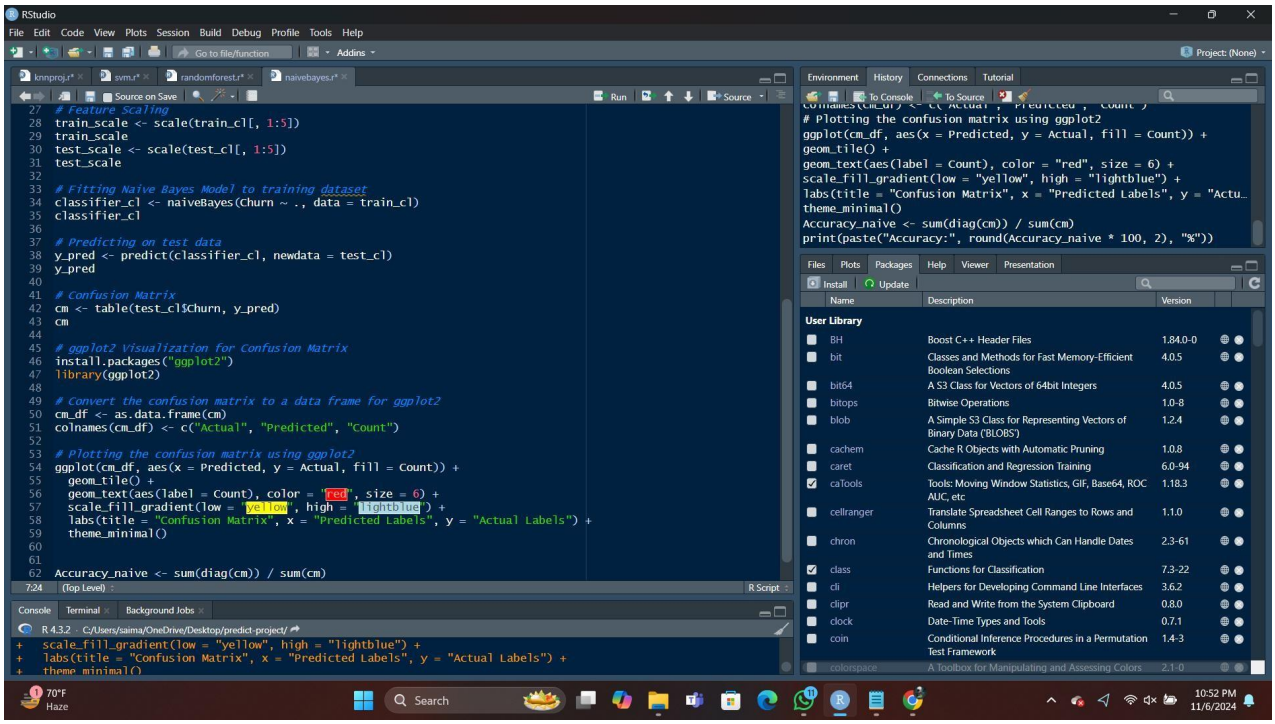


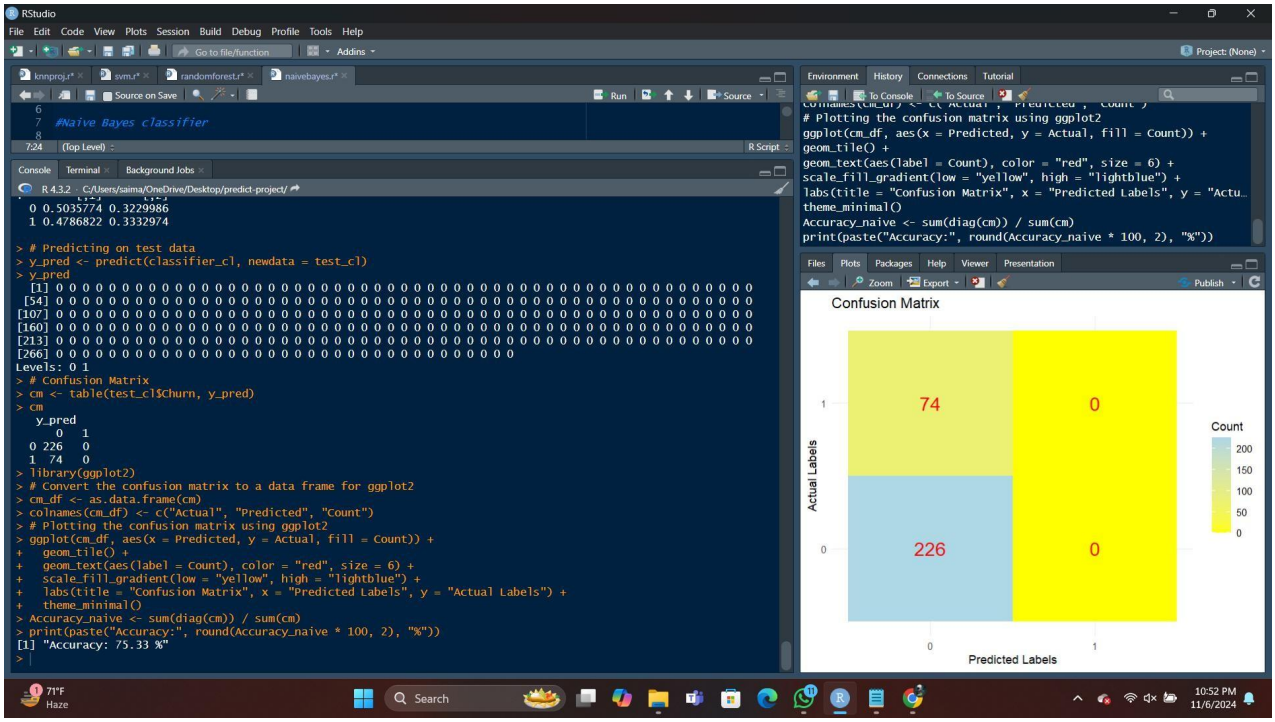
3. Random Forest:





4.Naive Bayes





Conclusion:

The K-Nearest Neighbors model yielded the highest accuracy, closely followed by Naive Bayes, indicating that the dataset benefits from models that capture both local structure (KNN) and probabilistic structure (Naive Bayes). However, all models produced similar accuracies, suggesting that additional feature engineering or ensemble methods could further improve performance