Maternal Health Risk Detection: A Comparative Analysis of Classification Models

Nikhil Keshri

2024-10-28

Maternal health refers to the health of women during pregnancy, childbirth, and the postpartum period. It encompasses the physical, mental, and social well-being of women as they go through these life stages, and it is critical to ensuring the health and well-being of both mothers and their newborns.

Key components of maternal health include:

- Prenatal Care: Medical and nutritional care provided to women during pregnancy. This includes regular check-ups, screenings for health conditions, and guidance on diet and lifestyle, which help ensure both mother and baby stay healthy.
- Safe Childbirth: Access to skilled healthcare providers, such as midwives and obstetricians, during delivery. Safe childbirth practices reduce risks associated with complications like infections, hemorrhaging, and obstructed labor.
- Postpartum Care: Health care and support provided to women after delivery to help them recover physically and emotionally. This period includes monitoring for postpartum depression, infections, and other conditions that can arise after childbirth.
- Family Planning: Access to information and services that allow women to decide if and when they want to have children. Family planning is essential for maternal health because it allows women to space births in a way that minimizes health risks.
- Education and Support Services: Maternal health also includes education and support on topics like breastfeeding, mental health, nutrition, and recognizing signs of complications.

Maternal health is a major public health priority worldwide, as complications related to pregnancy and childbirth remain leading causes of mortality and morbidity among women of reproductive age, particularly in low- and middle-income countries. Improving maternal health can significantly reduce infant mortality, improve community health, and empower women through better health outcomes.

Maternal Health Risk Detection

Maternal health risk prediction is a vital tool for healthcare professionals to assess and mitigate potential risks for pregnant individuals. This analysis evaluates various machine

learning models to predict risk levels based on health indicators, ultimately helping prioritize timely interventions for those at higher risk.

- This project explores multiple classification models to identify the best-performing model, examining each in detail with visual comparisons and accuracy metrics.

1. Libraries and Data Loading

First, we import the necessary libraries and load the dataset. These libraries cover data processing, model training, and visualization.

Importing Important Libraies

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.4.1
```

```
library(class)
```

```
## Warning: package 'class' was built under R version 4.4.1
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.4.1
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.4.1
```

```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.4.1
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.1
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

Loading Dataset of Maternal Health Risk Detection to perform predictive analysis and see the result of different classifications models to find that which model is giving high accuracy.

```
dataset = read.csv('Maternal Health risk Data Set.csv')
str(dataset)
```

```
## 'data.frame':    1014 obs. of  7 variables:
##  $ Age        : int  25 35 29 30 35 23 23 35 32 42 ...
##  $ SystolicBP : int  130 140 90 140 120 140 130 85 120 130 ...
##  $ DiastolicBP: int  80 90 70 85 60 80 70 60 90 80 ...
##  $ BS         : num  15 13 8 7 6.1 7.01 7.01 11 6.9 18 ...
##  $ BodyTemp   : num  98 98 100 98 98 98 98 102 98 98 ...
##  $ HeartRate  : int  86 70 80 70 76 70 78 86 70 70 ...
##  $ RiskLevel  : chr  "high risk" "high risk" "high risk" "high risk" ...
```

Checking Summary of dataset

```
summary(dataset)
```

```
##       Age          SystolicBP     DiastolicBP          BS
##  Min.   :10.00   Min.   : 70.0   Min.   : 49.00   Min.   : 6.000
##  1st Qu.:19.00   1st Qu.:100.0   1st Qu.: 65.00   1st Qu.: 6.900
##  Median :26.00   Median :120.0   Median : 80.00   Median : 7.500
##  Mean   :29.87   Mean   :113.2   Mean   : 76.46   Mean   : 8.726
##  3rd Qu.:39.00   3rd Qu.:120.0   3rd Qu.: 90.00   3rd Qu.: 8.000
##  Max.   :70.00   Max.   :160.0   Max.   :100.00   Max.   :19.000
##     BodyTemp        HeartRate     RiskLevel
##  Min.   : 98.00   Min.   : 7.0   Length:1014
##  1st Qu.: 98.00   1st Qu.:70.0   Class :character
##  Median : 98.00   Median :76.0   Mode  :character
##  Mean   : 98.67   Mean   :74.3
##  3rd Qu.: 98.00   3rd Qu.:80.0
##  Max.   :103.00   Max.   :90.0
```

In dataset we see that column name “RiskLevel” is given in categorical non numerical way, so we are converting it into numerical form for better analysis.

The target variable, RiskLevel, is currently categorical. We convert it to a numerical factor, which facilitates better compatibility with most machine learning algorithms.

```
dataset$RiskLevel = factor(dataset$RiskLevel, levels = c('low risk','mid
risk','high risk'), labels = c(1,2,3))

summary(dataset)
```

```
##       Age          SystolicBP     DiastolicBP          BS
##  Min.   :10.00   Min.   : 70.0   Min.   : 49.00   Min.   : 6.000
##  1st Qu.:19.00   1st Qu.:100.0   1st Qu.: 65.00   1st Qu.: 6.900
##  Median :26.00   Median :120.0   Median : 80.00   Median : 7.500
##  Mean   :29.87   Mean   :113.2   Mean   : 76.46   Mean   : 8.726
##  3rd Qu.:39.00   3rd Qu.:120.0   3rd Qu.: 90.00   3rd Qu.: 8.000
##  Max.   :70.00   Max.   :160.0   Max.   :100.00   Max.   :19.000
##     BodyTemp        HeartRate     RiskLevel
##  Min.   : 98.00   Min.   : 7.0   1:406
##  1st Qu.: 98.00   1st Qu.:70.0   2:336
##  Median : 98.00   Median :76.0   3:272
##  Mean   : 98.67   Mean   :74.3
##  3rd Qu.: 98.00   3rd Qu.:80.0
##  Max.   :103.00   Max.   :90.0
```

We split the data into training (80%) and testing (20%) subsets to evaluate model performance effectively.

```
#library(caTools)
split = sample.split(dataset$RiskLevel, SplitRatio = 0.8)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
```

```
train_label <- subset(dataset$RiskLevel, split == TRUE)
test_label <- subset(dataset$RiskLevel, split == FALSE)
```

Model Training and Evaluation

We apply various classification models, analyze their confusion matrices, and calculate accuracy metrics.

K-Nearest Neighbors (KNN)

KNN classifies data points based on the proximity to labeled points. We evaluate its accuracy below.

```
#library(class)
knn_class <- knn(train = training_set, test = test_set, cl = train_label, k = 5)
cm_knn <- table(test_label, knn_class)
cm_knn
```

```
##           knn_class
## test_label  1  2  3
##          1 62 18  1
##          2 20 45  2
##          3  2 12 40
```

Analysis of Accuracy of KNN model.

```
acc_knn <- sum(diag(cm_knn)) / sum(cm_knn)
print(paste("Accuracy KNN: ", round(acc_knn * 100, 2), "%"))
```

```
## [1] "Accuracy KNN:  72.77 %"
```

Naive Bayes Model

The Naive Bayes model, based on Bayes' theorem, is particularly suited to categorical data.

```
#library(e1071)
model_naive <- naiveBayes(RiskLevel ~ ., data = training_set)
predict_naive <- predict(model_naive, newdata = test_set)
cm_naive <- table(test_label, predict_naive)
cm_naive
```

```
##           predict_naive
## test_label  1  2  3
##          1 68  9  4
##          2 40 23  4
##          3  8 15 31
```

Analysis of Accuracy of Naive Bayes Model.

```
acc_naive <- sum(diag(cm_naive)) / sum(cm_naive)
print(paste("Accuracy Naive Bayes: ", round(acc_naive * 100, 2), "%"))
```

```
## [1] "Accuracy Naive Bayes:  60.4 %"
```

Support Vector Machine Model

```
#library(e1071)
svm_model <- svm(RiskLevel ~ ., data = training_set, kernel = "linear")
predict_svm <- predict(svm_model, test_set)
cm_svm <- table(test_label, predict_svm)
cm_svm
```

```
##           predict_svm
## test_label  1  2  3
##          1 66 12  3
##          2 29 30  8
##          3  1 15 38
```

Analysis of Accuracy of SVM

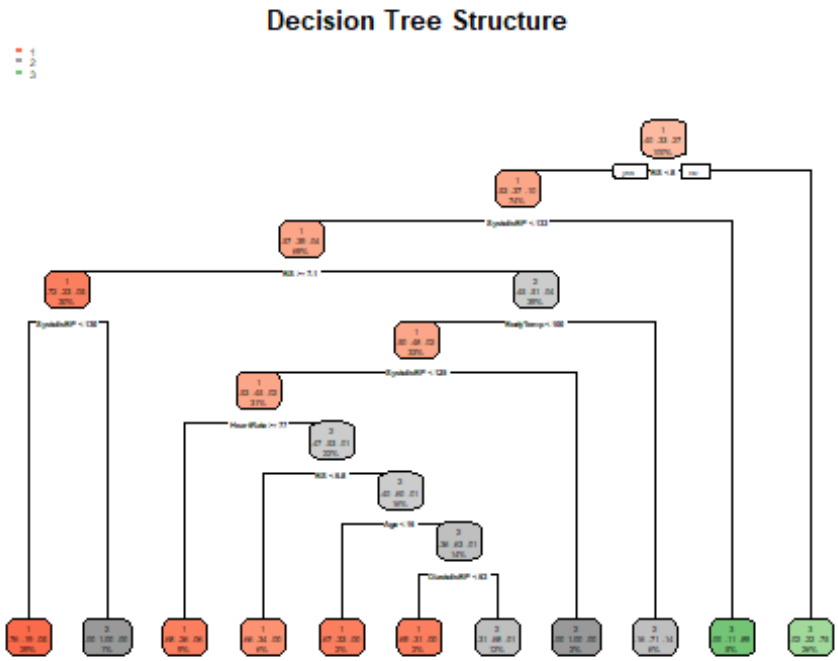
```
acc_svm <- sum(diag(cm_svm)) / sum(cm_svm)
print(paste("Accuracy SVM: ", round(acc_svm * 100, 2), "%"))
```

```
## [1] "Accuracy SVM:  66.34 %"
```

Decision Tree Model

Decision Trees split data by selecting features that provide the most significant information gain at each step.

```
#library(rpart)
#library(rpart.plot)
dt_model <- rpart(RiskLevel ~ ., data = training_set, method = "class")
rpart.plot(dt_model, main = "Decision Tree Structure")
```



Prediction of Decision tree

```
predict_dt <- predict(dt_model, test_set, type = "class")
cm_dt <- table(test_label, predict_dt)
cm_dt
```

```
##           predict_dt
## test_label  1  2  3
##          1 65 13  3
##          2 31 25 11
##          3  6  4 44
```

Accuracy of Decision Tree

```
acc_dt <- sum(diag(cm_dt)) / sum(cm_dt)
print(paste("Accuracy Decision Tree: ", round(acc_dt * 100, 2), "%"))
```

```
## [1] "Accuracy Decision Tree:  66.34 %"
```

Random Forest Model

Random Forest is an ensemble method that builds multiple decision trees and averages them to make a final prediction.

```
#library(randomForest)
rf_model <- randomForest(RiskLevel ~ ., data = training_set)
predict_rf <- predict(rf_model, test_set)
```

```r
cm_rf <- table(test_label, predict_rf)
cm_rf
```

```
##           predict_rf
## test_label  1  2  3
##          1 63 15  3
##          2 10 55  2
##          3  2  5 47
```

Accuracy of Random Forest

```r
acc_rf <- sum(diag(cm_rf)) / sum(cm_rf)
print(paste("Accuracy Random Forest: ", round(acc_rf * 100, 2), "%"))
```

```
## [1] "Accuracy Random Forest:  81.68 %"
```

Ploting Function to Plot Confusion Matrix

```r
plot_cm <- function(cm, title) {
  cm_df <- as.data.frame(cm)  # Convert table to data frame
  names(cm_df) <- c("Actual", "Predicted", "Freq")  # Rename columns for ggplot compatibility

  ggplot(cm_df, aes(x = Predicted, y = Actual)) +
    geom_tile(aes(fill = Freq), color = "white") +
    scale_fill_gradient(low = "green", high = "red") +
    geom_text(aes(label = Freq)) +
    labs(title = title, x = "Predicted", y = "Actual", fill = "Count") +
    theme_minimal()
}
```

KNN

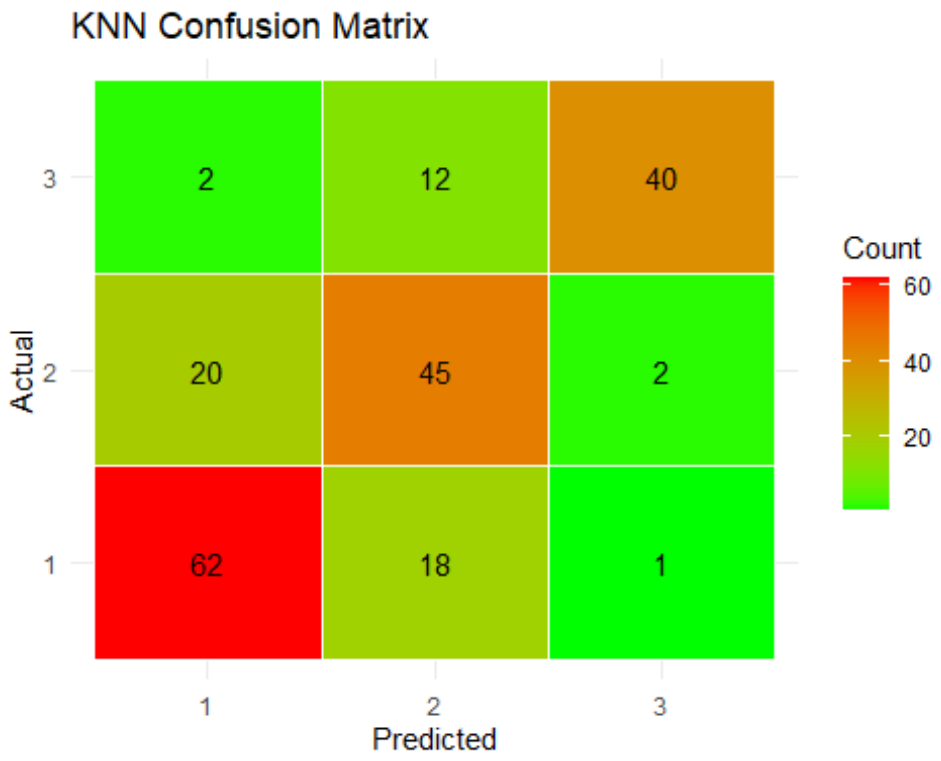
```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

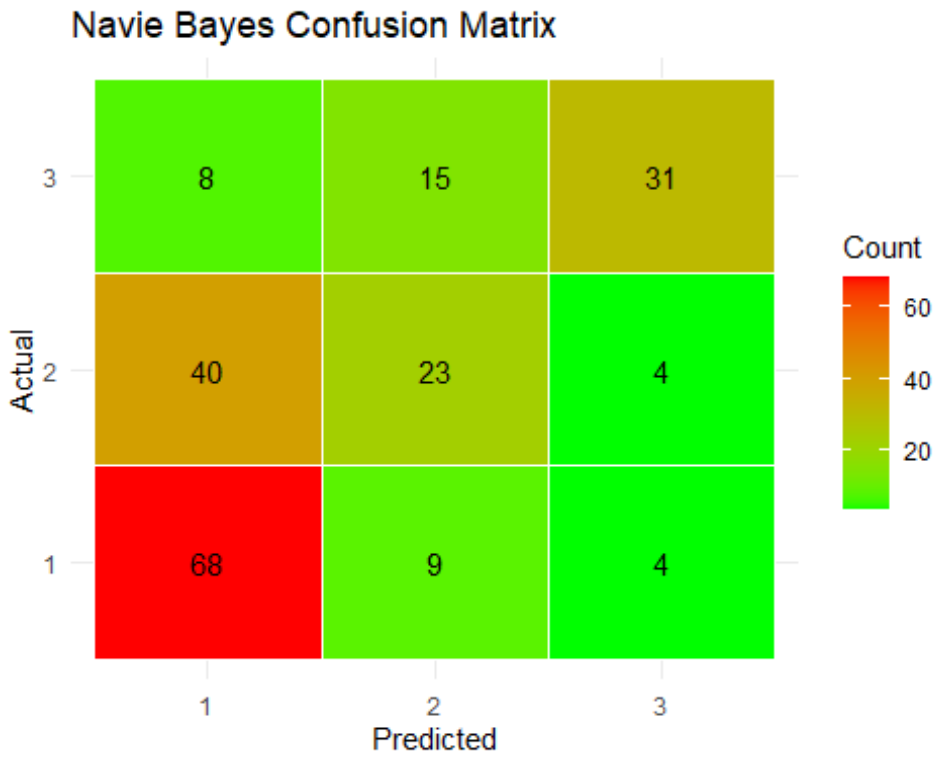
```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```r
plot_cm(cm_knn, "KNN Confusion Matrix")
```

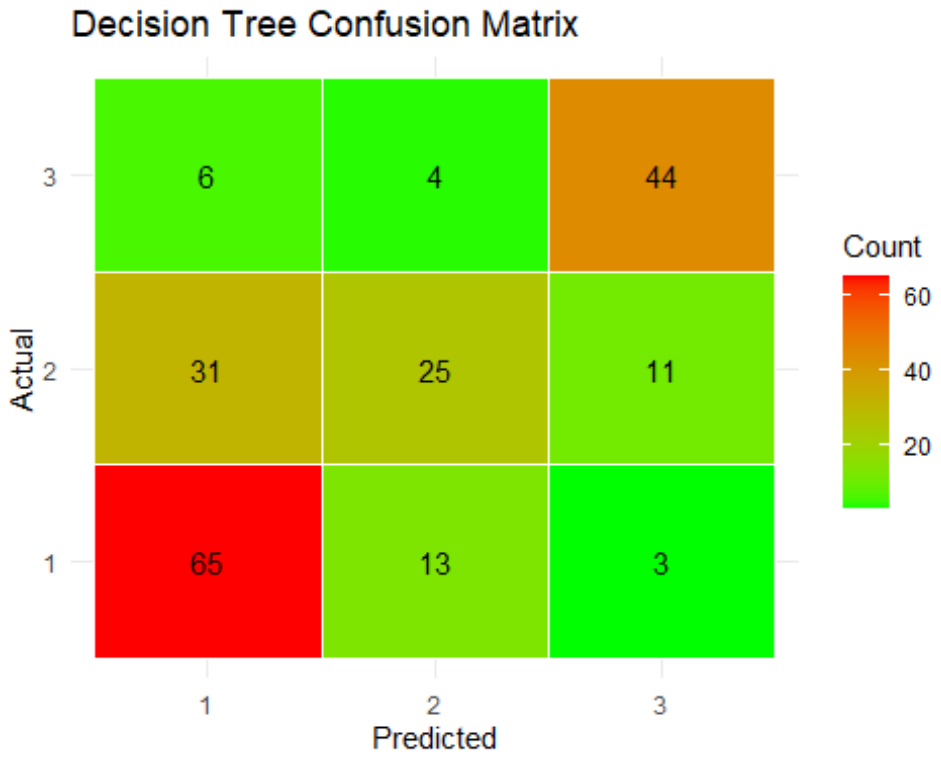



```
plot_cm(cm_naive,"Navie Bayes Confusion Matrix")
```



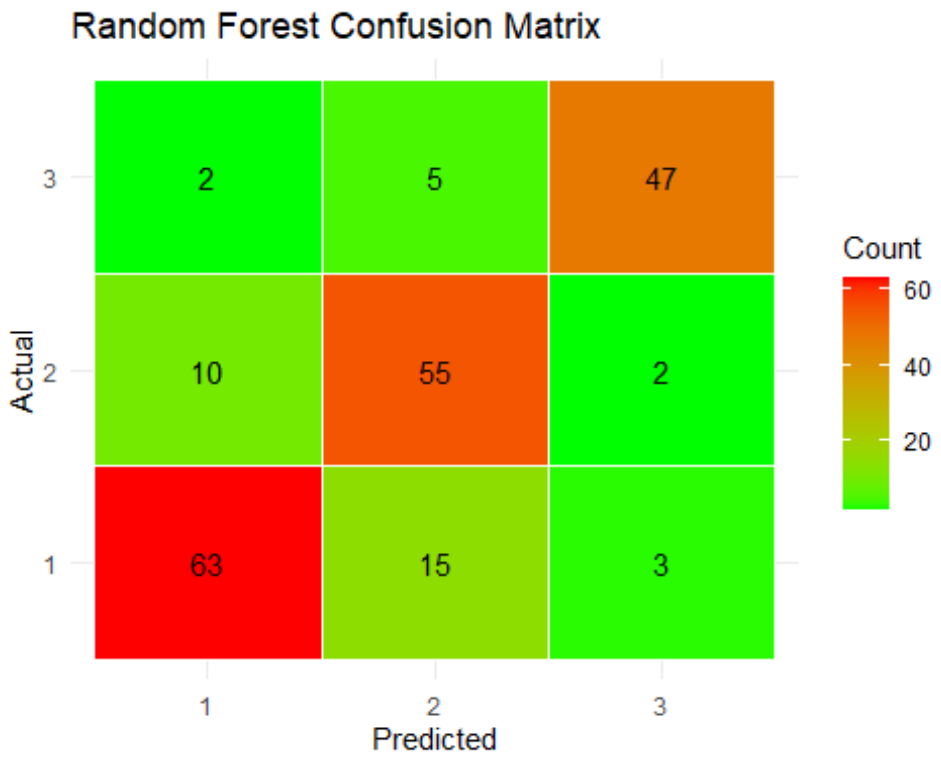
Decision tree

```
plot_cm(cm_dt,"Decision Tree Confusion Matrix")
```



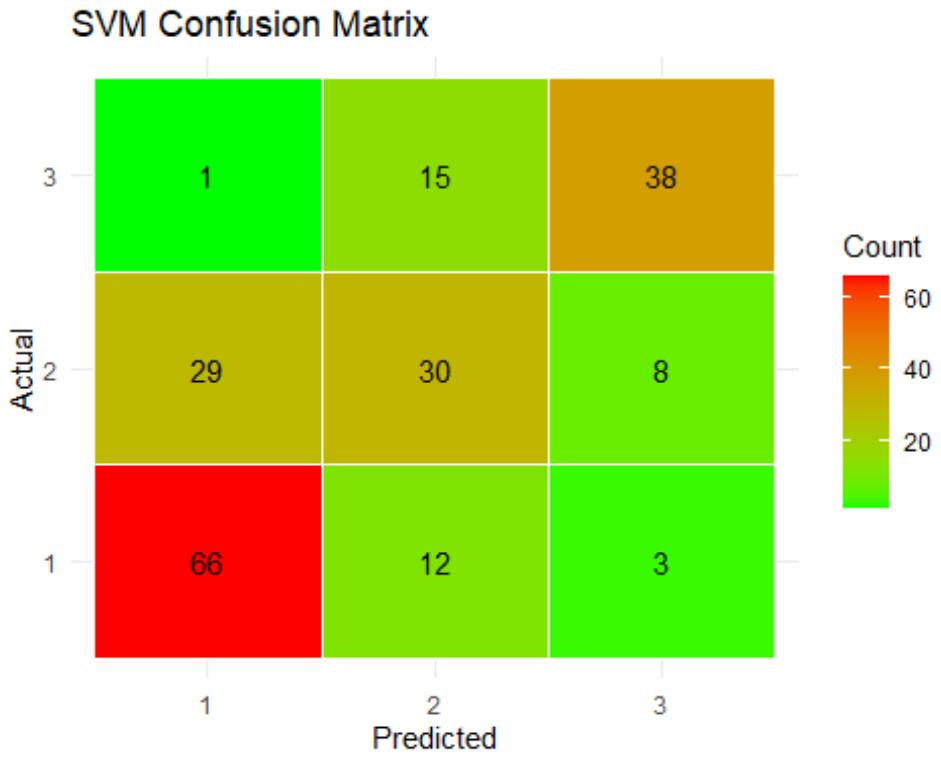
Random Forest

```
plot_cm(cm_rf,"Random Forest Confusion Matrix")
```



SVM

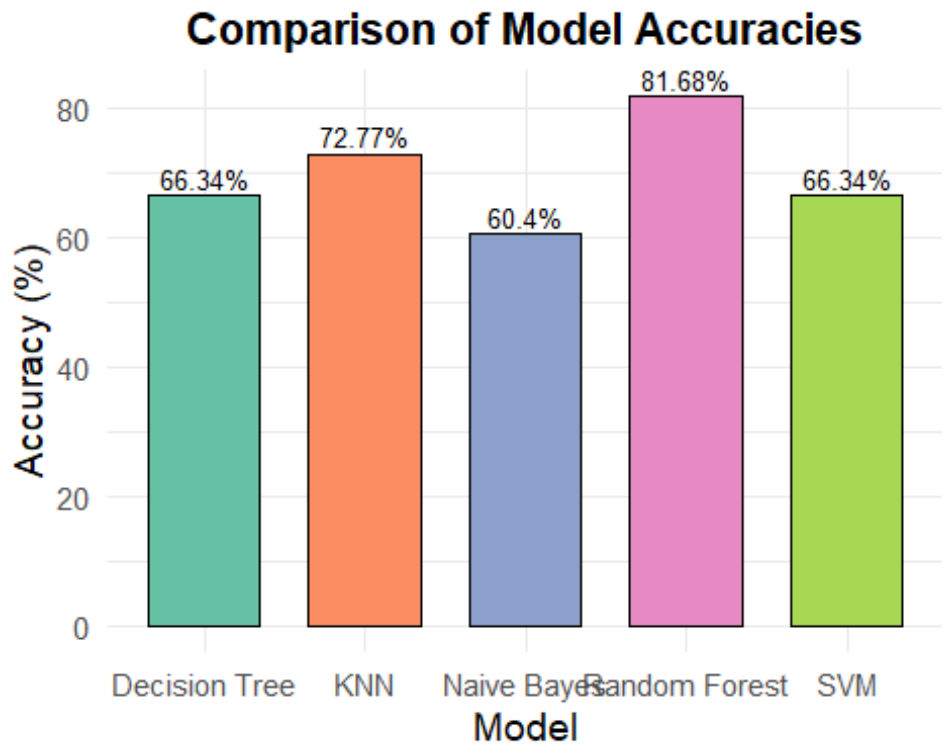
```
plot_cm(cm_svm,"SVM Confusion Matrix")
```



Accuracy of All Models

```r
accuracy <- data.frame(
  Model = c("KNN", "Naive Bayes", "SVM", "Decision Tree", "Random Forest"),
  Accuracy = c(acc_knn, acc_naive, acc_svm, acc_dt, acc_rf)
)

ggplot(accuracy, aes(x = Model, y = Accuracy * 100, fill = Model)) +
  geom_bar(stat = "identity", color = "black", width = 0.7) +
  geom_text(aes(label = paste0(round(Accuracy * 100, 2), "%")), vjust = -0.3,
size = 3.5) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Comparison of Model Accuracies",
    y = "Accuracy (%)",
    x = "Model"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "none"
  )
```



Conclusions

Summary of Findings Model Performance: Random Forest achieved the highest accuracy among the models tested, indicating its strength in handling complex, non-linear data interactions within the dataset.

Feature Importance: The feature importance plot reveals the health indicators most predictive of risk level, guiding healthcare professionals on which factors to prioritize for early intervention.

Impact on Healthcare: Predictive analytics in maternal health can play a transformative role, allowing practitioners to identify high-risk cases proactively and allocate resources effectively.

Future Recommendations Further improvements can be explored by incorporating additional data, applying alternative models, or conducting feature engineering to enhance predictive accuracy and generalizability in other healthcare datasets.