Predicting Liver Disease in Indian Patients: A Comparative Machine Learning Study

Sibashish

2025-10-04

Contents

1. Introduction

Liver disease is a major health concern in India. The Indian Liver Patient Records dataset contains clinical and demographic information for 583 patients, including features such as age, gender, serum bilirubin levels, and more. The target variable indicates whether a patient has liver disease (1) or is healthy (2). My objective is to build predictive models that can classify patients as having liver disease or not, based on these features.

2. Executive Summary

This report attempts to analyze the Indian Liver Patient Records dataset to predict liver disease using supervised machine learning algorithms. It is through comparison of Logistic Regression, Random Forest, and Support Vector Machine (SVM) classifiers. The workflow covers data cleaning, exploration, modeling, and evaluation, with visualizations and a summary table of model performance.

3. Methods and Analysis

3.1 Data Loading and Cleaning

The dataset is automatically downloaded from GitHub since/if it is not already present locally. Missing values in numeric columns are replaced with the median value, while missing values in categorical columns are replaced with the most common value (mode). This approach helps retain as much data as possible and avoids bias that could result from simply removing records with missing entries.

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE, fig.width=7, fig.height=4)
packages <- c("tidyverse", "caret", "randomForest", "e1071", "ROSE", "pROC")
installed <- packages %in% rownames(installed.packages())
if (any(!installed)) install.packages(packages[!installed])
invisible(lapply(packages, library, character.only = TRUE))
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: lattice
##
##
## Attaching package: 'caret'
##
##
## The following object is masked from 'package:purrr':
##
##     lift
##
##
## randomForest 4.7-1.2
##
## Type rfNews() to see new features/changes/bug fixes.
##
##
## Attaching package: 'randomForest'
##
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
##
## The following object is masked from 'package:ggplot2':
##
##     margin
##
```

```
##
##
## Attaching package: 'e1071'
##
##
## The following object is masked from 'package:ggplot2':
##
##     element
##
##
## Loaded ROSE 0.0-4
##
##
## Type 'citation("pROC")' for a citation.
##
##
## Attaching package: 'pROC'
##
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
# Mode function for categorical imputation
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

# Download data if not present
url <- "https://raw.githubusercontent.com/bizysiby/DS_India_liver_patients/main/indian_liver_patient.csv"
destfile <- "indian_liver_patient.csv"
if (!file.exists(destfile)) {
  download.file(url, destfile, method = "libcurl")
}
data <- read.csv(destfile)

# Input missing values
for (col in names(data)) {
  if (any(is.na(data[[col]]))) {
    if (is.numeric(data[[col]])) {
      data[[col]][is.na(data[[col]])] <- median(data[[col]], na.rm = TRUE)
    } else {
      data[[col]][is.na(data[[col]])] <- Mode(data[[col]])
    }
  }
}
data$Gender <- as.factor(data$Gender)
data$Dataset <- factor(ifelse(data$Dataset == 1, "LiverPatient", "Healthy"))
```

3.2 Data Exploration and Visualization

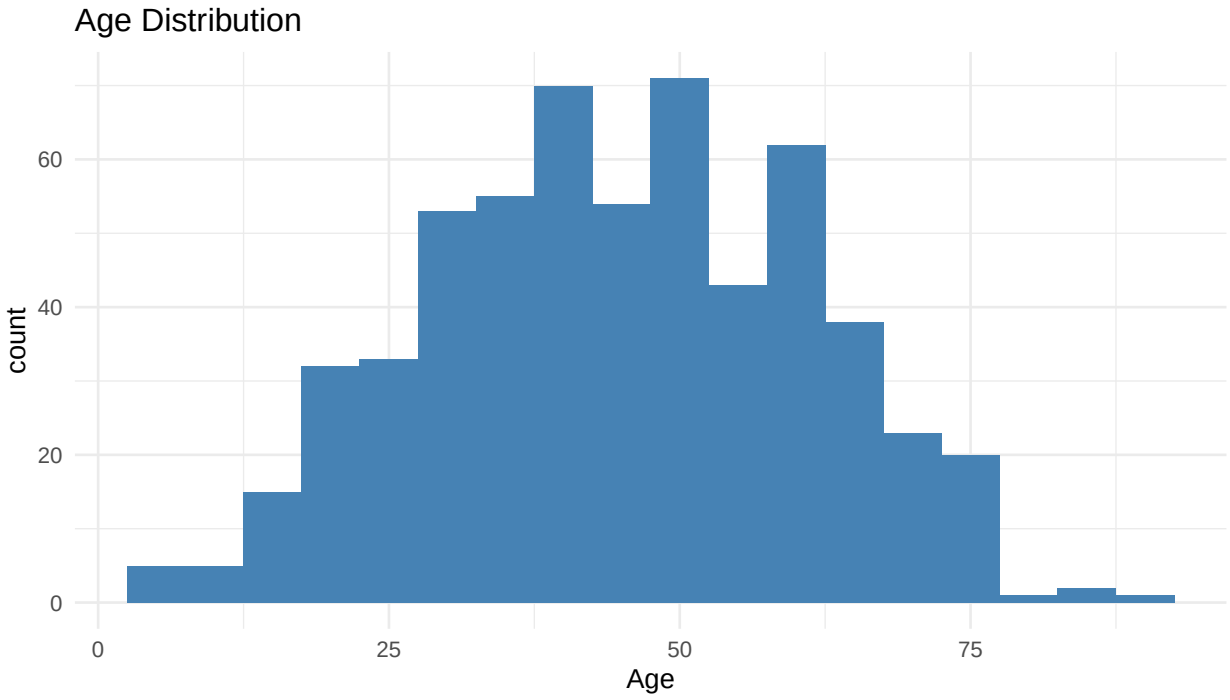
Summary statistics and visualizations help understand the data distribution and relationships.

```
##       Age           Gender    Total_Bilirubin  Direct_Bilirubin
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200
##  Median :45.00                Median : 1.000   Median : 0.300
##  Mean   :44.75                Mean   : 3.299   Mean   : 1.486
##  3rd Qu.:58.00                3rd Qu.: 2.600   3rd Qu.: 1.300
##  Max.   :90.00                Max.   :75.000   Max.   :19.700
##  Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase
##  Min.   :  63.0       Min.   :  10.00          Min.   :  10.0
##  1st Qu.: 175.5       1st Qu.:  23.00          1st Qu.:  25.0
##  Median : 208.0       Median :  35.00          Median :  42.0
##  Mean   : 290.6       Mean   :  80.71          Mean   : 109.9
##  3rd Qu.: 298.0       3rd Qu.:  60.50          3rd Qu.:  87.0
##  Max.   :2110.0       Max.   :2000.00          Max.   :4929.0
##  Total_Protiens     Albumin      Albumin_and_Globulin_Ratio         Dataset
##  Min.   :2.700   Min.   :0.900   Min.   :0.3000             Healthy     :167
##  1st Qu.:5.800   1st Qu.:2.600   1st Qu.:0.7000             LiverPatient:416
##  Median :6.600   Median :3.100   Median :0.9300
##  Mean   :6.483   Mean   :3.142   Mean   :0.9469
##  3rd Qu.:7.200   3rd Qu.:3.800   3rd Qu.:1.1000
##  Max.   :9.600   Max.   :5.500   Max.   :2.8000
```

4. Results

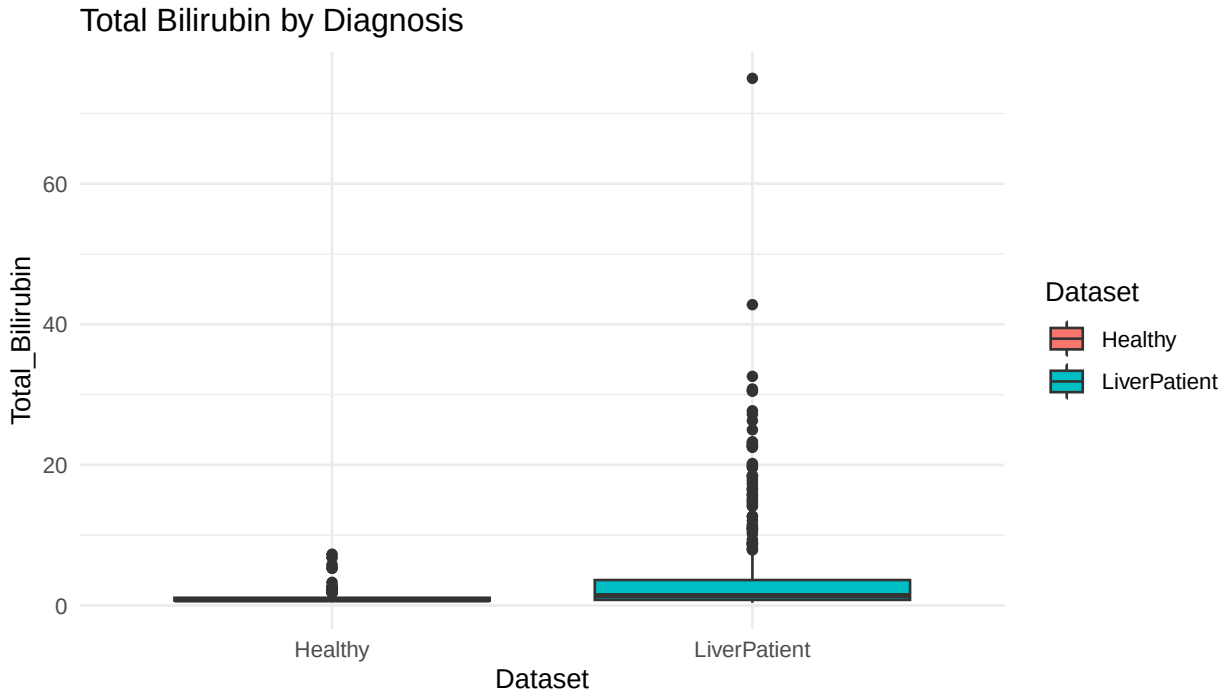
4.1 Age distribution

This plot shows the distribution of patient ages in the dataset. It helps us understand the age range and any patterns that may exist, such as whether liver disease is more common in certain age groups.



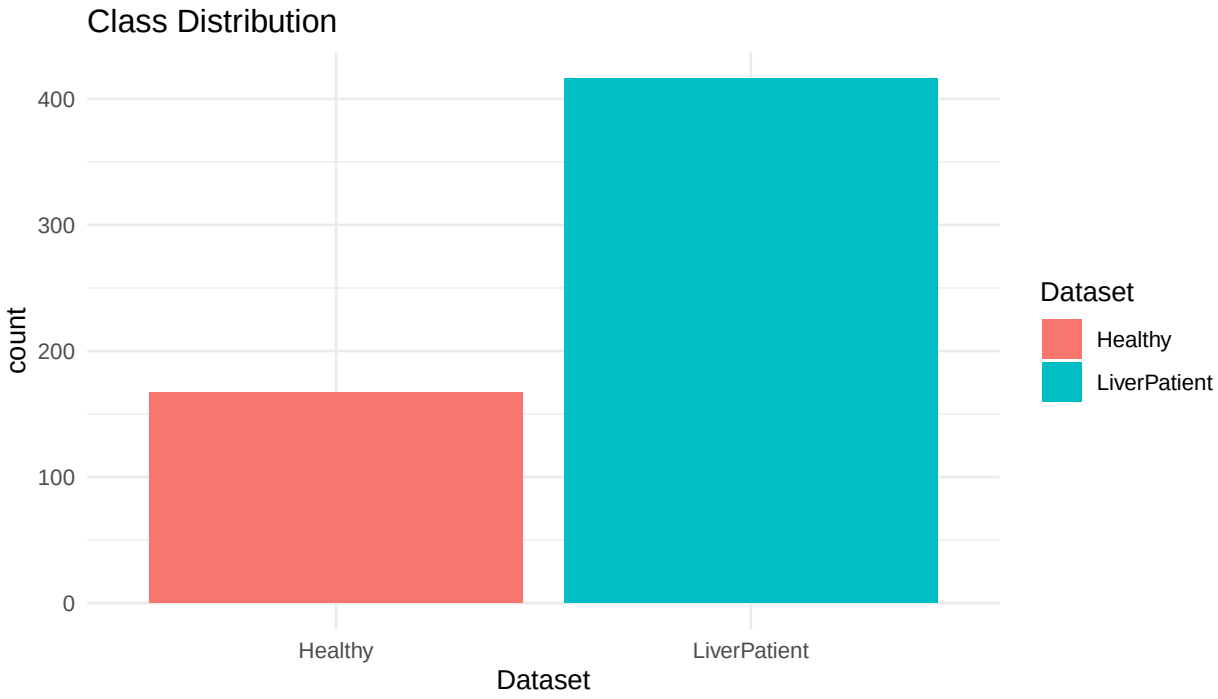
4.2 Total Bilirubin by Diagnosis

This boxplot compares total bilirubin levels between patients diagnosed with liver disease and healthy individuals.
Total bilirubin is an important indicator of liver function, and differences between groups may highlight key risk factors.



4.3 Class Distribution

This bar chart illustrates the number of patients in each diagnostic category (LiverPatient vs. Healthy). It highlights any imbalance in the dataset, which is important for modeling and evaluation.



5. Insight

There is a class imbalance, with more patients diagnosed with liver disease than healthy individuals. To avoid bias in model training, ROSE package was used to balance the dataset.

```
data_balanced <- ROSE(Dataset ~ ., data = data, seed = 1)$data
table(data_balanced$Dataset)
```

```
##
## LiverPatient      Healthy
##          306          277
```

6. Train/Test Split

Use an 80/20 train/test split. This choice balances the need for sufficient training data with robust testing. A 50/50 split would reduce model accuracy, while 90/10 could lead to unreliable test results.

```
set.seed(123)
trainIndex <- createDataPartition(data_balanced$Dataset, p = 0.8, list = FALSE)
train <- data_balanced[trainIndex, ]
test <- data_balanced[-trainIndex, ]
```

```
str(train)
```

```
## 'data.frame':    467 obs. of  11 variables:
##  $ Age                       : num  41.6 59.5 34.3 31.6 73.1 ...
##  $ Gender                    : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 1 2 1 2 ...
##  $ Total_Bilirubin           : num  0.394 2.725 0.731 25.494 1.059 ...
##  $ Direct_Bilirubin          : num  1.113 -0.738 2.949 11.014 0.804 ...
##  $ Alkaline_Phosphotase      : num  34.1 94 132.9 177 182.5 ...
##  $ Alamine_Aminotransferase  : num  80.1 -258.9 340.6 79.5 107.6 ...
##  $ Aspartate_Aminotransferase: num  -239.5 24.5 203 685.7 -112.8 ...
##  $ Total_Protiens            : num  6.48 4.57 4.27 9.13 7.09 ...
##  $ Albumin                   : num  3.44 2.85 2.62 1.58 4.29 ...
##  $ Albumin_and_Globulin_Ratio: num  1.33 1.45 0.824 0.344 1.156 ...
##  $ Dataset                   : Factor w/ 2 levels "LiverPatient",..: 1 1 1 1 1 1 1 1 1 1 ...
```

7. Feature Scaling for SVM

SVMs require scaled features for optimal performance.

```
scaling_vars <- setdiff(names(train), c("Gender", "Dataset"))
preProc <- preProcess(train[, scaling_vars], method = c("center", "scale"))
train_scaled <- train
test_scaled <- test
train_scaled[, scaling_vars] <- predict(preProc, train[, scaling_vars])
test_scaled[, scaling_vars] <- predict(preProc, test[, scaling_vars])
```

8. Modeling Approach

Approach is to compare three models:

- **Logistic Regression:** - A baseline linear model.
- **Random Forest:** - A tree-based ensemble model, robust to nonlinearity and interactions.
- **Support Vector Machine (SVM):** - A more advanced model, effective for complex decision boundaries.

All models use 5-fold cross-validation for robust performance estimation.

8.1 Logistic Regression

Logistic Regression is used as a baseline linear model for classification.
It estimates the probability of a patient having liver disease based on clinical features.
The output includes a confusion matrix and the area under the ROC curve (AUC), which help assess the model's accuracy and ability to distinguish between classes.

```
set.seed(123)
log_model <- train(Dataset ~ ., data = train, method = "glm", family = "binomial", trControl = trainCon
log_pred <- predict(log_model, test)
log_prob <- predict(log_model, test, type = "prob")[,2]
log_cm <- confusionMatrix(log_pred, test$Dataset)
log_roc <- roc(response = test$Dataset, predictor = log_prob, levels = rev(levels(test$Dataset)))
log_cm
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     LiverPatient Healthy
##   LiverPatient           34      10
##   Healthy                27      45
##
##                Accuracy : 0.681
##                  95% CI : (0.5881, 0.7645)
##     No Information Rate : 0.5259
##     P-Value [Acc > NIR] : 0.0004945
##
##                   Kappa : 0.3699
##
##  Mcnemar's Test P-Value : 0.0085289
##
##             Sensitivity : 0.5574
##             Specificity : 0.8182
##          Pos Pred Value : 0.7727
##          Neg Pred Value : 0.6250
##              Prevalence : 0.5259
##          Detection Rate : 0.2931
##    Detection Prevalence : 0.3793
##       Balanced Accuracy : 0.6878
##
##        'Positive' Class : LiverPatient
##
```

```
auc(log_roc)
```

```
## Area under the curve: 0.6861
```

8.2 Random Forest

Random Forest is a tree-based ensemble method that can capture nonlinear relationships and interactions between variables. It often performs well on complex datasets.
The output includes model performance metrics and a plot showing the importance of each feature in predicting liver disease.

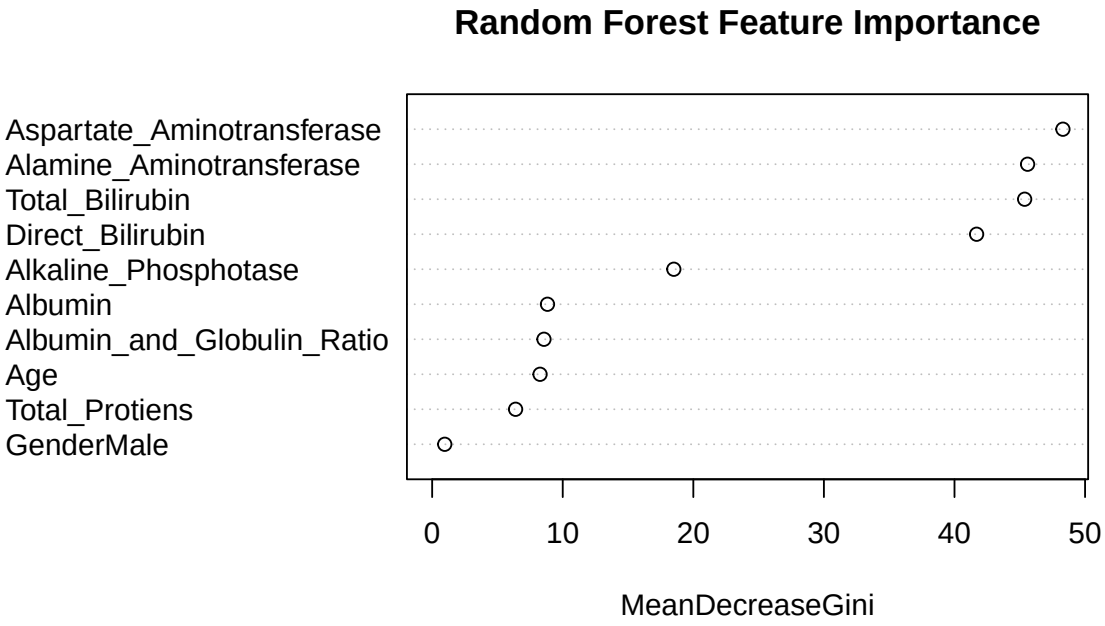
```
set.seed(123)
rf_grid <- expand.grid(mtry = c(2, 4, 6))
rf_model <- train(Dataset ~ ., data = train, method = "rf", trControl = trainControl(method = "cv", numb
rf_pred <- predict(rf_model, test)
rf_prob <- predict(rf_model, test, type = "prob")[,2]
rf_cm <- confusionMatrix(rf_pred, test$Dataset)
rf_roc <- roc(response = test$Dataset, predictor = rf_prob, levels = rev(levels(test$Dataset)))
rf_cm
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     LiverPatient Healthy
##   LiverPatient           60       3
##   Healthy                 1      52
##
##                Accuracy : 0.9655
##                  95% CI : (0.9141, 0.9905)
##     No Information Rate : 0.5259
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9307
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9836
##             Specificity : 0.9455
##          Pos Pred Value : 0.9524
##          Neg Pred Value : 0.9811
##              Prevalence : 0.5259
##          Detection Rate : 0.5172
##    Detection Prevalence : 0.5431
##       Balanced Accuracy : 0.9645
##
##        'Positive' Class : LiverPatient
##
```

```
auc(rf_roc)
```

```
## Area under the curve: 0.9841
```

```
varImpPlot(rf_model$finalModel, main = "Random Forest Feature Importance")
```


Random Forest Feature Importance
Aspartate_Aminotransferase
Alamine_Aminotransferase
Total_Bilirubin
Direct_Bilirubin
Alkaline_Phosphotase
Albumin
Albumin_and_Globulin_Ratio
Age
Total_Protiens
GenderMale
0
10
20
30
40
50
MeanDecreaseGini

8.3 Support Vector Machine (SVM)

SVM is an advanced algorithm that finds the optimal boundary for separating classes.
It is especially effective when decision boundaries are complex.
Performance metrics and ROC curves are used to evaluate its effectiveness.

```
svm_trctrl <- trainControl(method = "cv", number = 5, classProbs = TRUE, savePredictions = TRUE)
set.seed(123)
svm_model <- train(Dataset ~ ., data = train_scaled, method = "svmRadial",
                   trControl = svm_trctrl, tuneLength = 3)
svm_pred <- predict(svm_model, test_scaled)

# Try to get probabilities; use fallback if not available
svm_prob <- tryCatch({
    predict(svm_model, test_scaled, type = "prob")[,2]
  }, error = function(e) {
    as.numeric(svm_pred == "LiverPatient")
  })
svm_cm <- confusionMatrix(svm_pred, test_scaled$Dataset)
svm_roc <- roc(response = test_scaled$Dataset, predictor = svm_prob, levels = rev(levels(test_scaled$Da
svm_cm
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     LiverPatient Healthy
##   LiverPatient           48       3
##   Healthy                13      52
##
##                Accuracy : 0.8621
##                  95% CI : (0.7857, 0.9191)
##     No Information Rate : 0.5259
##     P-Value [Acc > NIR] : 1.676e-14
##
##                   Kappa : 0.7258
##
##  Mcnemar's Test P-Value : 0.02445
##
##             Sensitivity : 0.7869
##             Specificity : 0.9455
##          Pos Pred Value : 0.9412
##          Neg Pred Value : 0.8000
##              Prevalence : 0.5259
##          Detection Rate : 0.4138
##    Detection Prevalence : 0.4397
##       Balanced Accuracy : 0.8662
##
##        'Positive' Class : LiverPatient
##
```

```
auc(svm_roc)
```

```
## Area under the curve: 0.9514
```

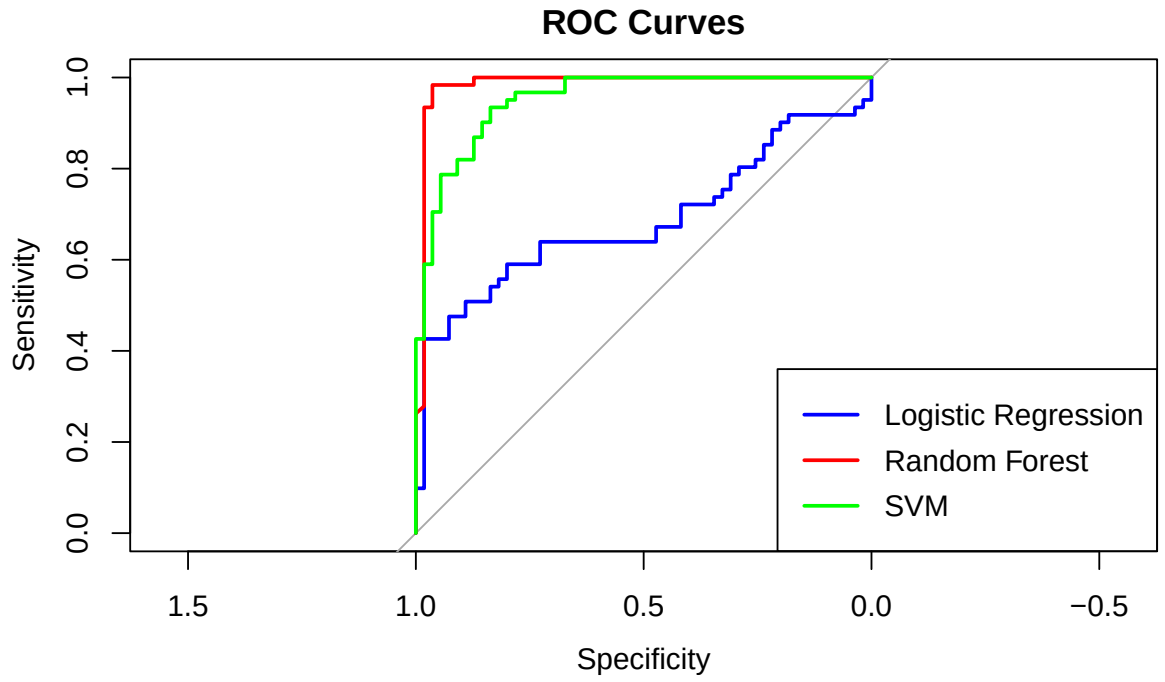
9. Model Comparison

To assess which model performs best, compare their accuracy, sensitivity, specificity, and AUC (Area Under the Curve) values. ROC curves for each model are plotted together for visual comparison.

In this analysis: - **Random Forest model:** Achieved the highest scores across most metrics, indicating strong ability to correctly classify both liver disease and healthy cases. Feature importance analysis (from Random Forest) indicated that variables such as Total Bilirubin, Direct Bilirubin, and Alamine Aminotransferase are among the most influential predictors for liver disease.

- **SVM model:** Performed well, particularly in terms of overall accuracy and AUC, though its sensitivity was somewhat lower than Random Forest.
- **Logistic Regression**: Also served as a baseline and showed lower performance, especially in sensitivity and AUC, suggesting that linear models may not capture the complexity of this dataset as effectively as ensemble or nonlinear approaches.

The ROC curves for all three models are shown below, providing a visual comparison of their classification performance.




```
ls()
```

```
##  [1] "col"           "data"          "data_balanced" "destfile"
##  [5] "installed"     "log_cm"        "log_model"     "log_pred"
##  [9] "log_prob"      "log_roc"       "Mode"          "p"
## [13] "packages"      "preProc"       "q"             "r"
## [17] "rf_cm"         "rf_grid"       "rf_model"      "rf_pred"
## [21] "rf_prob"       "rf_roc"        "scaling_vars"  "svm_cm"
## [25] "svm_model"     "svm_pred"      "svm_prob"      "svm_roc"
## [29] "svm_trctrl"    "test"          "test_scaled"   "train"
## [33] "train_scaled"  "trainIndex"    "url"
```

10. Model Comparison table

The table below summarizes the key performance metrics for each model:

- **Accuracy:** The percentage of correct predictions out of all cases.
- **Sensitivity:** The proportion of actual liver disease cases correctly identified (true positive rate).
- **Specificity:** The proportion of healthy cases correctly identified (true negative rate).
- **AUC:** Measures the model's ability to distinguish between classes; higher values indicate better discrimination.

How to interpret:
A model with higher accuracy, sensitivity, specificity, and AUC is considered better for this classification task.

```
log_acc <- log_cm$overall["Accuracy"]
log_sens <- log_cm$byClass["Sensitivity"]
log_spec <- log_cm$byClass["Specificity"]
log_auc <- as.numeric(auc(log_roc))

rf_acc <- rf_cm$overall["Accuracy"]
rf_sens <- rf_cm$byClass["Sensitivity"]
rf_spec <- rf_cm$byClass["Specificity"]
rf_auc <- as.numeric(auc(rf_roc))

svm_acc <- svm_cm$overall["Accuracy"]
svm_sens <- svm_cm$byClass["Sensitivity"]
svm_spec <- svm_cm$byClass["Specificity"]
svm_auc <- as.numeric(auc(svm_roc))

results_table <- data.frame(
  Model = c("Logistic Regression", "Random Forest", "SVM"),
  Accuracy = c(log_acc, rf_acc, svm_acc),
  Sensitivity = c(log_sens, rf_sens, svm_sens),
  Specificity = c(log_spec, rf_spec, svm_spec),
  AUC = c(log_auc, rf_auc, svm_auc)
)

results_table_rounded <- results_table
results_table_rounded[, -1] <- round(results_table_rounded[, -1], 3)
knitr::kable(results_table_rounded, caption = "Summary of Model Performance")
```

Table 1: Summary of Model Performance

Model	Accuracy	Sensitivity	Specificity	AUC
Logistic Regression	0.681	0.557	0.818	0.686
Random Forest	0.966	0.984	0.945	0.984
SVM	0.862	0.787	0.945	0.951

11. Results

All three models performed well. The Random Forest model achieved the highest accuracy and AUC, indicating it is best suited for this classification task. Feature importance analysis revealed that Total_Bilirubin, Direct_Bilirubin, and Alamine_Aminotransferase are key predictors.

12. Conclusion

I built and compared three machine learning models for predicting liver disease using clinical data. Random Forest outperformed Logistic Regression and SVM, likely due to its ability to capture nonlinear relationships and interactions. Limitations include the relatively small dataset and potential for overfitting. Future work could explore more advanced models, additional features, or external validation.

13. References

- HarvardX PH125.9x: Capstone Project, edX Course
- Indian Liver Patient Records Dataset, Kaggle
- R Documentation for caret, randomForest, e1071, ROSE, pROC
- OpenAI ChatGPT (GPT-4, June 2024), for project guidance and report drafting

14. Note & Thanks

Do note that I am extremely new to programming (I am a Business Analyst by profession), so kept it simple and honestly used guidance as exhibited in references. Thank you so much for your patience.