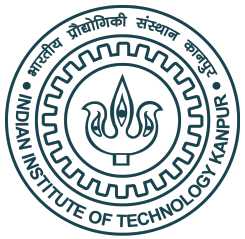
# Nonparametric Tree Based Algorithms

R. Vashistha* R. Saha* S. Bhowmik*

*Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

April 16, 2022




Supervised by: Dr. Dootika Vats

# Contents

# Classification and Regression Trees (CART)

Proposed by Breiman et al. (1984)

- Trees partition the feature space into a number of heterogeneous regions.
- Prediction for a given observation is made using the mean or the mode of response value for the training observations in the region to which it belongs.

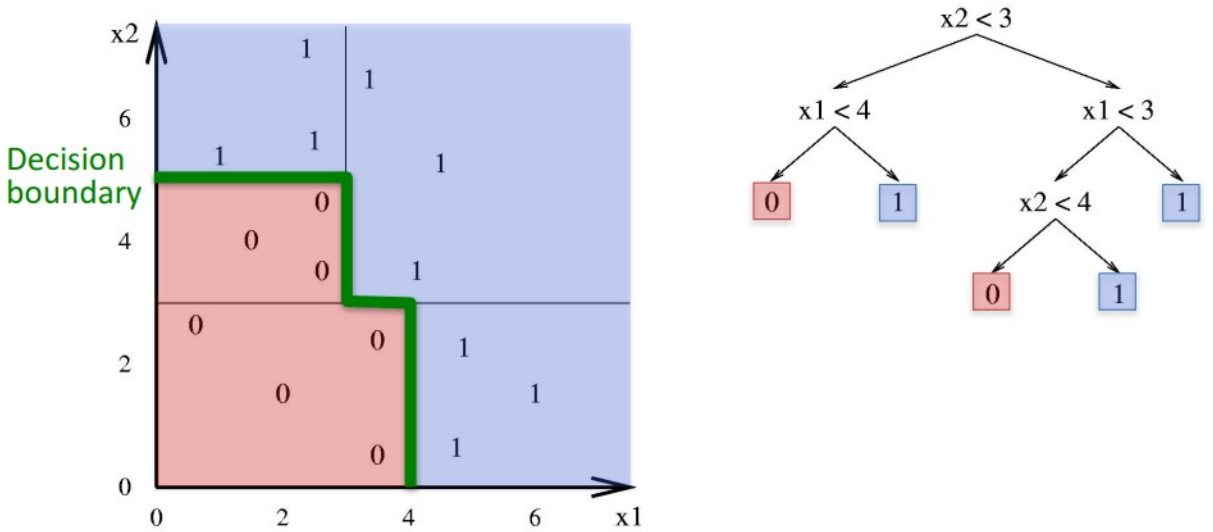



Figure: An example of a decision tree

# Problems associated with CART

- High variance and unstability
- Prone to overfitting

# Why Random Forest (RF)?

- Overcomes the issues of high variance by "averaging" over multiple independently grown random CARTs.
- Shown to outperform CART Breiman (2001).
- Similar level of bias as CART but with significantly low variance.
- Performs well with very little tuning required.

# Algorithm

(Hastie et al. (2009))

- For m trees, i=1 to m
  a) Draw bootstrap sample from the training data
  b) Grow a random-forest tree $g_i$ to the bootstrapped data, by recursively selecting k variables at random from the p variables.
- Output the ensemble of trees $\{g_i\}_1^m$
- To make a prediction at new point x:
  Regression:

$$\hat{g}_{rf}^m(x) = \frac{1}{m}\sum_{i=1}^{m} g_i(x)$$

  Classification: if $\hat{C}_{rf}^m(x)$ is the class prediction of $i^{th}$ random forest tree, Then

$$\hat{C}_{rf}^m(x) = \text{ majority vote } \{\hat{C}_i(x)\}_1^m$$

# Algorithm

(Hastie et al. (2009))

- For m trees, i=1 to m
  a) Draw bootstrap sample from the training data
  b) Grow a random-forest tree $g_i$ to the bootstrapped data, by recursively selecting k variables at random from the p variables.
- Output the ensemble of trees $\{g_i\}_1^m$
- To make a prediction at new point x:
  Regression:

$$\hat{g}_{rf}^m(x) = \frac{1}{m}\sum_{i=1}^{m} g_i(x)$$

Classification: if $\hat{C}_{rf}^m(x)$ is the class prediction of $i^{th}$ random forest tree, Then

$$\hat{C}_{rf}^m(x) = \text{majority vote } \{\hat{C}_i(x)\}_1^m$$

# Algorithm

(Hastie et al. (2009))

- For m trees, i=1 to m
  a) Draw bootstrap sample from the training data
  b) Grow a random-forest tree $g_i$ to the bootstrapped data, by recursively selecting k variables at random from the p variables.
- Output the ensemble of trees $\{g_i\}_1^m$
- To make a prediction at new point x:
  Regression:

$$\hat{g}_{rf}^m(x) = \frac{1}{m}\sum_{i=1}^{m} g_i(x)$$

  Classification: if $\hat{C}_{rf}^m(x)$ is the class prediction of $i^{th}$ random forest tree, Then

$$\hat{C}_{rf}^m(x) = \textit{ majority vote } \{\hat{C}_i(x)\}_1^m$$

# Why Bayesian Additive Regression Trees (BART) ?

- Bayesian alternative to boosted regression trees
  - Intelligent priors for the depth of each tree and the shrinkage factor
- Embeds what is normally an algorithmic approach in a likelihood framework to produce coherent uncertainty intervals, unusual for machine learning approaches.

# Model

BART sum-of-trees model Chipman et al. (2010) :

$$Y = g(x; T_1, M_1) + \ldots + g(x; T_m, M_m) + \varepsilon = f(x) + \varepsilon$$

- Tree model $(T, M)$
  - $T$ is a binary tree
  - $M = \{\mu_1, \mu_2, \ldots, \mu_b\}$ is the vector of means in the $b$ terminal nodes of the tree
- $g(x; T, M)$ : value obtained by following observation $x$ down tree and returning mean for the terminal node in which it lands
- $\varepsilon \sim N(0, \sigma^2)$

# Model

BART sum-of-trees model Chipman et al. (2010) :

$$Y = g(x; T_1, M_1) + \ldots + g(x; T_m, M_m) + \varepsilon = f(x) + \varepsilon$$

- Tree model $(T, M)$
  - $T$ is a binary tree
  - $M = \{\mu_1, \mu_2, \ldots, \mu_b\}$ is the vector of means in the $b$ terminal nodes of the tree
- $g(x; T, M)$ : value obtained by following observation $x$ down tree and returning mean for the terminal node in which it lands
- $\varepsilon \sim N(0, \sigma^2)$

# BART Prior

The prior of the BART model can be written as

$$\pi(\theta) = \pi((T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \sigma).$$

$\pi$ wants:

- Each $T$ small.
- Each $\mu$ small.
- "nice" $\sigma$ (smaller than least squares estimate).

We refer to $\pi$ as a regularization prior because it keeps the overall fit small.
In addition, it keeps the contribution of each $g(x; T_i, M_i)$ model component small.

# BART Prior

The prior of the BART model can be written as

$$\pi(\theta) = \pi((T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \sigma).$$

$\pi$ wants:

- Each $T$ small.
- Each $\mu$ small.
- "nice" $\sigma$ (smaller than least squares estimate).

We refer to $\pi$ as a regularization prior because it keeps the overall fit small.
In addition, it keeps the contribution of each $g(x; T_i, M_i)$ model component small.

# BART Prior

The prior of the BART model can be written as

$$\pi(\theta) = \pi((T_1, M_1), (T_2, M_2), \ldots, (T_m, M_m), \sigma).$$

$\pi$ wants:

- Each $T$ small.
- Each $\mu$ small.
- "nice" $\sigma$ (smaller than least squares estimate).

We refer to $\pi$ as a regularization prior because it keeps the overall fit small.
In addition, it keeps the contribution of each $g(x; T_i, M_i)$ model component small.

# Simulation Scenario - I

- $Y_i = f(10sin(\pi X_{1i} X_{2i}) + 15(X_{3i} - 0.5)^2 - 15X_{4i} + 5X_{5i} + \varepsilon_i)$, where f(.) is the logistic function,
- $X_i \sim U(0,1)$,
- $\varepsilon \sim$ N(0,1),
- $p \in \{20, 50, 100, 500, 1000\}$,
- $n = 1000$

# RF vs BART I

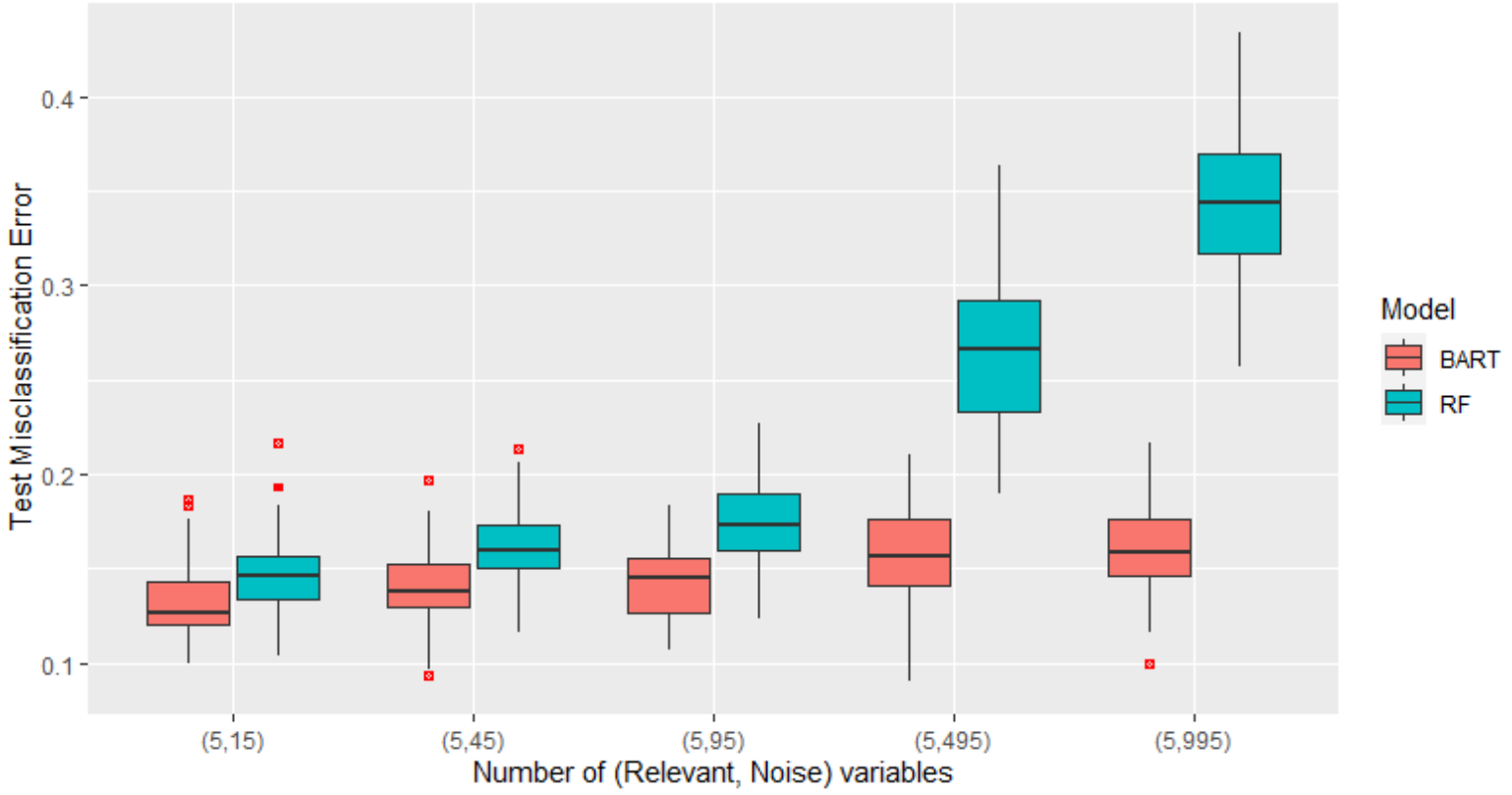



Figure: Comparison of RF and BART on increasing number of noise variables. Results are based on 50 simulations for each pair, with a training sample of 700 and a test sample of 300

# Simulation Scenario - II

- $Y_i = f(\beta' X_i + \varepsilon_i)$, where f(.) is the logistic function.
- $X_i \sim N(0, \Sigma)$,
  - $\Sigma_{jj} = 1, \Sigma_{jk} = 0.3$
- $\varepsilon \sim$ Cauchy distribution
- $$\beta = (\underbrace{3, \ldots, 3}_{p_0}, \underbrace{0, \ldots, 0}_{p - p_0})$$
- $p \in \{10, 20, 30, 50\}$
- $p_0 \in \{5, 15, 25, 45\}$
- $n = 1000$

# RF vs BART II

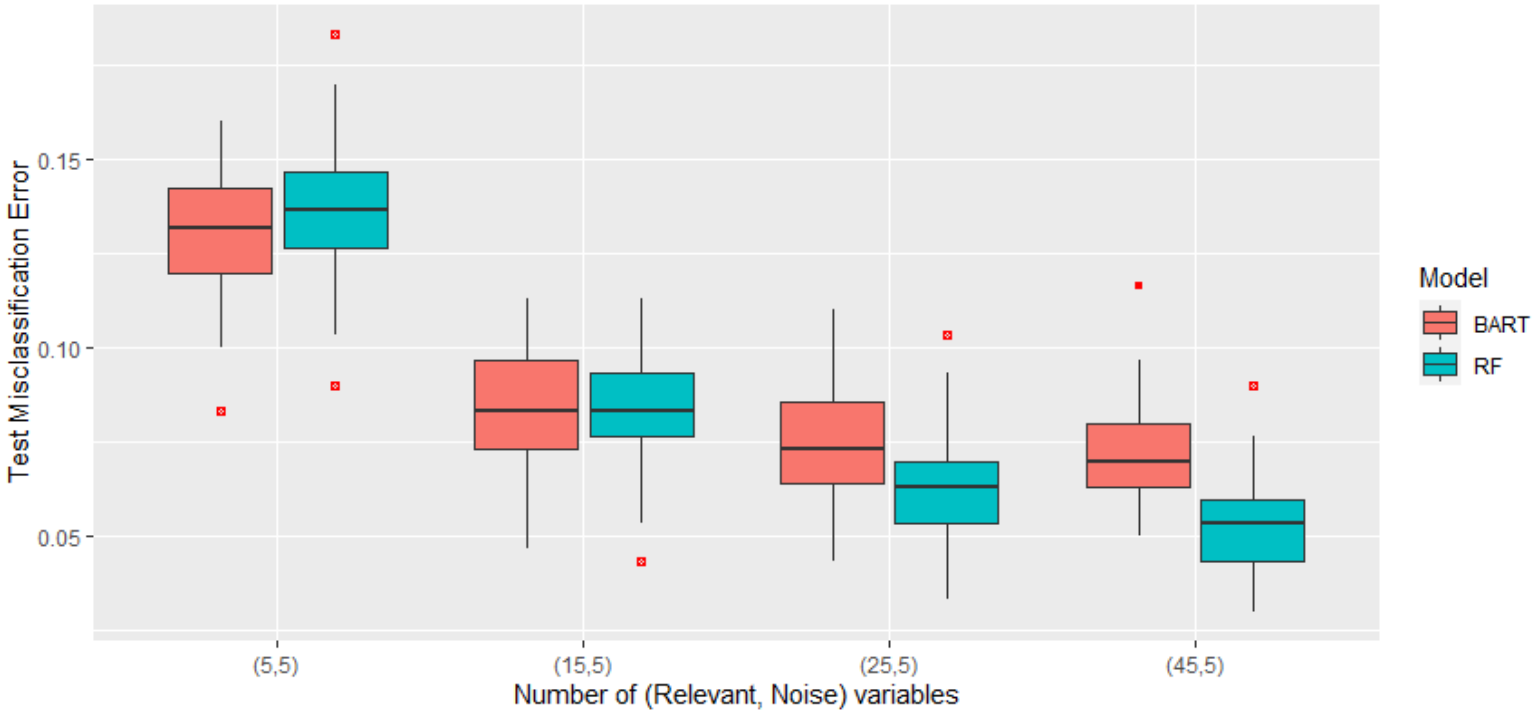



Figure: Comparison of RF and BART on decreasing number of noise variables. Results are based on 50 simulations for each pair, with a training sample of 700 and a test sample of 300

# Data Description

- **"Framingham Heart Disease Dataset"**
  - n = 4238
  - p = 16
  - 645 missing values
- We have applied **"Multivariate Imputation via Chained Equations" (MICE)** to handle the missing values.
- Since the dataset is highly imbalanced, we have applied **Oversampling** technique to make the dataset balanced.

# Data Description

- **"Framingham Heart Disease Dataset"**
  - n = 4238
  - p = 16
  - 645 missing values
- We have applied **"Multivariate Imputation via Chained Equations" (MICE)** to handle the missing values.
- Since the dataset is highly imbalanced, we have applied **Oversampling** technique to make the dataset balanced.

# Data Description

- **"Framingham Heart Disease Dataset"**
  - n = 4238
  - p = 16
  - 645 missing values
- We have applied **"Multivariate Imputation via Chained Equations" (MICE)** to handle the missing values.
- Since the dataset is highly imbalanced, we have applied **Oversampling** technique to make the dataset balanced.

# Applying Models on the Data

- We have used the train set to train the models using **Logistic Regression, CART, Random Forest, BART.**
- After training each model, we have applied the model on cross validation set.
- We have computed F-Score and obtained ROC-AUC curve for each case and chosen that model to be the best that has the highest F-Score and AUC.
- Finally we have applied the best chosen model on the test set and looked for final F-Score and AUC.

# Applying Models on the Data

- We have used the train set to train the models using **Logistic Regression, CART, Random Forest, BART.**
- After training each model, we have applied the model on cross validation set.
- We have computed F-Score and obtained ROC-AUC curve for each case and chosen that model to be the best that has the highest F-Score and AUC.
- Finally we have applied the best chosen model on the test set and looked for final F-Score and AUC.

# Applying Models on the Data

- We have used the train set to train the models using **Logistic Regression, CART, Random Forest, BART.**
- After training each model, we have applied the model on cross validation set.
- We have computed F-Score and obtained ROC-AUC curve for each case and chosen that model to be the best that has the highest F-Score and AUC.
- Finally we have applied the best chosen model on the test set and looked for final F-Score and AUC.

# ROC Curve

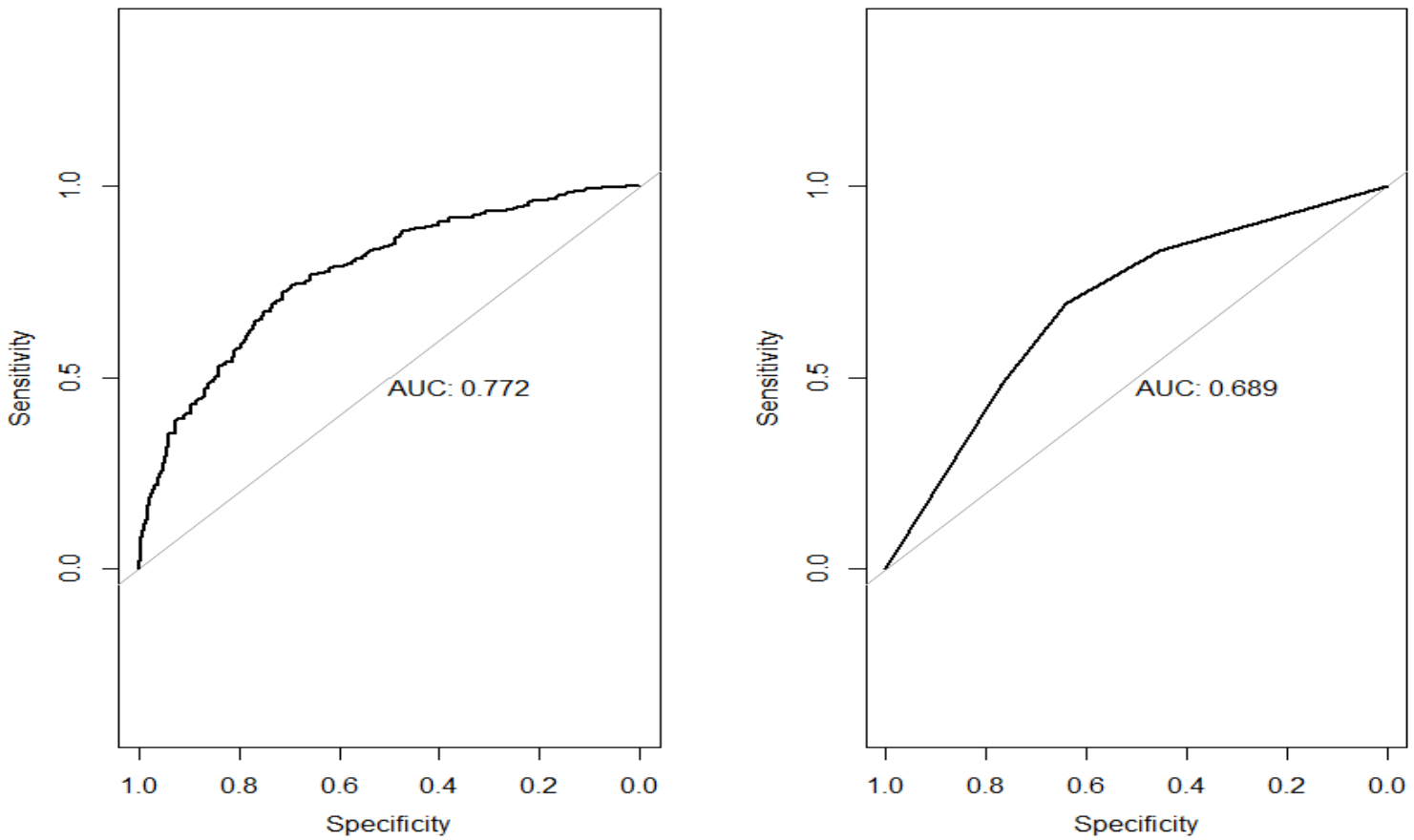



Figure: ROC Curve for Logistic Regression and CART respectively

# ROC Curve

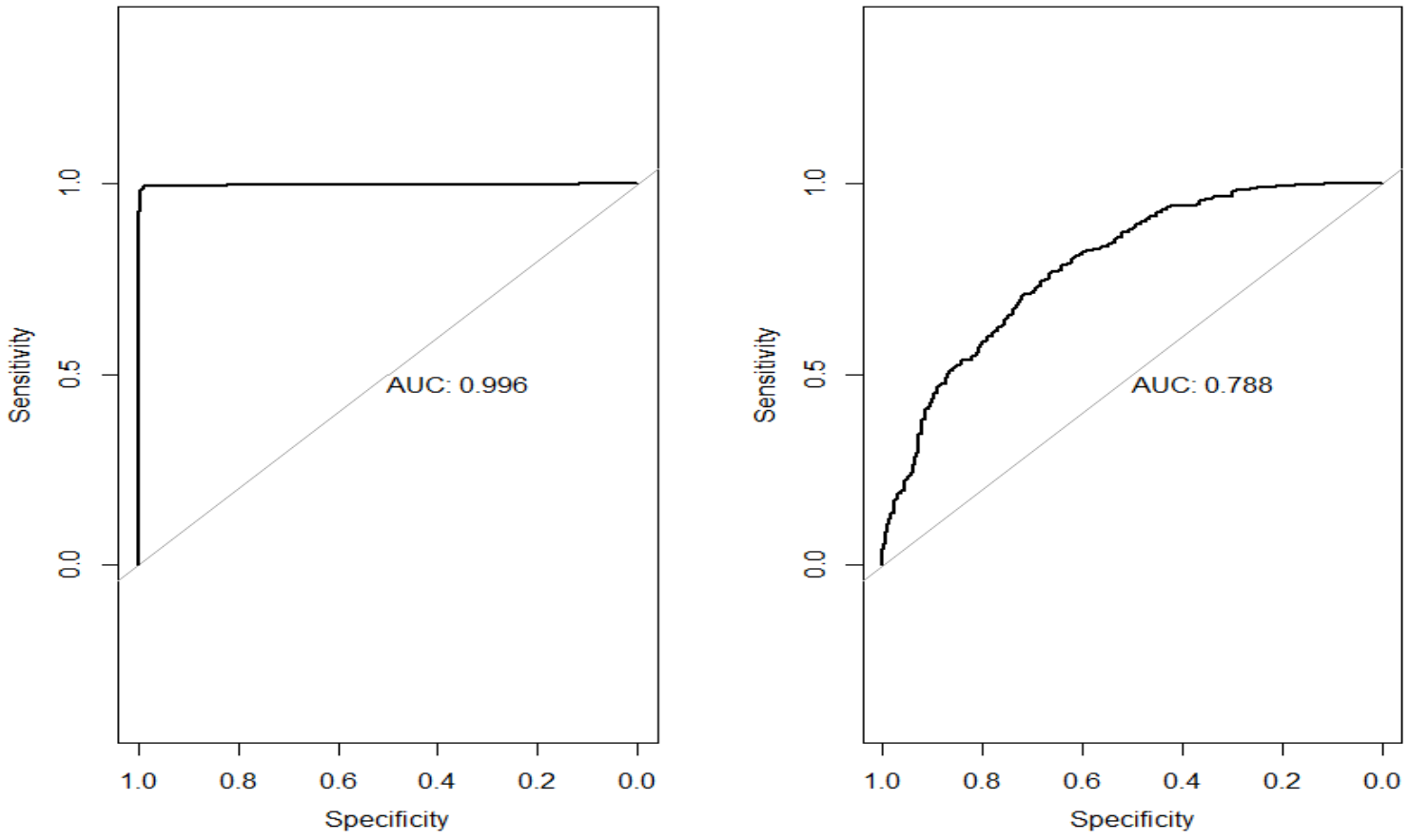



Figure: ROC Curve for RF and BART respectively

# Comparing F-Score and AUC

| Method | F-Score | AUC |
|---|---|---|
| Logistic Regression | 0.6916 | 0.7723 |
| Decision Tree | 0.6657 | 0.6891 |
| Random Forest | 0.9791 | 0.9965 |
| BART | 0.7177 | 0.7880 |

- From the above table, we can see that, Random Forest outperforms all the other models.
- So, **we conclude that Random Forest is the best model to predict if a person has the risk of ten year coronary heart disease or not.**

# Comparing F-Score and AUC

| Method | F-Score | AUC |
|---|---|---|
| Logistic Regression | 0.6916 | 0.7723 |
| Decision Tree | 0.6657 | 0.6891 |
| Random Forest | 0.9791 | 0.9965 |
| BART | 0.7177 | 0.7880 |

- From the above table, we can see that, Random Forest outperforms all the other models.
- So, **we conclude that Random Forest is the best model to predict if a person has the risk of ten year coronary heart disease or not.**

# Test Data Results

| Evaluation Metric | Values |
|---|---|
| F-Score | 0.9665 |
| AUC | 0.9874 |

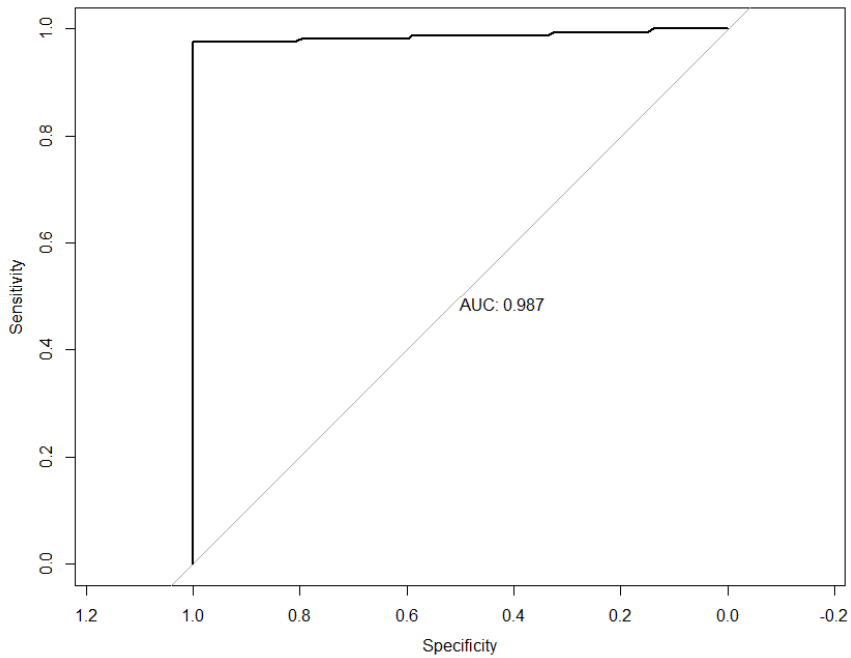



Figure: ROC curve for RF on test data

# Final Remarks

- Ensemble methods based on trees such as BART and RF perform very well empirically in comparison to other similar methods.
- BART performs better than RF in presence of high amount of 'noise' variables.
- RF performs well in situations with low numbers of noise variables and does not necessarily need tuning.
- For predicting the risk of coronary heart disease, RF gave significantly better prediction results than other competing methods.
- Open problems: Interpretability and computation issues in case of Big Data.

# Final Remarks

- Ensemble methods based on trees such as BART and RF perform very well empirically in comparison to other similar methods.
- BART performs better than RF in presence of high amount of 'noise' variables.
- RF performs well in situations with low numbers of noise variables and does not necessarily need tuning.
- For predicting the risk of coronary heart disease, RF gave significantly better prediction results than other competing methods.
- Open problems: Interpretability and computation issues in case of Big Data.

# Final Remarks

- Ensemble methods based on trees such as BART and RF perform very well empirically in comparison to other similar methods.
- BART performs better than RF in presence of high amount of 'noise' variables.
- RF performs well in situations with low numbers of noise variables and does not necessarily need tuning.
- For predicting the risk of coronary heart disease, RF gave significantly better prediction results than other competing methods.
- Open problems: Interpretability and computation issues in case of Big Data.

# Final Remarks

- Ensemble methods based on trees such as BART and RF perform very well empirically in comparison to other similar methods.
- BART performs better than RF in presence of high amount of 'noise' variables.
- RF performs well in situations with low numbers of noise variables and does not necessarily need tuning.
- For predicting the risk of coronary heart disease, RF gave significantly better prediction results than other competing methods.
- Open problems: Interpretability and computation issues in case of Big Data.

# Final Remarks

- Ensemble methods based on trees such as BART and RF perform very well empirically in comparison to other similar methods.
- BART performs better than RF in presence of high amount of 'noise' variables.
- RF performs well in situations with low numbers of noise variables and does not necessarily need tuning.
- For predicting the risk of coronary heart disease, RF gave significantly better prediction results than other competing methods.
- Open problems: Interpretability and computation issues in case of Big Data.

# Thank You!

References:

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. Chapman and Hall.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sparapani, R., Spanbauer, C., and McCulloch, R. (2021). Nonparametric machine learning and efficient computation with Bayesian additive regression trees: The BART R package. *Journal of Statistical Software*, 97(1):1–66.

# Appendix

# Computation

Software: R Core Team (2020)

- BART: Sparapani et al. (2021)
- Random Forest: Liaw and Wiener (2002)

# BART Prior

- The $T_j$ prior, p($T_j$), is specified by three aspects:
  - Probability that a node at depth d = (0,1,2,...) is nonterminal is given by: $\frac{\alpha}{(1+d)^{\beta}}$ with $\alpha \in (0,1)$ and $\beta \in [0,\infty)$
  - Uniform distribution on splitting variable assignments at each interior node.
  - Uniform distribution on splitting rule assignment at each interior node conditional on the splitting variable.
- The $\mu_{ij}|T_j$ prior: $\mu_{ij} \sim N(0, {\sigma_\mu}^2)$
- The $\sigma$ prior: Inverse Chi-Square distribution.

# BART MCMC

The model/prior is described by

$$Y = g(x; T_1, M_1) + \ldots + g(x; T_m, M_m) + \varepsilon$$

plus

$$((T_1, M_1), \ldots . (T_m, M_m), \sigma)$$

First, it is a "simple" Gibbs sampler:

$$(T_i, M_i) \,|\, (T_1, M_1, \ldots, T_{i-1}, M_{i-1}, T_{i+1}, M_{i+1}, \ldots, T_m, M_m, \sigma)$$
$$\sigma \,|\, (T_1, M_1, \ldots, \ldots, T_m, M_m)$$

To draw $(T_i, M_i) \,|$. we subtract the contributions of the other trees from both sides to get a simple one-tree model.
We integrate out $M$ to draw $T$ and then draw $M \,|\, T$.

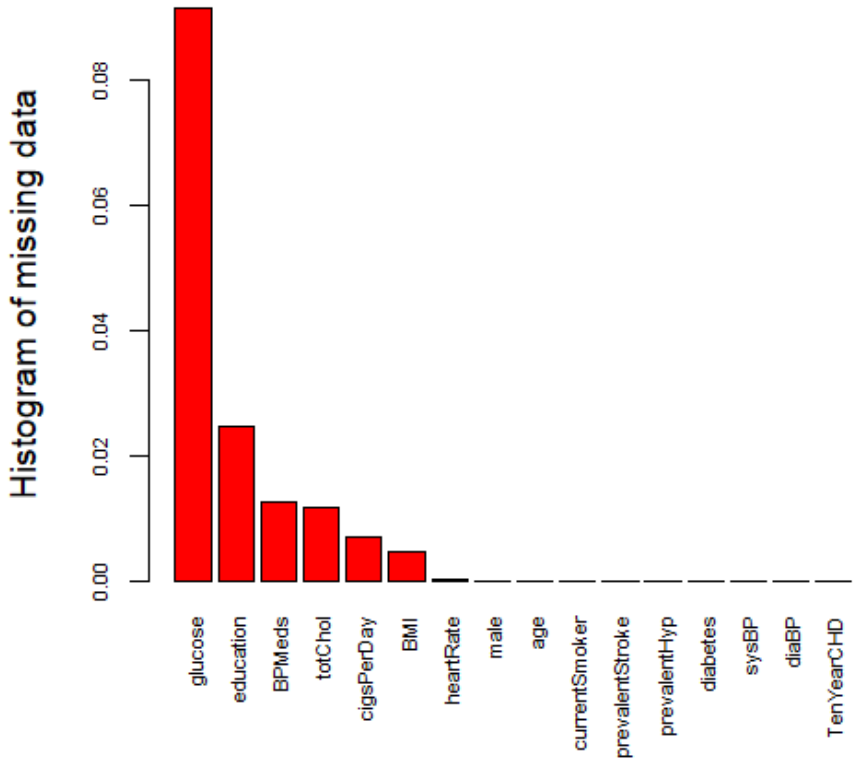
# Data info.

The dataset is available on the Kaggle, it is a study on residents of the town of Framingham, Massachusetts. The classification goal is to predict 10y risk of coronary heart disease (CHD). It contains 15 attributes like age, sex, Education, Smoker, Cigs/day, BP, Cholesterol, BMI, Heart rate, Glucose etc.

# MICE algorithm

Multiple imputation using chained equations or MICE uses multiple imputations as opposed to single imputation. It is regarded as a fully conditional specification or sequential regression multiple imputation. It has become one of the principal methods of addressing missing data.
(i) Do mean imputation temporarily ("place holder")
(ii) Set back to missing the temporary "place holder" imputations for some variables
(iii) Regress that variable on the other variables containing missing values while fitting, take only those rows, where dependent variable had no missing values
(iv) From regression equation predict the place holder values obtained in (ii)
(v) Repeat (ii)$-$ (iv)

# Missingness in Data



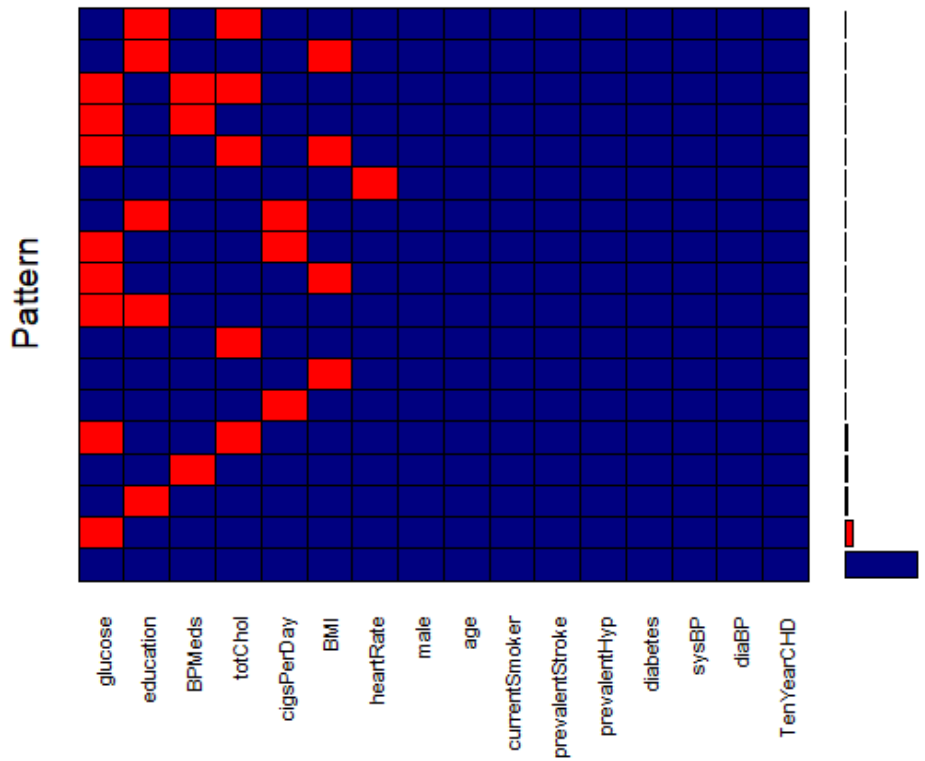




Figure: Patterns in Missing Values

# Imbalanced data

We have used 'Bootstrap' technique with minor class to deal with Imbalance data.
Randomly over-sampled by selecting random samples from minority class with replacement.
The flaws of this technique includes computational cost, Over-fitting, etc.

# Logistic Regression

After applying the logistic regression model on Cross Validation Set, we have got the following result,-

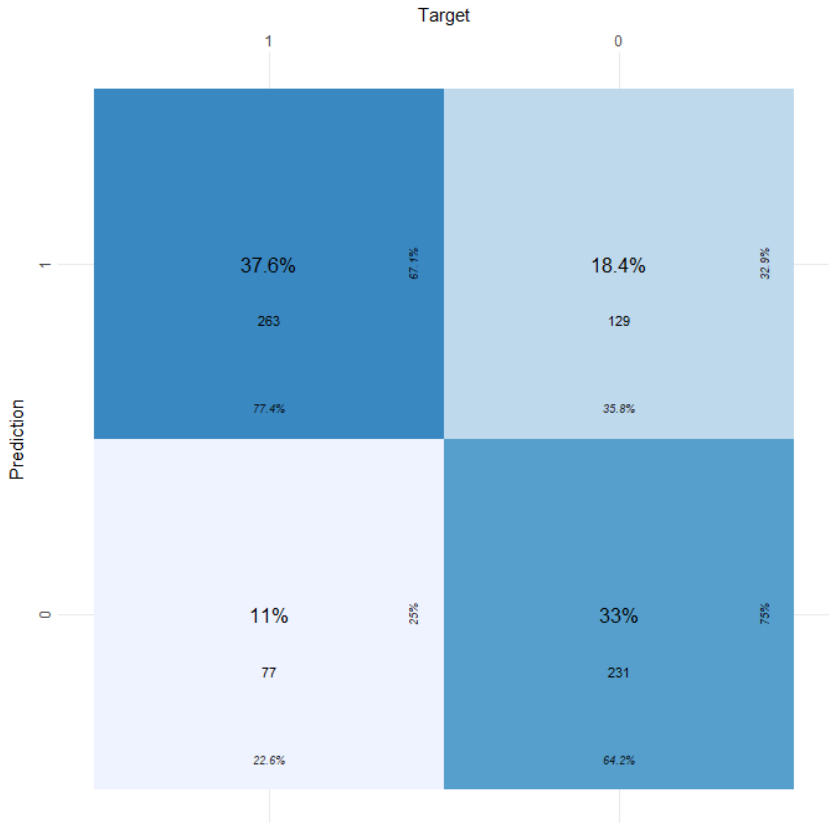



Figure: Plot 7

# Decision Tree

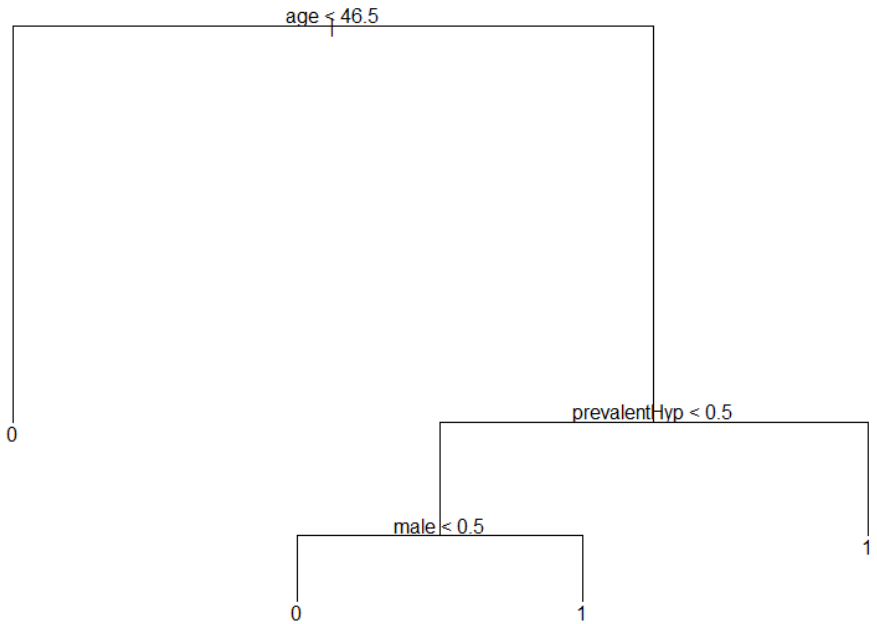



Figure: Plot 8

# Decision Tree

After applying the model on cross validation set, we have obtained the following result,

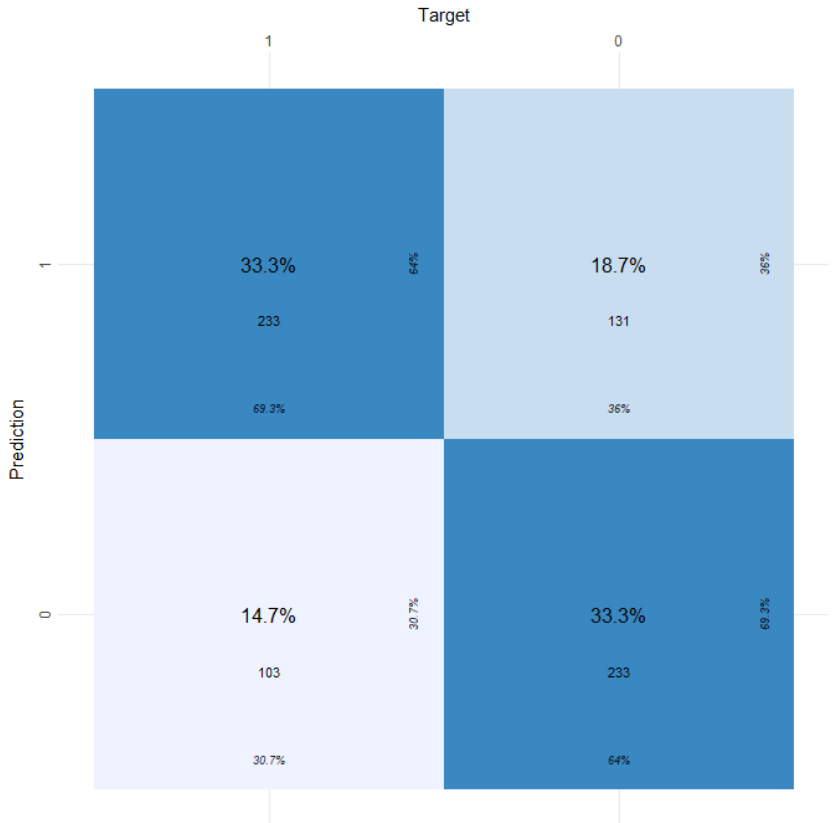



Figure: Plot 9

# Random Forest

After applying the decision tree model on cross validation set, we have obtained the following result,

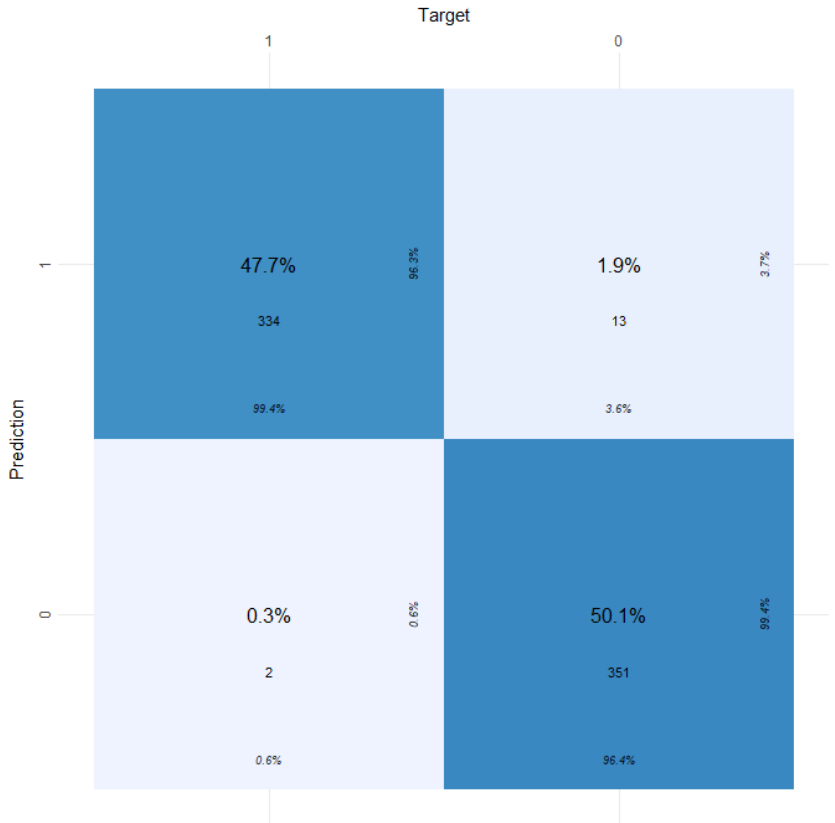



Figure: Plot 10

# BART

After applying the random forest model on cross validation set, we have obtained the following result,

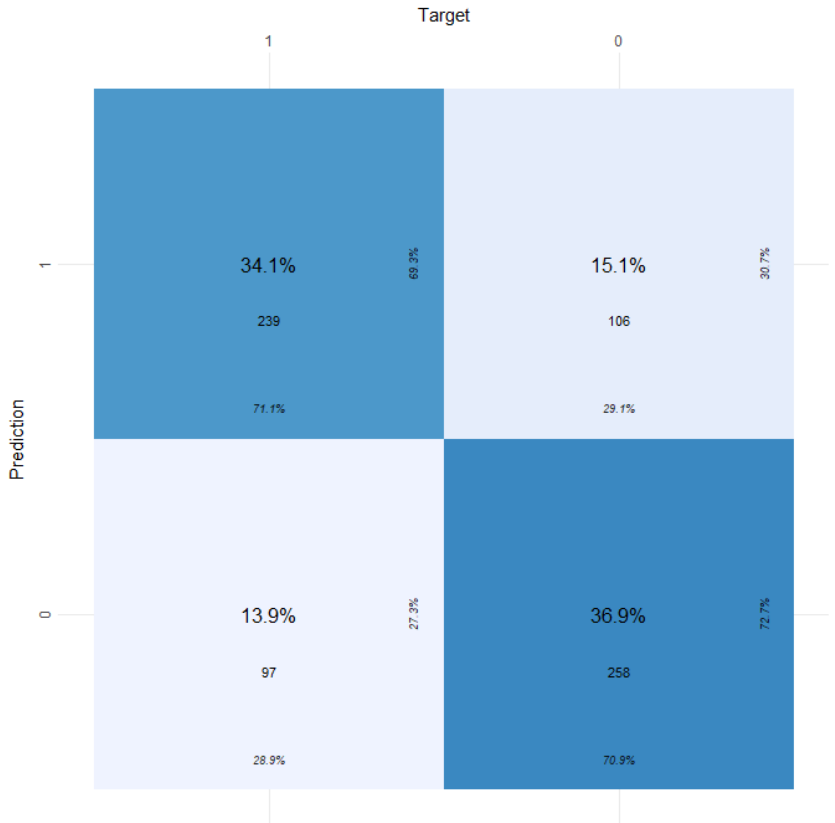



Figure: Plot 11

# Applying Final Model (RF) on Test Data

After applying the decision tree model on test set, we have obtained the following result,

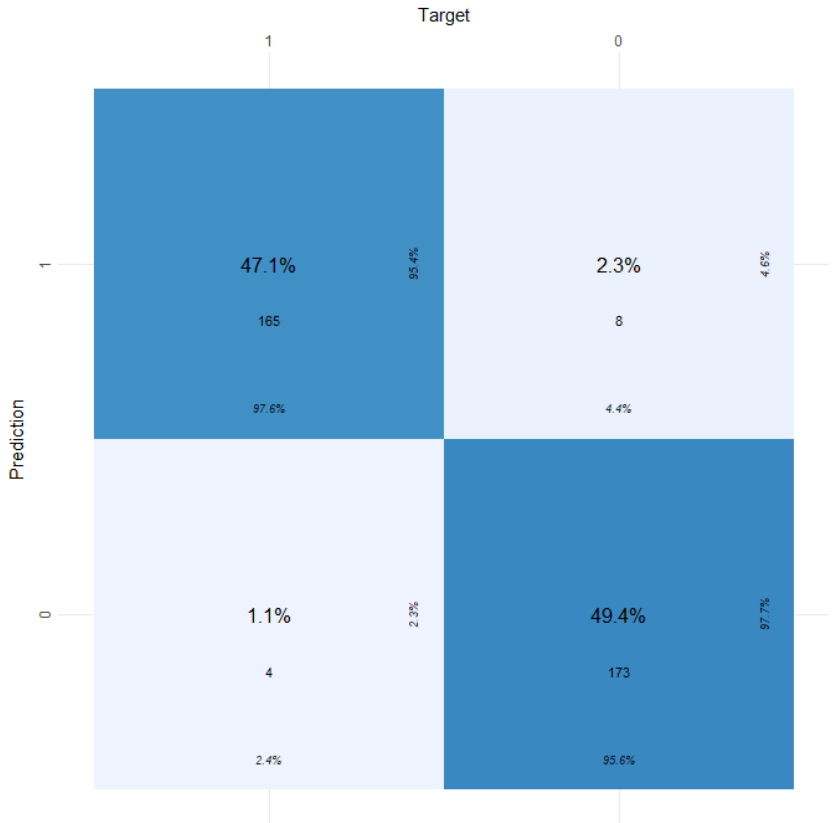



Figure: Plot 5