Heart disease Prediction with different Supervised Learning Models using R

Predicting Heart disease using different Supervised Learning Models to find out which of these models are good at predicting having heart disease (positive) and not having heart disease (negative) cases correctly.

The data has collected on health profile parameters of people showing symptoms of heart disease and their diagnostic results are given in the Heart_Disease_Data file. The list of health profile features on which data is collected are given below:

S.No.	Feature Name	Description
1	Age	Age
2	Sex	Sex
3	CP	Chest pain type
4	RestBP	Resting blood pressure
5	Cholesterol	Serum cholesterol in mg/dl
6	FBP	Fasting blood sugar > 120 mg/dl
7	RestECG	Resting electrocardiographic results
8	Max_HR	Maximum heart rate achieved
9	ExAngina	Exercise-induced angina
10	Oldpeak	ST depression induced by exercise relative to rest
11	Slope	The slope of the peak exercise ST segment
12	CA	Number of major vessels (0-3) colored by flourosopy
13	Thal	3 = normal; 6 = fixed defect; 7 = reversible defect

This dataset is having 13 explanatory variables and 1 response variable namely Result which is composed of binary values such as 1 (positive) and 0 (negative) for having heart disease.

Since the Response Variable in the dataset is categorical value (binary), we perform the following Supervised Learning Models to check which of these models suits better.

- Logistic Regression
- Classification Tree
- Bagging
- Random Forest
- Naïve Bayes
- K Nearest Neighbors
- Support Vector Machine

This dataset comprises of 5 of the variables as numerical (both continuous & discrete) and the remaining 9 of its variables are categorical.

Descriptive summary of the features in the dataset Heart_Disease_Data file:

	Age	*RestBP*	*Cholesteral*	*Max_HR*	*Oldpeak*
Min	29	94	126	71	0
1st Qu	47.50	120	211	133.5	0
Median	55	130	240	153	0.80
Mode	54.37	131.6	246.3	149.6	1.04
3rd Qu	61	140	274.5	166	1.60
Max	77	200	564	202	6.20

Sex	**CP**	**FBP**	**RestECG**	**ExAngina**	**Slope**	**CA**	**Thal**	**Result**
0 – 96	0 – 143	0 – 258	0 – 147	0 – 204	0 – 21	0 – 175	0 – 2	0 – 138
1 - 207	1 – 50	1 - 45	1 – 152	1 – 99	1 – 140	1 – 65	1 – 18	1 - 165
	2 – 87		2 - 4		2 – 142	2 – 38	2 – 166	
	3 - 23					3 – 20	3 – 117	
						4 - 5		

After Splitting the Heart_Disease_Data file of 303 observations randomly into training data (80%) and test data (20%) we get:

- training data is having 254 observations/records with 14 variables
- test data is having 49 observations/records with 14 variables

Logistic Regression:

Fitting the Logistic Regression model into the training data we get

Coefficients:

	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	3.308881	2.777817	1.191	0.23358
Age	-0.009372	0.025712	-0.364	0.71549
Sex	-1.677954	0.524706	-3.198	0.00138 **
CP	0.927122	0.223258	4.153	3.29e-05 ***
RestBP	-0.023732	0.011920	-1.991	0.04649 *
Cholesteral	-0.004595	0.004172	-1.101	0.27078
FBP	0.202298	0.624153	0.324	0.74585
RestECG	0.459137	0.391617	1.172	0.24103
Max_HR	0.026677	0.011846	2.252	0.02432 *
ExAngina	-1.104905	0.484340	-2.281	0.02253 *
Oldpeak	-0.505632	0.245519	-2.059	0.03945 *
Slope	0.380050	0.429940	0.884	0.37672
CA	-0.831948	0.212843	-3.909	9.28e-05 ***
Thal	-0.644373	0.333442	-1.932	0.05330 .

From the Coefficient table we observe that for Age,RestBP,Cholesteral,FBP,RestECG,Slope the p value is >0.05. So we can check through ANOVA test, which tells which of the variables are significant.

	Df	Deviance Resid.	Df Resid.	Dev	Pr(>Chi)
NULL			253	349.85	
Age	1	16.844	252	333.00	4.058e-05 ***
Sex	1	22.092	251	310.91	2.599e-06 ***
CP	1	63.116	250	247.80	1.949e-15 ***
RestBP	1	7.424	249	240.37	0.006435 **
Cholesteral	1	1.679	248	238.69	0.195070
FBP	1	0.014	247	238.68	0.905622
RestECG	1	0.942	246	237.74	0.331717
Max_HR	1	22.641	245	215.09	1.952e-06 ***
ExAngina	1	6.371	244	208.72	0.011598 *

Oldpeak	1	15.936	243	192.79	6.552e-05 ***
Slope	1	0.099	242	192.69	0.753147
CA	1	18.595	241	174.09	1.616e-05 ***
Thal	1	3.720	240	170.37	0.053776 .

From the ANOVA test we observe that only for Cholesteral, FBP, RestECG, Slope the p value is >0.05. Hence these variables must be removed from the model to make the model significant.

Therefore, the mathematical expression of the new model becomes:

y = 3.21524 –0.01264*Age –1.47520*Sex +0.90583*CP -0.023382*RestBP +0.02640* Max_HR -1.12766*ExAngina -0.62145*Oldpeak -0.77124*CA -0.64812*Thal

To check whether our model is significant or not, we need to first create a null model and compare it with our model.

Resid.	Df Resid.	Dev	Df	Deviance	Pr(>Chi)
1	253	349.85			
2	244	174.83	9	175.02	< 2.2e-16 ***

From the above ANOVA result of null model we observe that the p value is <0.05 which implies that our model is superior to the null model. Hence we can conclude that the model is significant.

Actual Vs Predicted table for training data:

	Predicted Class	
Actual	0	1
0	92	23
1	15	124

	Predicted Class	
Actual	0	1
0	36.22	9.06
1	5.91	48.82

- accuracy % = (36.22 + 48.82) = 85.03 %
- misclassification % = (5.91 + 9.06) = 14.97 %

From the above obtained accuracy % we can conclude that our model is accurate to 85.03 % which is a very good response.

Actual Vs Predicted table for test data:

	Predicted Class	
Actual	0	1
0	15	8
1	5	21

	Predicted Class	
Actual	0	1
0	30.61	16.33
1	10.20	42.86

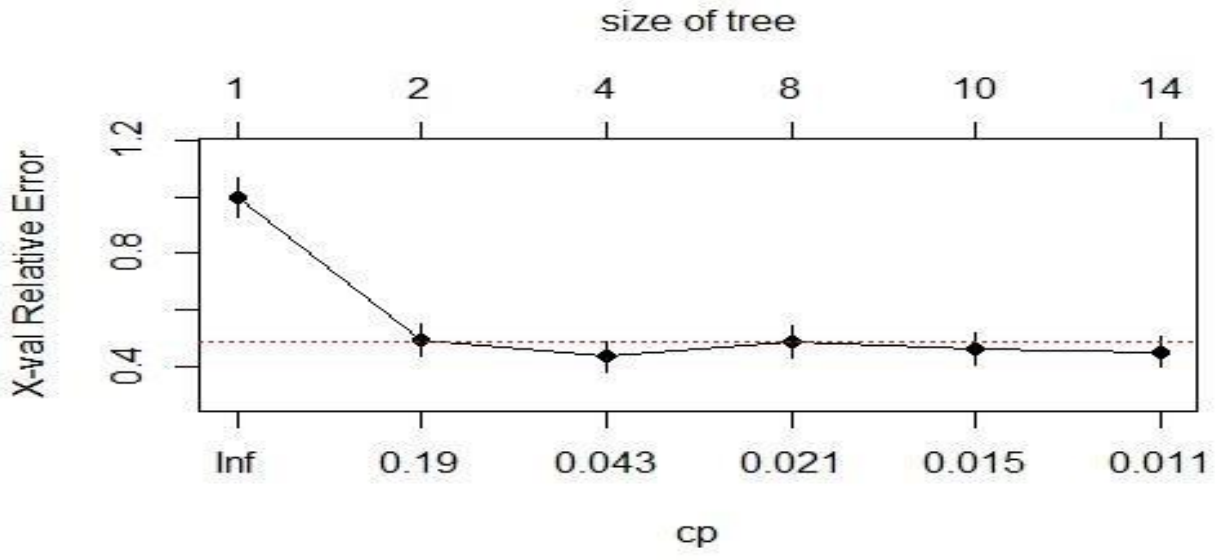
- accuracy % = (30.61 + 42.86) = 73.47 %
- misclassification % = (10.20 + 16.33) = 26.53 %

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists some deterioration between the training and test data. Hence, we can use this model for prediction because the model is not deteriorating too much which implies that this Logistic Regression model can be generalizable.

Classification Tree:

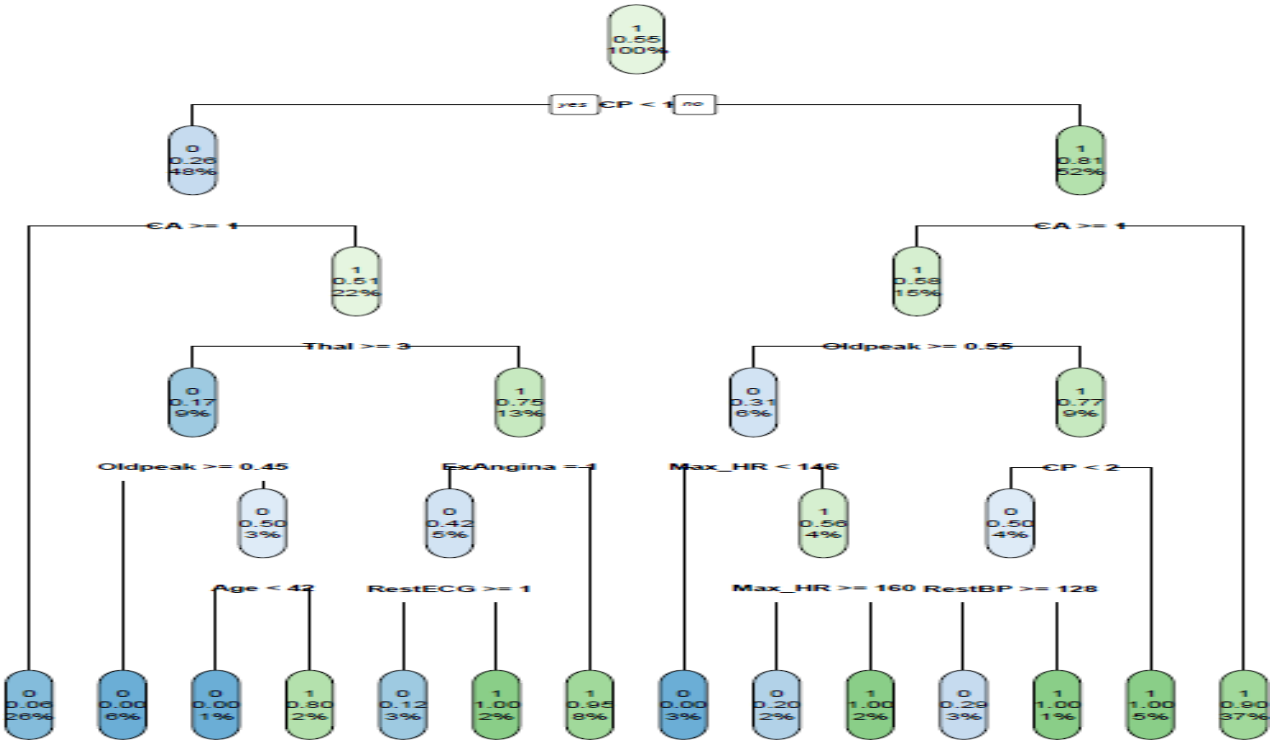
By assuming the minimum samples split to be 2, we develop a Classification Tree model

To find the optimum Cp value, we plot Cp plot



From the cp plot we observe that the Cross Validation Error is minimum at number of terminating nodes are 14 and the corresponding Cp value is 0.011.

Pruning the tree with the optimum Cp value



Actual Vs Predicted table for training data:

	Predicted Class	
Actual	0	1
0	104	11
1	8	131

	Predicted Class	
Actual	0	1
0	40.94	4.33
1	3.15	51.57

- accuracy % = (40.94 + 51.57) = 92.51%
- misclassification % = (3.15 + 4.33) = 7.48%

From the above obtained accuracy % we can conclude that the training data of Classification Tree Model is accurate to 92.51% which is a good response.

Actual Vs Predicted table for test data:

	Predicted Class	
Actual	0	1
0	15	8
1	8	18

	Predicted Class	
Actual	0	1
0	30.61	16.33
1	16.33	36.73

- accuracy % = (30.61 + 36.73) = 67.34 %
- misclassification % = (16.33 + 16.33) = 32.66 %

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists huge deterioration between the training and test data. Hence, we cannot use this model for prediction because the model is deteriorating too much which implies that this Classification Tree model can't be generalizable.

Bagging:

Fitting the Bagging model into the training data we get

Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 13

Actual Vs Predicted table for training data:

	Predicted Class	
Actual	0	1
0	86	29
1	21	118

- accuracy % = (86 + 118)/254 = 80.31%
- misclassification % = (21 + 29)/254 = 19.69%

From the above obtained accuracy % we can conclude that the training data of Bagging Model is accurate to 80.31% which is a good response.

Actual Vs Predicted table for test data:

	Predicted Class	
Actual	0	1
0	14	9
1	5	21

	Predicted Class	
Actual	0	1
0	28.57	18.37
1	10.20	42.86

- accuracy % = (28.57 + 42.86) = 71.43 %
- misclassification % = (10.20 + 18.37) = 28.57 %

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists some deterioration between the training and test data. Hence, we cannot use this model for prediction because the model is not deteriorating too much which implies that this Bagging model can be generalizable.

Random Forest:

Fitting the Random Forest model into the training data we get

Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 3

Actual Vs Predicted table for training data:

	Predicted Class	
Actual	0	1
0	88	27
1	17	122

- accuracy % = (88 + 122)/254 = 82.68%
- misclassification % = (17 + 27)/254 = 17.32%

From the above obtained accuracy % we can conclude that the training data of Random Forest Model is accurate to 82.68% which is a good response.

Actual Vs Predicted table for test data:

	Predicted Class	
Actual	0	1
0	16	7
1	5	21

	Predicted Class	
Actual	0	1
0	32.65	14.29
1	10.20	42.86

- accuracy % = (32.65 + 42.86) = 75.51 %
- misclassification % = (10.20 + 14.29) = 24.49 %

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists some deterioration between the training and test data. Hence, we can use this model for prediction because the model is not deteriorating too much which implies that this Random Forest model can be generalizable.

Naive Bayes:

Fitting the Naïve Bayes model into the training data we get the accuracy% and mis-classification% as

Actual Vs Predicted table for training data:

	Predicted Class	
Actual	0	1
0	93	22
1	15	124

	Predicted Class	
Actual	0	1
0	36.61	8.66
1	5.91	48.82

- accuracy % = (36.61 + 48.82) = 85.43%
- misclassification % = (8.66 + 5.91) = 14.57%

From the above obtained accuracy % we can conclude that the training data of Naïve Bayes model is accurate to 85.43% which is a very good response.

Actual Vs Predicted table for test data of Naive Bayes model:

	Predicted Class	
Actual	0	1
0	17	6
1	6	20

	Predicted Class	
Actual	0	1
0	34.69	12.24
1	12.24	40.82

- accuracy % = (34.69 + 40.82) = 75.51%
- misclassification % = (12.24 + 12.24) = 24.49 %

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists some deterioration between the training and test data. Hence, we can use this model for prediction because the model is not deteriorating too much which implies that this Naive Bayes model can be generalizable.

K Nearest Neighbors:

By developing the KNN model using training data, the obtained optimum value of k is 11.

Actual Vs Predicted table for training data for KNN Model:

	Predicted Class	
Actual	0	1
0	78	37
1	27	112

	Predicted Class	
Actual	0	1
0	30.71	14.57
1	10.63	44.09

- accuracy % = (30.71 + 44.09) = 74.80 %
- misclassification % = (10.63 + 14.57) = 25.20 %

From the above obtained accuracy % we can conclude that the training data of KNN Model is accurate to 74.80% which is a good response.

Actual Vs Predicted table for test data of KNN Model:

	Predicted Class	
Actual	0	1
0	14	9
1	10	16

	Predicted Class	
Actual	0	1
0	28.57	18.37
1	20.41	32.65

- accuracy % = (28.57 + 32.65) = 61.22 %
- misclassification % = (18.37 + 20.41) = 38.78 %

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists some deterioration between the training and test data. Hence, we can use this model for prediction because the model is not deteriorating too much which implies that this KNN model can be generalizable.

Support Vector Machine:

Parameter values of **linear kernel**

Parameters:
SVM-Type: C-classification
SVM-Kernel: linear
cost: 1

Number of Support Vectors: 97
(48 49)
Number of Classes: 2
Levels:
0 1

Parameter values of **polynomial kernel**

Parameters:
SVM-Type: C-classification
SVM-Kernel: polynomial
cost: 5
degree: 3
coef.0: 0

Number of Support Vectors: 130
(69 61)
Number of Classes: 2
Levels:
0 1

Parameter values of **radial kernel**

Parameters:
SVM-Type: C-classification
SVM-Kernel: radial
cost: 5

Number of Support Vectors: 252
(137 115)
Number of Classes: 2
Levels:
0 1

Actual Vs Predicted table for training data of SVM Model:

Linear kernel

	Predicted Class	
Actual	0	1
0	89	26
1	11	128

	Predicted Class	
Actual	0	1
0	35.04	10.24
1	4.33	50.39

- accuracy % = (35.04 + 50.39) = 85.43%
- misclassification % = (4.33 + 10.24) = 14.57%

Polynomial kernel

	Predicted Class	
Actual	0	1
0	109	6
1	2	137

	Predicted Class	
Actual	0	1
0	42.91	2.36
1	0.79	53.94

- accuracy % = (42.91 + 53.94) = 96.85%
- misclassification % = (0.79 + 2.36) = 3.15%

Radial kernel

	Predicted Class	
Actual	0	1
0	115	0
1	0	139

	Predicted Class	
Actual	0	1
0	45.28	0
1	0	54.72

- accuracy % = (45.28 + 54.72) = 100%
- misclassification % = 0%

	Accuracy %	**Misclassification %**
Linear	85.3	14.57
Polynomial	96.85	3.15
Radial	100	0

From the obtained values of accuracy% & misclassification% of training data, radial kernel is giving much better performance (very high accuracy) than linear kernel and polynomial kernel. In this case, we go with radial kernel which gives the best model. Therefore, we conclude that the radial kernel of SVM model is accurate to 100%.

Actual Vs Predicted table for test data of SVM Model:

Linear kernel

	Predicted Class	
Actual	0	1
0	15	8
1	4	22

	Predicted Class	
Actual	0	1
0	30.61	16.33
1	8.16	44.90

- accuracy % = (30.61 + 44.90) = 75.51%
- misclassification % = (8.16 + 16.33) = 24.49%

Polynomial kernel

	Predicted Class	
Actual	0	1
0	14	9
1	6	20

	Predicted Class	
Actual	0	1
0	28.57	18.37
1	12.24	40.82

- accuracy % = (28.57 + 40.82) = 69.39%
- misclassification % = (12.24 + 18.37) = 30.61%

Radial kernel

	Predicted Class	
Actual	0	1
0	4	19
1	0	26

	Predicted Class	
Actual	0	1
0	8.16	38.78
1	0	53.06

- accuracy % = (8.16 + 53.06) = 61.22%
- misclassification % = 38.78%

	Accuracy %	**Misclassification %**
Linear	75.51	24.49
Polynomial	69.39	30.61
Radial	61.22	38.78

From observing the Accuracy% & misclassification% for both the training data & test data we notice that there exists huge deterioration between the training and test data. Hence, we cannot use this model for prediction because the model is deteriorating too much which implies that this SVM model can't be generalizable.

Conclusion:

		Accuracy %	Misclassification%
Logistic Regression	**Training**	85.03	14.97
	Test	73.47	26.53
Classification Tree	**Training**	92.51	7.48
	Test	67.34	32.66
Bagging	**Training**	80.31	19.69
	Test	71.43	28.57
Random Forest	**Training**	82.68	17.32
	Test	75.51	24.49
Naive Bayes	**Training**	85.43	14.57
	Test	75.51	24.49
KNN	**Training**	74.80	25.20
	Test	61.22	38.78
SVM	**Training**	100	0
	Test	61.22	38.78

From observing the above obtained values of Accuracy% & Misclassification% of different Supervised Learning Models, we notice that except for Classification Tree & SVM models, all the models are having good performance since the deterioration between the training data values and test data values is less. For Classification Tree & SVM models the training data performance is very good but for test data the accuracy of the model is greatly reduced. Hence these two models can't be used for predicting to find out whether having heart disease or not as these may lead to wrong conclusions.