Heart disease Prediction with different Supervised Learning Models using Python

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**
count	303	303	303	303	303
mean	54.37	131.62	246.26	149.65	1.04
std	9.08	17.54	51.83	22.91	1.16
min	29	94	126	71	0
25%	47.50	120	211	133.50	0
50%	55	130	240	153	0.80
75%	61	140	274.50	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 242 observations/records
- test data is having 61 observations/records

Logistic Regression:

Fitting the Logistic Regression model into the training data we get the accuracy% and mis-classification% as

- accuracy % = 86.36%
- misclassification % = 13.64%

Actual Vs Predicted table for training data of Logistic Regression model:

	Predicted Class	
Actual	0	1
0	80	25
1	8	129

From the above obtained accuracy % we can conclude that the training data of Logistic Regression model is accurate to 86.36% which is a very good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 85.25%
- misclassification % = 14.75%

Actual Vs Predicted table for test data of Logistic Regression model:

	Predicted Class	
Actual	0	1
0	26	7
1	2	26

From observing the accuracy% & misclassification% for both the training data & teat data we notice that these values are very much close to each other and there exists some minor 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 25, we develop a Classification Tree model which is found to be 85.54% accurate.

After optimizing, the hyperparameter values we get the optimum parameter value to be with

```
{'criterion': 'gini', 'min_samples_split': 10}
```

The optimum model with minimum samples split of 10 & with criterion – gini is found to be accurate to

- accuracy % = 92.56 %
- misclassification % = 7.44 %

Actual Vs Predicted table for training data of Classification Tree model:

	Predicted Class	
Actual	0	1
0	93	12
1	6	131

From the above obtained accuracy % we can conclude that the training data of Classification Tree Model is accurate to 92.56% which is a very good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 77.05%
- misclassification % = 22.95%

Actual Vs Predicted table for test data of Classification Tree model:

	Predicted Class	
Actual	0	1
0	23	10
1	4	24

From observing the accuracy% & misclassification% for both the training data & teat data we notice that there exists high 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:

By assuming the minimum samples split to be 25 & n-estimators to be 500, we develop a model using Bagging method which is found to be 89.67% accurate.

After optimizing the hyperparameter values, we get the optimum parameter values to be with

```
{'criterion': 'entropy', 'min_samples_split': 10, 'n_estimators': 100}
```

The optimum model with with criterion – entropy, minimum samples split of 10 & n-estimator value of 100 is found to be accurate to

- accuracy % = 95.87 %
- misclassification % = 4.13 %

Actual Vs Predicted table for training data of Bagging model:

	Predicted Class	
Actual	0	1
0	98	7
1	3	134

From the above obtained accuracy % we can conclude that the training data of Bagging Model is accurate to 95.87% which is a very good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 85.25%
- misclassification % = 14.75%

Actual Vs Predicted table for test data of Bagging model:

	Predicted Class	
Actual	0	1
0	25	8
1	1	27

From observing the accuracy% & misclassification% for both the training data & teat data we notice that there exists high 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 Bagging model can’t be generalizable.

Random Forest:

By assuming the minimum samples split to be 25 & n-estimators to be 500, we developa Random Forest model which is found to be 89.67% accurate.

After optimizing the hyperparameter values, we get the optimum parameter values to be with

```
{'criterion': 'entropy', 'min_samples_split': 20, 'n_estimators': 200}
```

The optimum model with with criterion – entropy, minimum samples split of 10 & n-estimator value of 100 is found to be accurate to

- accuracy % = 91.32 %
- misclassification % = 8.68 %

Actual Vs Predicted table for training data of Random Forest model:

	Predicted Class	
Actual	0	1
0	89	16
1	5	132

From the above obtained accuracy % we can conclude that the training data of Rando Forest Model is accurate to 91.32% which is a very good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 86.89%
- misclassification % = 13.11%

Actual Vs Predicted table for test data of Random Forest model:

	Predicted Class	
Actual	0	1
0	26	7
1	1	27

From observing the accuracy% & misclassification% for both the training data & teat data we notice that there exists some minor 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

- accuracy % = 83.88%
- misclassification % = 16.12%

Actual Vs Predicted table for training data of Naïve Bayes model:

	Predicted Class	
Actual	0	1
0	81	24
1	15	122

From the above obtained accuracy % we can conclude that the training data of Naïve Bayes model is accurate to 83.88% which is a good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 86.89%
- misclassification % = 13.11%

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

	Predicted Class	
Actual	0	1
0	28	5
1	3	25

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that there exists some minor deterioration between the training and test data and test data is performing well than training 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 assuming the nearest neighbours to be 5, we develop a KNN model which is found to be 75.21% accurate.

After optimizing the hyperparameter values, we get the optimum parameter value to be with

```
{'n_neighbors': 15}
```

The optimum model with with n-neighbors of 15 is found to be accurate to

- accuracy % = 66.12 %
- misclassification % = 33.88 %

Actual Vs Predicted table for training data for KNN Model:

	Predicted Class	
Actual	0	1
0	50	55
1	27	110

From the above obtained accuracy % we can conclude that the training data of KNN Model is accurate to 66.12% which is a good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 67.21%
- misclassification % = 32.79%

Actual Vs Predicted table for test data of KNN Model:

	Predicted Class	
Actual	0	1
0	17	16
1	4	24

From observing the Accuracy% & misclassification% for both the training data & teat data we notice that both these values are very much close to each other and test data is performing well than training 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:

After optimizing the hyperparameter values, we get the optimum parameter values of SVM model are to be with

```
{'C': 10, 'degree': 1, 'kernel': 'linear'}
```

The optimum model with with linear kernel, degree of 1 and cost value of 10 is found to be accurate to

- accuracy % = 86.36 %
- misclassification % = 13.64 %

Actual Vs Predicted table for training data of SVM Model:

	Predicted Class	
Actual	0	1
0	82	23
1	10	127

From the above obtained accuracy % we can conclude that the training data of SVM Model is accurate to 86.36% which is a good response.

Similarly, for test data we get the accuracy% and mis-classification% as

- accuracy % = 80.33%
- misclassification % = 19.67%

Actual Vs Predicted table for test data of SVM Model:

	Predicted Class	
Actual	0	1
0	25	8
1	4	24

From observing the Accuracy% & misclassification% for both the training data & test data we notice that there exists some minor 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 SVM model can be generalizable.

Conclusion:

		Accuracy %	**Misclassification%**
Logistic Regression	**Training**	86.36	13.64
	Test	85.25	14.75
Classification Tree	**Training**	92.56	7.44
	Test	77.05	22.95
Bagging	**Training**	95.87	4.13
	Test	85.25	14.75
Random Forest	**Training**	91.32	8.68
	Test	86.89	13.11
Naive Bayes	**Training**	83.88	16.12
	Test	86.89	13.11
KNN	**Training**	66.12	33.88
	Test	67.21	32.79
SVM	**Training**	86.36	13.64
	Test	80.33	19.67

From observing the above obtained values of Accuracy% & Misclassification% of different Supervised Learning Models, we notice that except for Classification Tree & Bagging models, all the models are having good performance since the deterioration between the training data values and test data values are very less. For Classification Tree & Bagging 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.