Introduction to Data Sciences

Application of classification models to determine patient's cardiac health using RapidMiner

Name: Tathagata Mookherjee
Roll Number: M21AI619

Table of Contents

Introduction

The objective of this assignment is to apply classification models in RapidMiner in order to classify instances as one of the below, based on a dataset obtained from Kaggle.com

- Heart Disease: 1, which indicates that instance has heart disease
- Heart Disease: 0, which indicates that instance does not have heart disease

We shall be using the CRISP-DM process for this assignment.

CRISP-DM Process

Phase 1: Business objectives

1. Correctly classify instances with heart-disease as 1
2. Minimize classification of instances as false-negative (Instances having heart-disease:1 classified as 0)
3. Correctly classify instances without heart-disease as 0

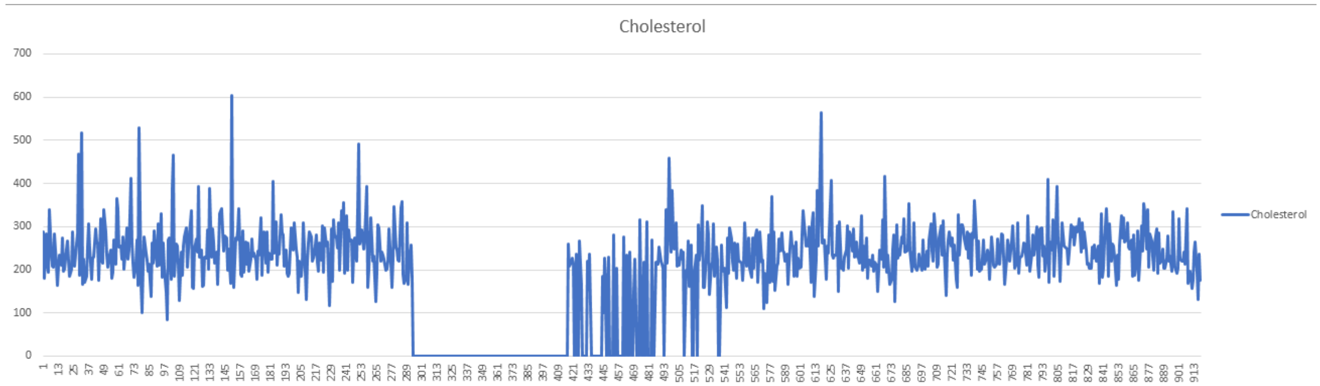
Phase 2: Data understanding

Count of data

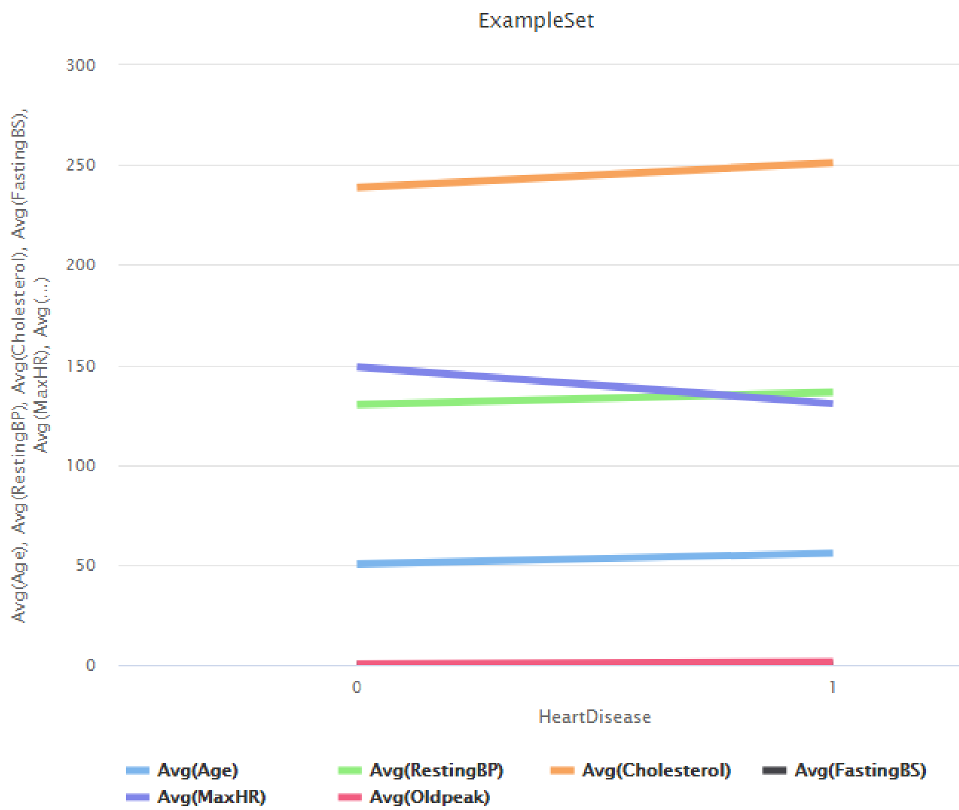
There are 918 rows of data (410 instances of heart-disease=0 and 518 instances of heart-disease=1). There seems to be a rough 60:40 split in the data. This is good as there should be enough scenarios for testing and validation.

Feature observations

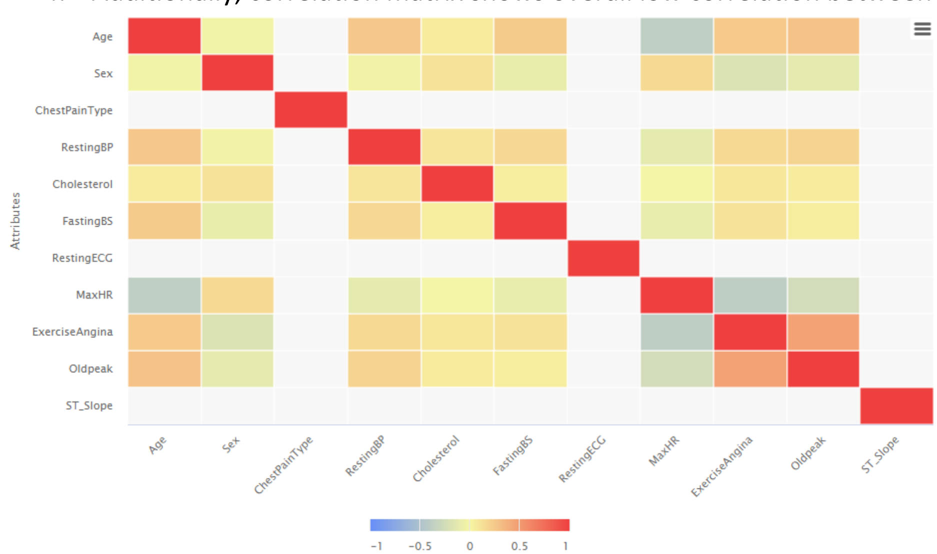
1. Cholesterol=0 values need to be filtered out as it is impossible for an instance to have 0 blood cholesterol.



2. There seem to be no missing values in the data.
3. Correlations,
    - The average Cholesterol, RestingBP and Age seems to increase for heart-disease:1
    - The average MaxHR seems to decrease for heart-disease:1



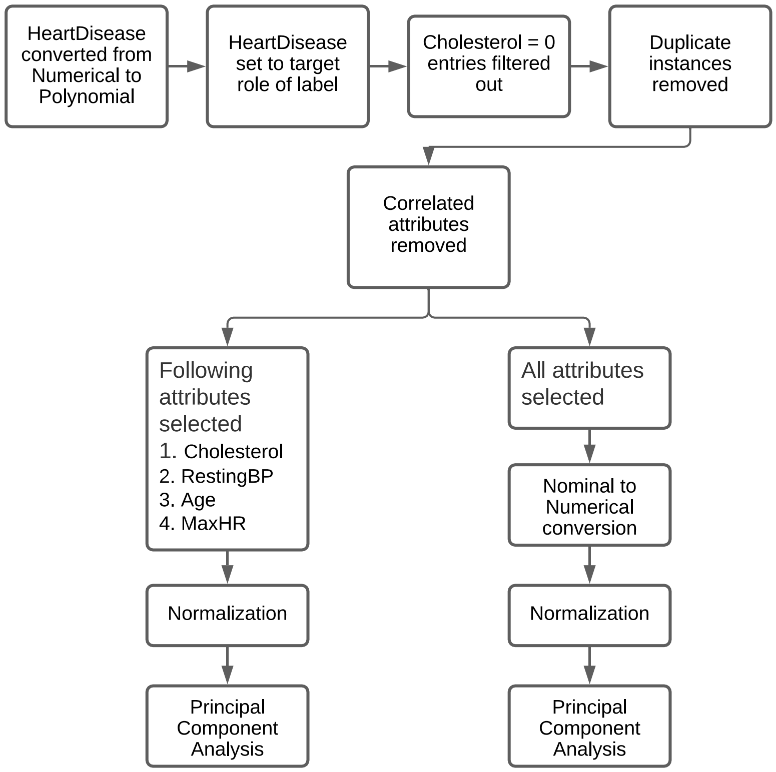
4. Additionally, correlation matrix shows overall low correlation between the features



Phase 3: Data Preparation

The following preprocessing has been applied after which the data has been split into the 2 below streams

1. Filtered data to have only Cholesterol, RestingBP, Age and MaxHR features as these seem to have the maximum change in their values (as per manual observation)
2. All features used for processing after initial cleanup.



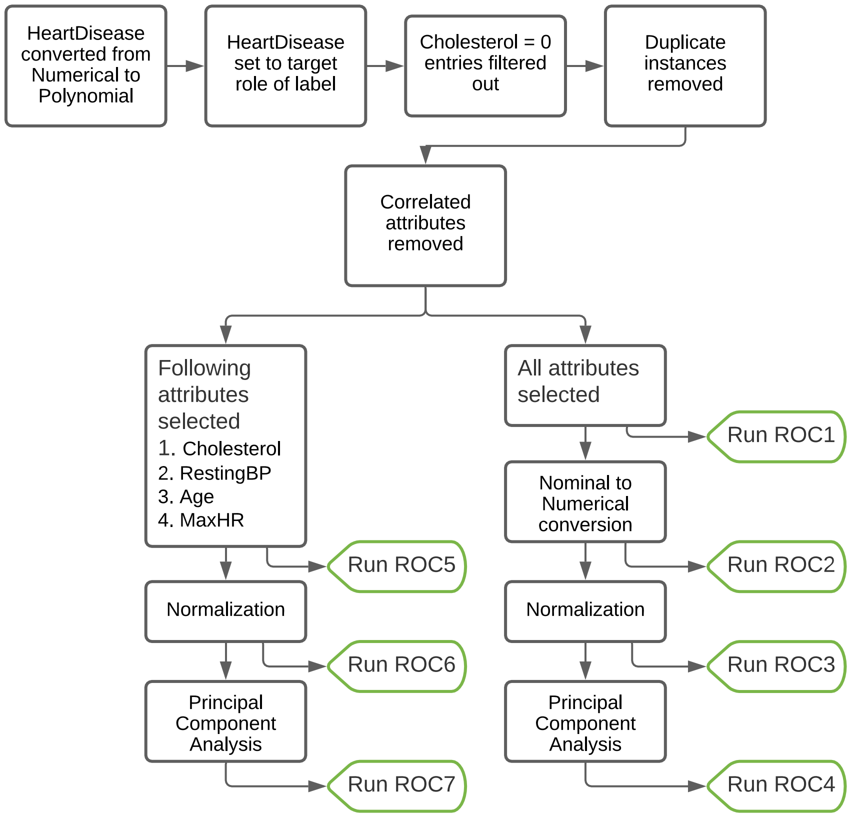
Phase 4: Modeling

Selecting models for testing

The below models shall be tested,

- Decision Tree
- Naive Bayes
- k-NN
- Random Forest
- Deep Learning
- Random Tree

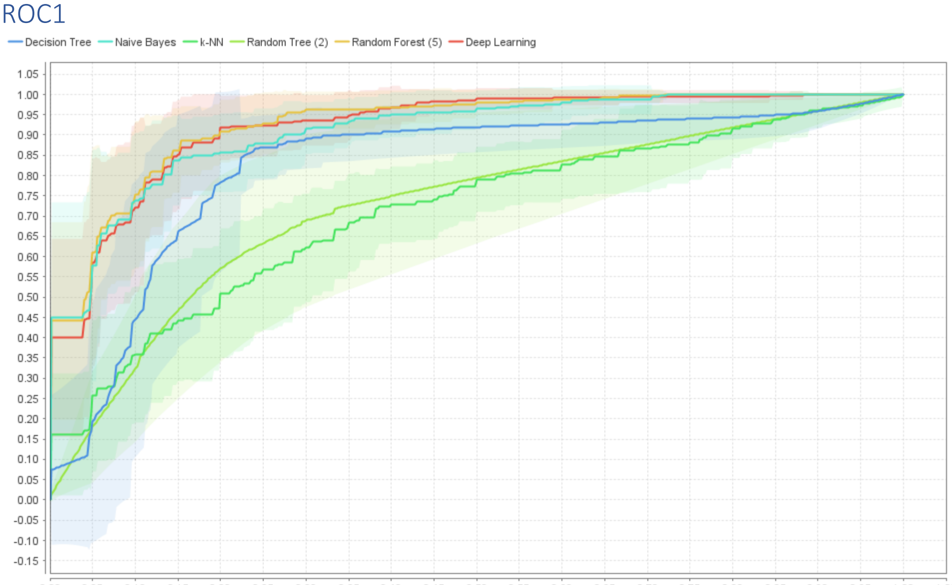
Checking ROCs



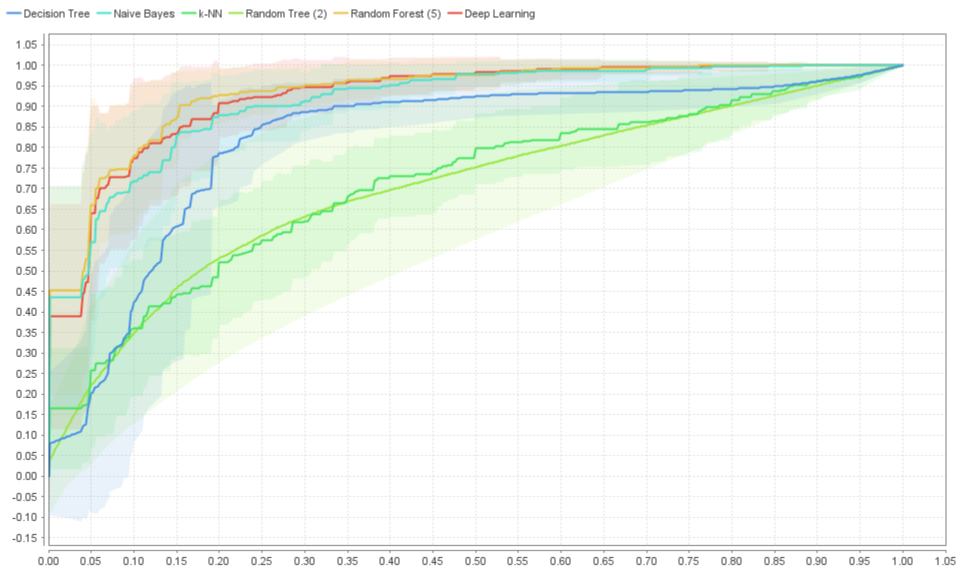
The following configuration has been applied before running ROC,

1. Pre-pruning and pruning have been enabled wherever applicable
2. K-NN model is using k=7. In fact, K-NN model has been tested with k values ranging from 1-10, and it has been observed that k=7,8 is providing the best performance. Hence k=7 has been selected and showed in the below graphs along with the other models.
3. Maximum tree depth set to 10 where applicable
4. 70% of the total data has been used in each ROC point, as 70% of the data will be used in training the right models.

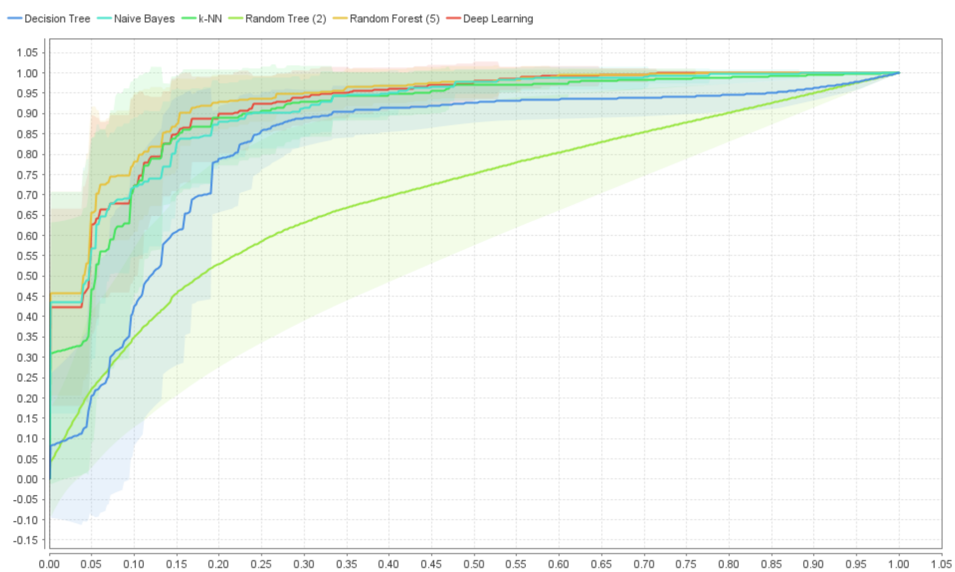
ROC1



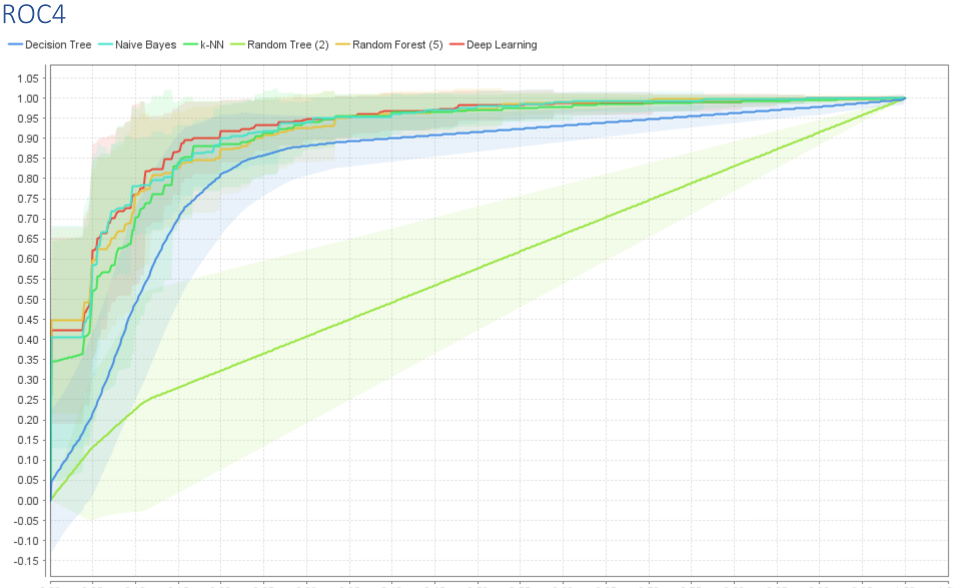
ROC2



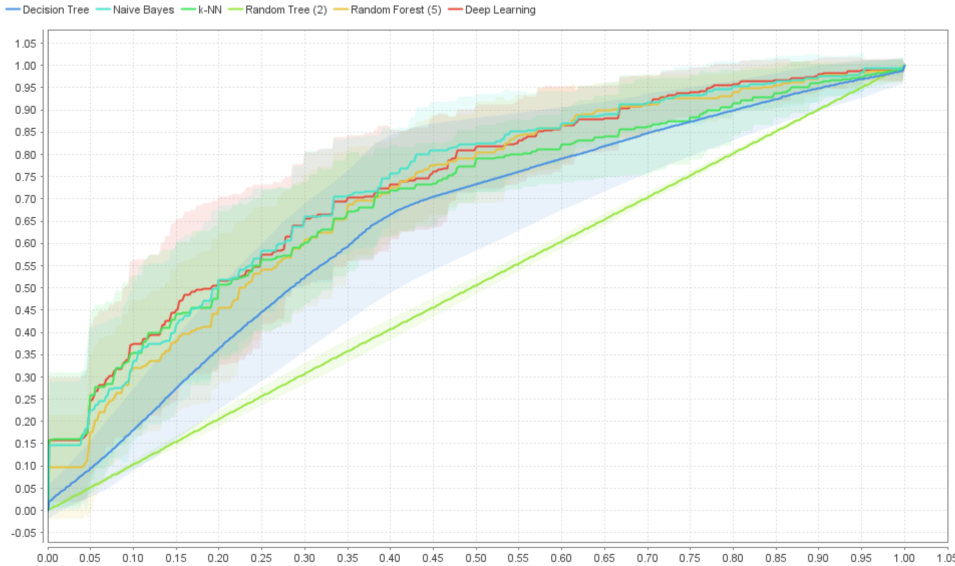
ROC3



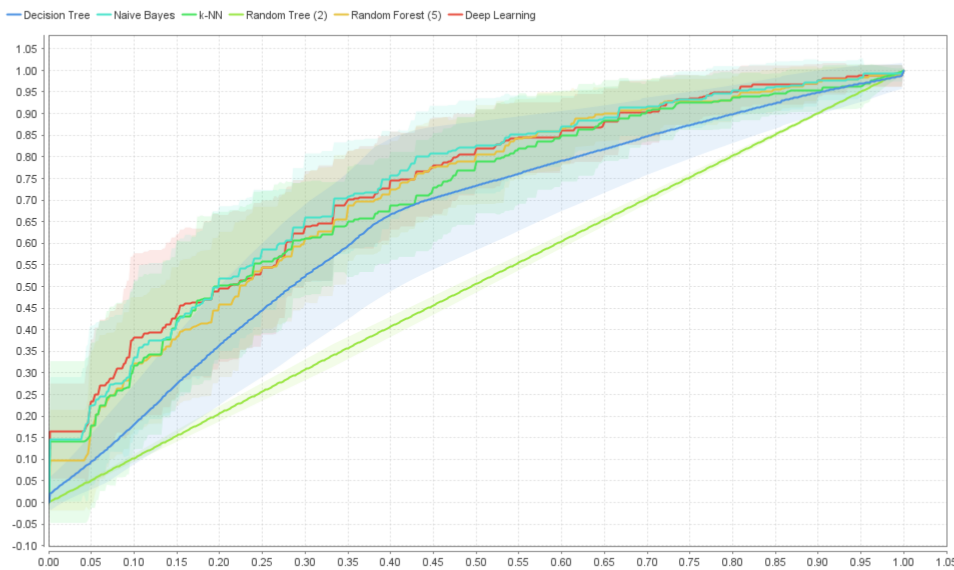
ROC4



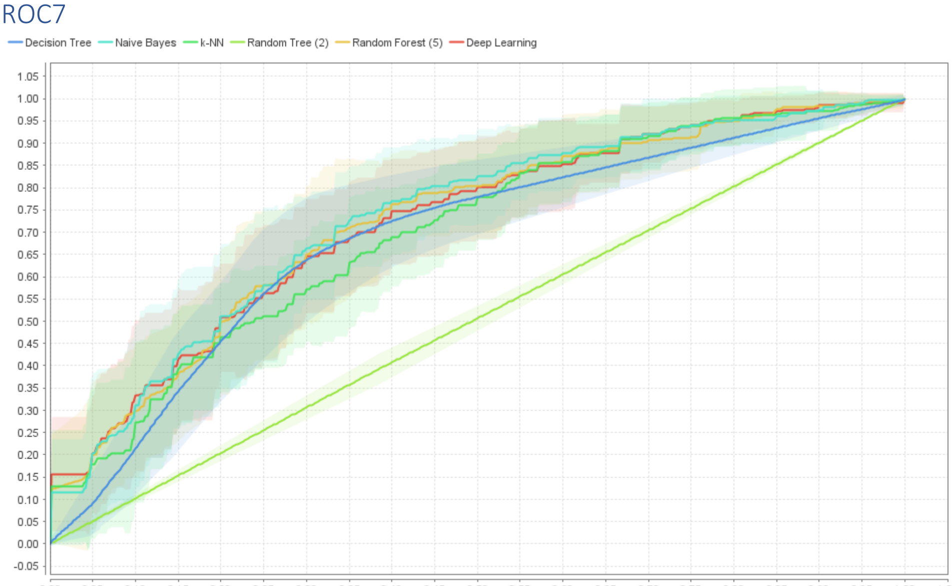
ROC5



ROC6



ROC7



From observation of the ROC curves, it is clearly evident that,

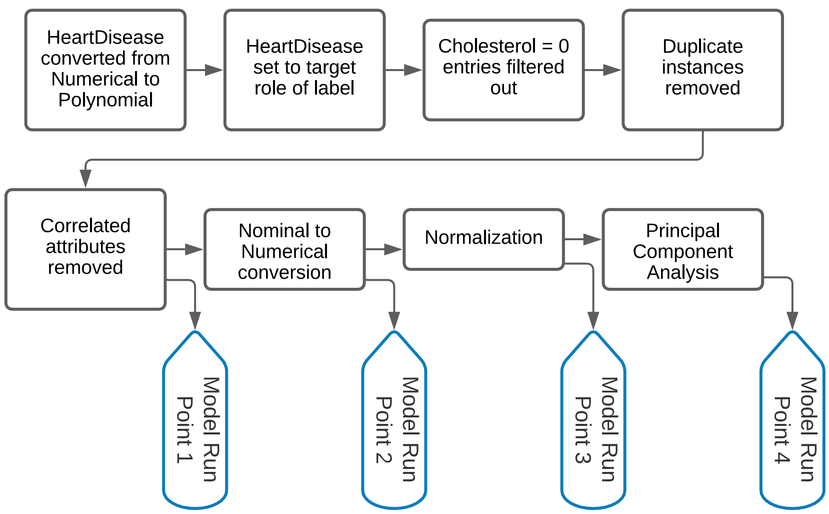
1. "All attributes selected" is the better path to follow. This the flow that contains ROC1 to ROC4.
2. Naive Bayes, Deep Learning & Random Forest models are performing better than others
3. K-NN model curve shows dramatic improvement after normalization of the data

Hence, we shall be applying all 4 models to each point where ROC1-4 was run.

Training models

The following configuration has been applied before training the models,

1. Data has been split in 7:3 ratio. 70% for training and 30% for testing.
2. Splitting process is using shuffled sampling along with a local random seed for maximum randomness
3. Pre-pruning and pruning have been enabled wherever applicable
4. K-NN model is using k=7
5. Maximum tree depth set to 10 where applicable



Below is the performance that we obtain from each run point,

SL No	Model Run Point	Model Name	Accuracy %	False Negative	True Positive	True Negative	False Positive	Spearman Rho	Kendall Tau
1	1	KNN	63.39	46	66	36	76	0.269	0.269
2	1	Deep Learning	86.16	11	101	92	20	0.726	0.726
3	1	Random Forest	86.61	13	99	95	17	0.733	0.733
4	1	Naive Bayes	85.71	12	100	92	20	0.716	0.716
5	2	KNN	63.39	46	66	76	36	0.269	0.269
6	2	Deep Learning	85.71	14	98	94	18	0.715	0.715
7	2	Random Forest	88.84	10	102	97	15	0.778	0.778
8	2	Naive Bayes	86.61	13	99	95	17	0.733	0.733
9	3	KNN	84.38	15	97	92	20	0.688	0.688
10	3	Deep Learning	84.82	17	95	95	17	0.696	0.696

11	3	Random Forest	88.84	10	102	97	15	0.778	0.778
12	3	Naive Bayes	86.61	13	99	95	17	0.733	0.733
13	4	KNN	84.82	15	97	93	19	0.697	0.697
14	4	Deep Learning	83.93	13	99	89	23	0.681	0.681
15	4	Random Forest	83.93	12	100	88	24	0.683	0.683
16	4	Naive Bayes	86.61	12	100	94	18	0.733	0.733

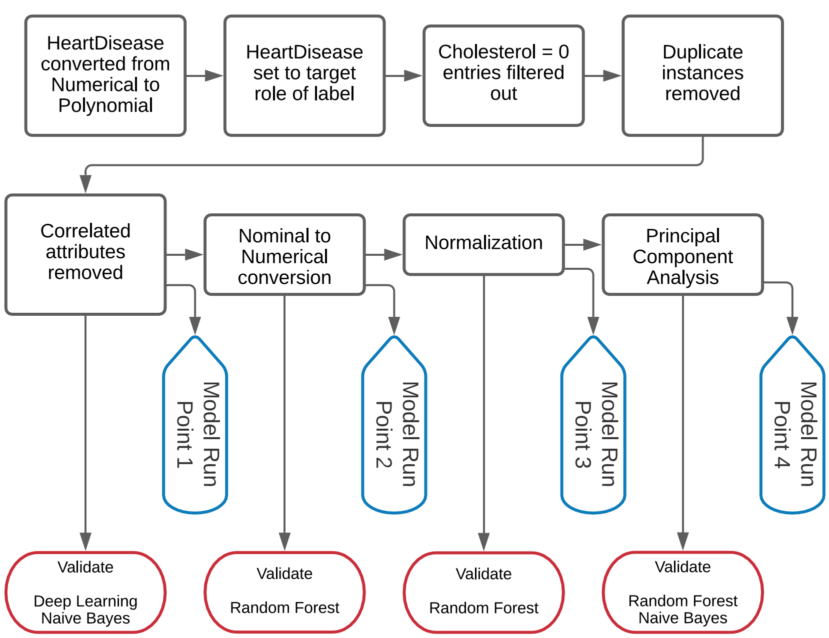
Looking back to our business objectives in phase-1, we can see that the below models in their respective run points provide the best adherence to our objectives. Hence these models will be used in our next validation phase.

SL No	Model Run Point	Model Name	Accuracy %	False Negative	True Positive	True Negative	False Positive	Spearman Rho	Kendall Tau
7	2	Random Forest	88.84	10	102	97	15	0.778	0.778
11	3	Random Forest	88.84	10	102	97	15	0.778	0.778
2	1	Deep Learning	86.16	11	101	92	20	0.726	0.726
16	4	Naive Bayes	86.61	12	100	94	18	0.733	0.733
4	1	Naive Bayes	85.71	12	100	92	20	0.716	0.716
15	4	Random Forest	83.93	12	100	88	24	0.683	0.683

Phase 5: Evaluation

The following configuration has been applied before validating the models,

1. Cross-validation has been used with 100% of the dataset.
2. The validation process is using shuffled sampling along with a local random seed for maximum randomness
3. Pre-pruning and pruning have been enabled wherever applicable
4. K-NN model is using k=7
5. Maximum tree depth set to 10 where applicable



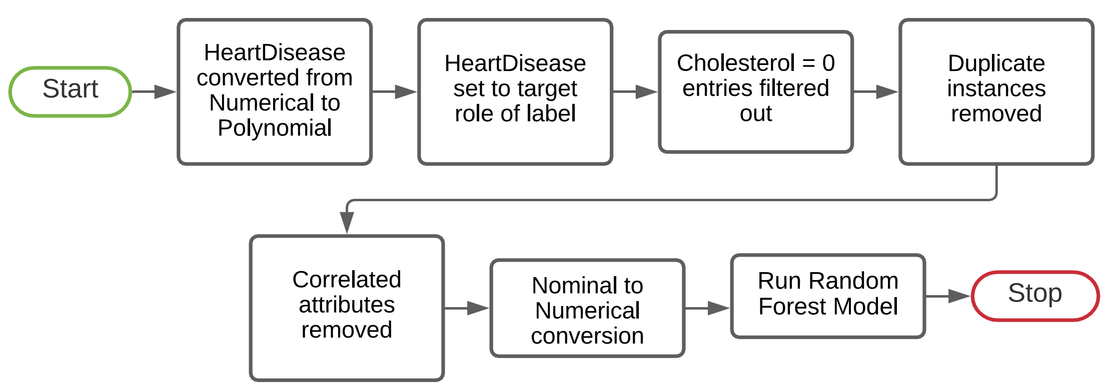
Below is the performance that we obtain from each validation point,

SL No	Validation Point	Model Name	Accuracy %	Accuracy % Standard Deviation	Max Accuracy %	Min Accuracy %	False Negative	True Positive	True Negative	False Positive	Spearman Rho	Kendall Tau
7	2	Random Forest	86.72	3.61	90.33	83.11	38	318	329	61	0.735 +/- 0.070	0.735 +/- 0.070
11	3	Random Forest	86.72	3.61	90.33	83.11	38	318	329	61	0.735 +/- 0.070	0.735 +/- 0.070
2	1	Deep Learning	84.44	5.12	89.56	79.32	49	307	323	67	0.696 +/- 0.092	0.696 +/- 0.092
16	4	Naive Bayes	85.38	3.45	88.83	81.93	52	304	333	57	0.705 +/- 0.068	0.705 +/- 0.068
15	4	Random Forest	84.31	3.86	88.17	80.45	56	300	329	61	0.685 +/- 0.078	0.685 +/- 0.078
4	1	Naive Bayes	84.18	3.67	87.85	80.51	59	297	331	59	0.681 +/- 0.074	0.681 +/- 0.074

Again, looking back at the business objectives in phase-1 can see that the best model to be applied to this scenario is SLNo=7 (Random Forest at Validation Point=2) because,

1. Highest maximum possible accuracy at 90.33%
2. Highest minimum possible accuracy at 83.11%
3. Highest count for "Correctly classify instances with heart-disease as 1" at 318
4. Lowest count for "Minimize classification of instances as false-negative" at 38
5. Highest count for "Correctly classify instances without heart-disease as 0" at 329
6. Is giving the same accuracy as SLNo=11 without the additional Normalization step. Hence this is more optimized from a performance perspective.

In conclusion we can see that the best model to be applied will follow the below flow,



Phase 6: Deployment

Out of scope of this assignment