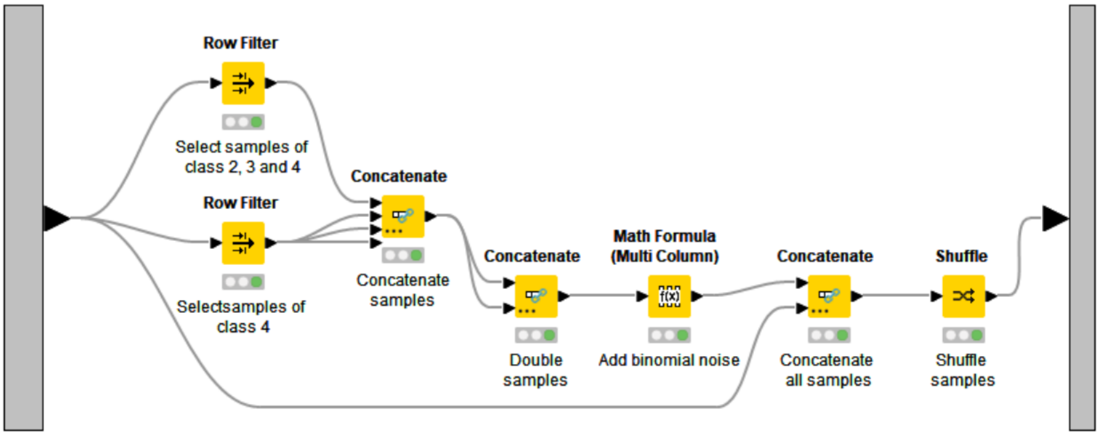
Module Code: CSMDS
Assignment report Title: Building and Testing models for Heart Disease Dataset
Student Number: 32824514
Actual hours spent on the assignment: 35
Which Artificial Intelligence tools used: None

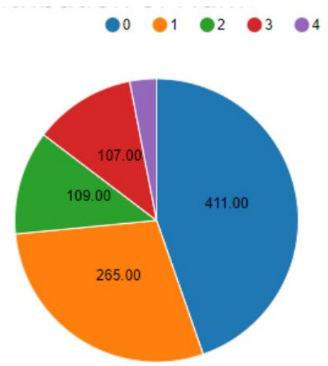
1. Discussion on Pre-processing

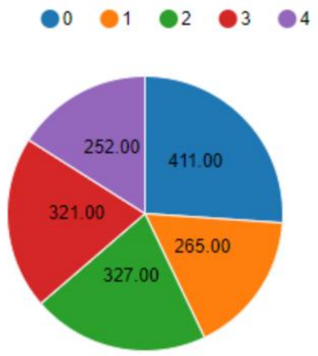
1.1 Summary

During the pre-processing, data was cleaned and normalized just like previous work. One thing worth noting is that the numbers of different types vary greatly, which will affect the model performance [1]. we increased some samples (Graph 1) and added binomial distribution noise to balance the uneven distribution (Graph 2). The Samples of Num 4 increased 9 times from 28 to 252. The samples of Num 2 and 3 increased triples. Finally, the numbers of different categories are very close (Graph 3). In this way, the cases of serious samples can be learned and predicted better by models.



Graph 1

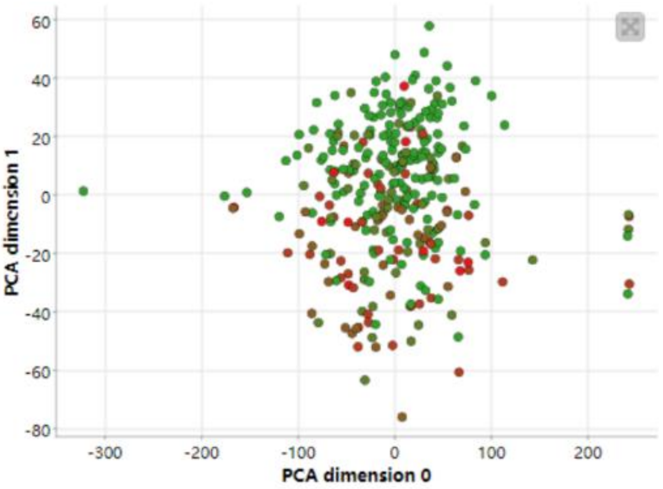




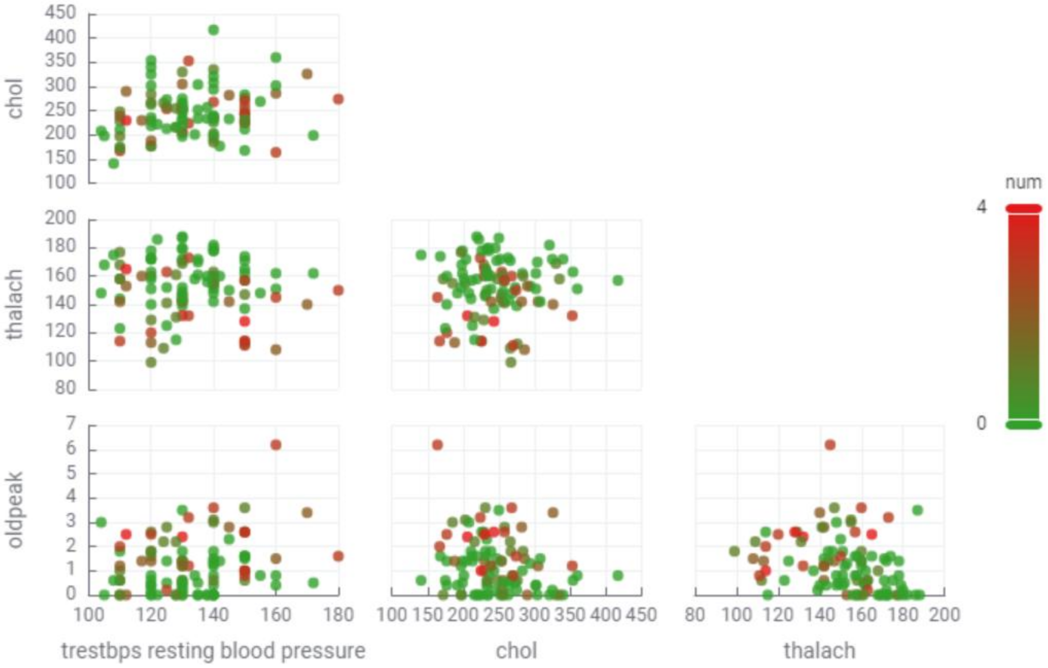
Graph 2 Graph 3

2. Discussions of Classification Solutions

UCI heart disease dataset is well organized and labeled by experts from healthy and different degrees of heart disease, which is goal of this job. As for attributes, they have age, sex, chol, thalach, etc., which are very different attributes. There is no obvious correlation or polynomial relationship between them. Additionally, the relevance matrices of two main PCA attributes (Graph 4) and chol, thalach, oldpeak, trestbps resting blood pressure attributes (Graph 5) show that there are not clear boundaries between classes. In summary, it is a classic supervised machine learning problem. There are various successful classificational algorithms addressed for this problem, such as rule-based algorithms and stochastic algorithms. This work uses different algorithms to build and to train models for prediction and then compares their accuracy and F-measure based on cross-validation methods.



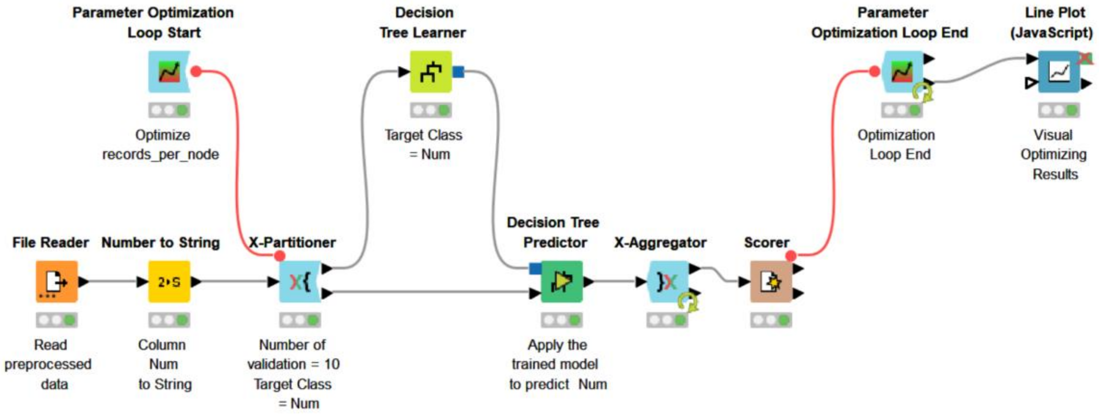
Graph 4



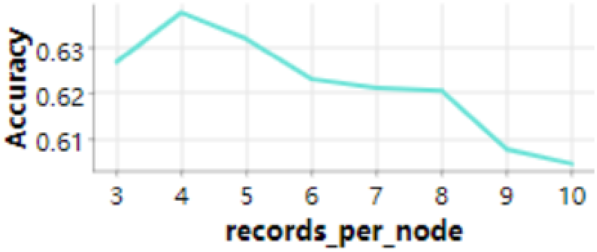
Graph 5

2.1 Decision Tree

Decision Tree (DT) algorithms are simple, efficient and easy to interpret and explain. They can easily handle complex relationships of attributes by splitting more nodes. There are some popular quality measures of node impurity such as Gini index and Gain ratio. Gini index for a given node t is $GINI(t) = 1 - \sum_j [p(j|t)]^2$(NOTE: $p(j|t)$ is the relative frequency of class j at node t). When a node p is split into k partitions, the quality of split computed as $GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$ (n_i= number of records at node i, n=number of records at node p). By using GINI index measure, we can get a model that records are equally distributed among all classes. Entropy measures are like the GINI index, which also measures the homogeneity of a node. Gain Ratio is such an entropy measure that it aims to have fewer small partitions. At this model, Gain Ratio is chosen because it prevents the model from focusing too much on details. The pruning method was applied to keep the decision tree fit to the main data. The pipeline (Graph 6) used a parameter optimization loop to select the value of variable Records-per-Node as it varies for different dataset and samples. As the results shown (Graph 7), the best value of Records-per-Node is 4 with an accuracy of about 63.5%.



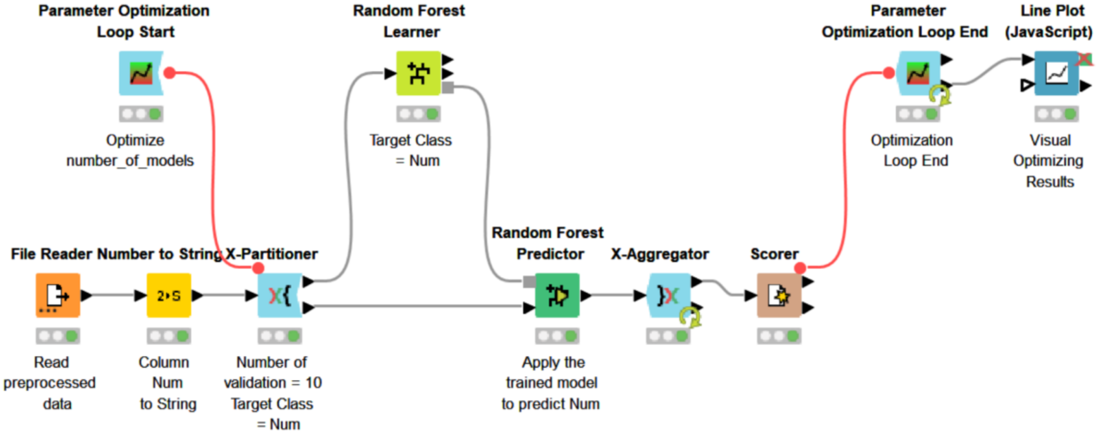
Graph 6



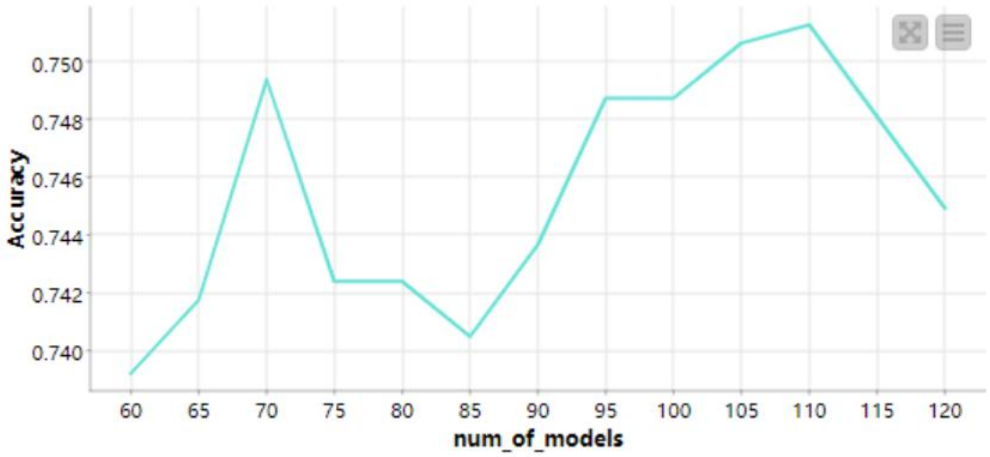
Graph 7

2.2 Random Forest

Random Forest (RF) algorithm is an ensemble of several decision trees and returns the result of the majority over all trees [2]. Each tree has its own random samples and a subset of all attributes. Through training, the best decision trees with a suitable split structure are selected and used. Normally, RF's accuracy is better than Decision Tree and slightly better than SVMs. This pipeline (Graph 8) optimized the number of models by a loop of the iteration from 60 to 120 and got the highest accuracy of 75.1% at number of models = 110 (Graph 9).



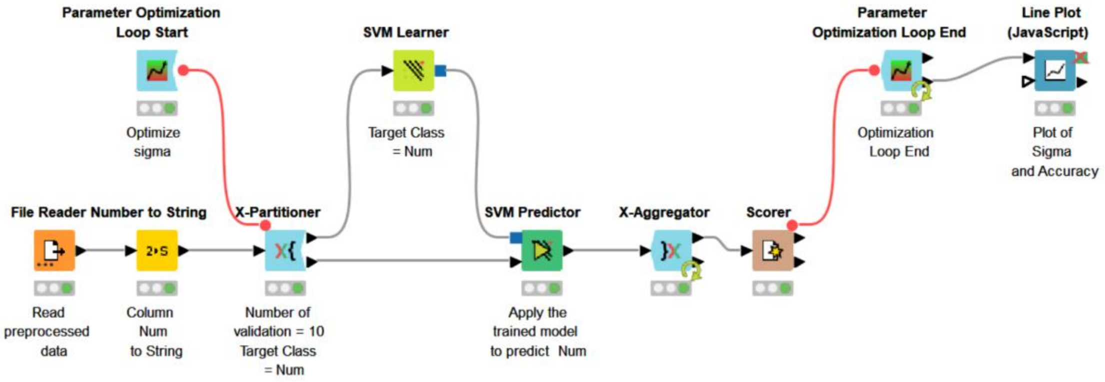
Graph 8



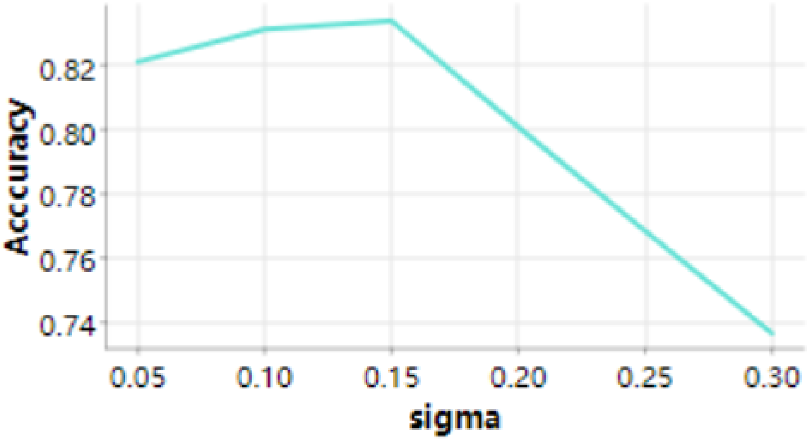
Graph 9

2.3 Support Vector Machine

Support Vector Machine (SVM) is a classification method, based on maximum margin discriminants. The target is to find suitable hyperplanes that separate the classes with the maximum margin. There are two main advantages of using SVM for classification. First, the SVM provides a geometric way to deal with supervised machine learning. Second, the optimization problem for SVM does not admit an analytic solution so that we need resort to a variety of optimization tools [3]. By using kernel measurements, we can extend SVM to support high-dimensional nonlinear space. Further, the kernel trick allows us to carry out flexible operations via the kernel function. There are 3 types of SVMs kernel in KNIME such as Polynomial, Hyper Tangent and Radial Basis Function (RBF). Polynomial Kernel is suitable for datasets where the relationship between attributes is polynomial. Hyper Tangent Kernel is useful for non-linear datasets, but its performance can be highly dependent on the choice of hyperparameters. RBF works well for non-linear classification where the corresponding feature space is high dimensional. For this dataset, attributes such as sex and age do not have a polynomial relationship. So, RBF kernel is more suitable for this dataset. This pipeline (Graph 10) optimized the number of models by a loop of the iteration from 0.05 to 0.3 and got the highest accuracy of 83.4% at sigma = 0.15 (Graph 11).



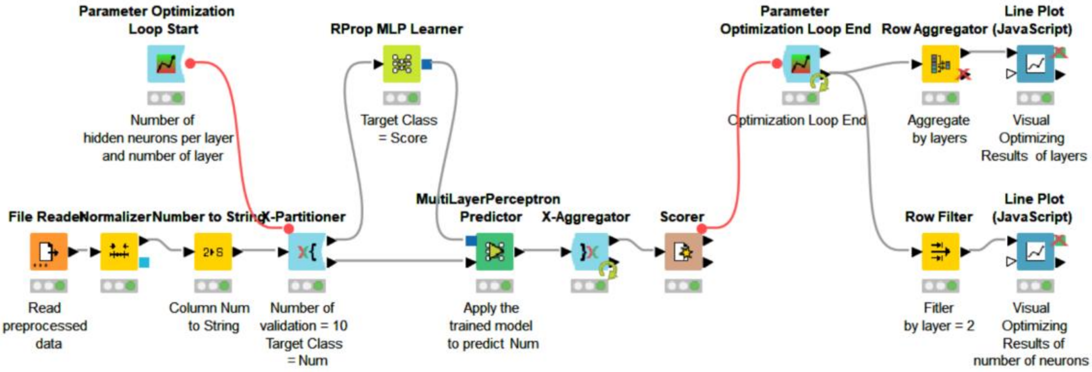
Graph 10



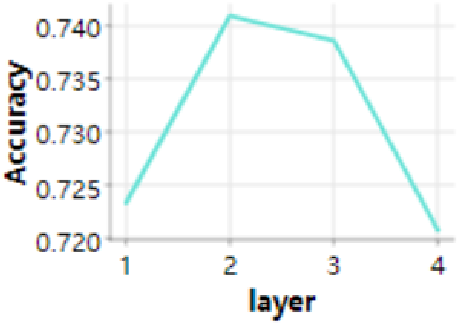
Graph 11

2.4 Neural Network

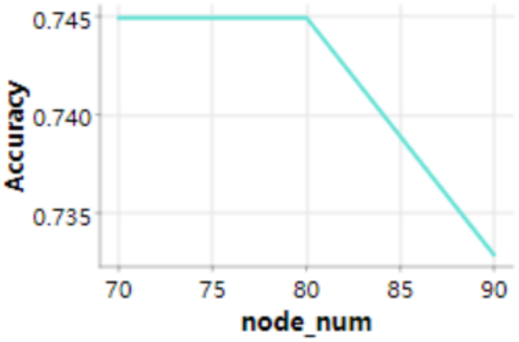
Neural Network (NN) is inspired by biological neuronal networks [5]. It is composed of abstract neural nodes that try to mimic real neurons by layers. It can be seen as a weighted directed graph with neural nodes and weighted edges. A multilayer perception (MLP) is one of the most popular Neural Network that has many neural layers. The inputs (attributes) construct the input layer, and the outputs (classifications) comprise the output layer. The neural nodes could be in one or more intermediate hidden layers. More neural nodes and layers mean the models can simulate more complex Boolean relationships. This technique is efficient for problems where the relationships between attributes and classification may be non-linear or very dynamic. This following pipeline (Graph 12) optimized the number of layers and neural nodes per layer by a loop and got the highest accuracy of 74.5% at layer = 2 (Graph 13) and sigma = 0.15 (Graph 14).


Parameter Optimization Loop Start
Number of hidden neurons per layer and number of layer
RProp MLP Learner
Target Class = Score
Parameter Optimization Loop End
Optimization Loop End
Row Aggregator
Aggregate by layers
Line Plot (JavaScript)
Visual Optimizing Results of layers
File Reader
Read preprocessed data
Normalizer
Number to String
Column Num to String
X-Partitioner
Number of validation = 10 Target Class = Num
MultiLayerPerceptron Predictor
Apply the trained model to predict Num
X-Aggregator
Scorer
Row Filter
Fitler by layer = 2
Line Plot (JavaScript)
Visual Optimizing Results of number of neurons

Graph 12



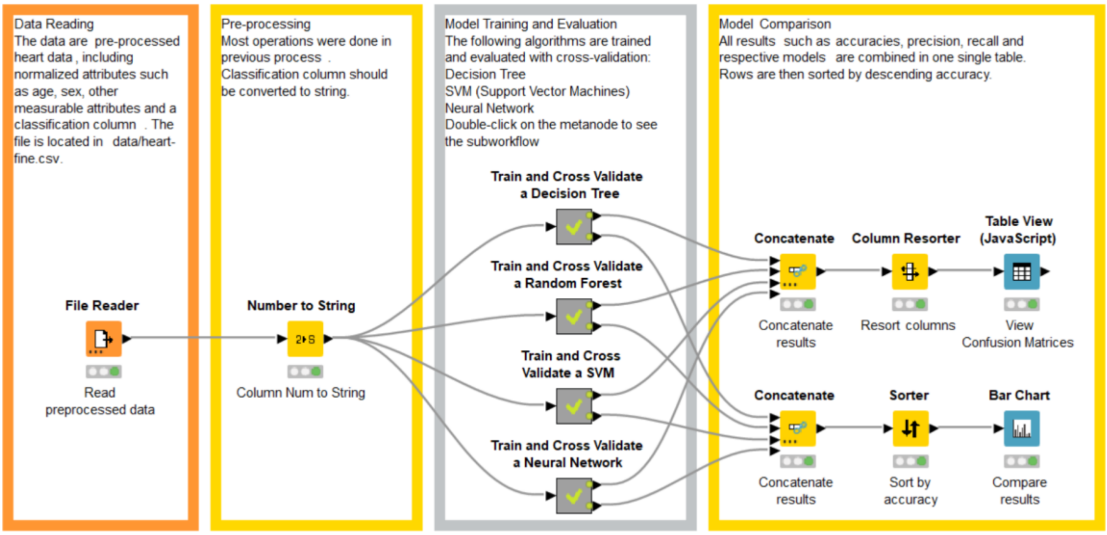
Graph 13



Graph 14

3. Model Evaluation, Results and Discussions

The following graph (Graph 15) shows the main pipeline of the project (for more details please see KNIME file main). Most pre-processing operations were done at preprocess pipeline. After reading the preprocessed data, a step is applied to convert categorical column to strings for predictions. Then data is put into 4 models to train and to predict. The output results of models then will be collected into confusion matrix and statistics matrix. Finally, the comparing results were shown in charts.



Data Reading
The data are pre-processed heart data , including normalized attributes such as age, sex, other measurable attributes and a classification column . The file is located in data/heart-fine.csv.

Pre-processing
Most operations were done in previous process . Classification column should be converted to string.

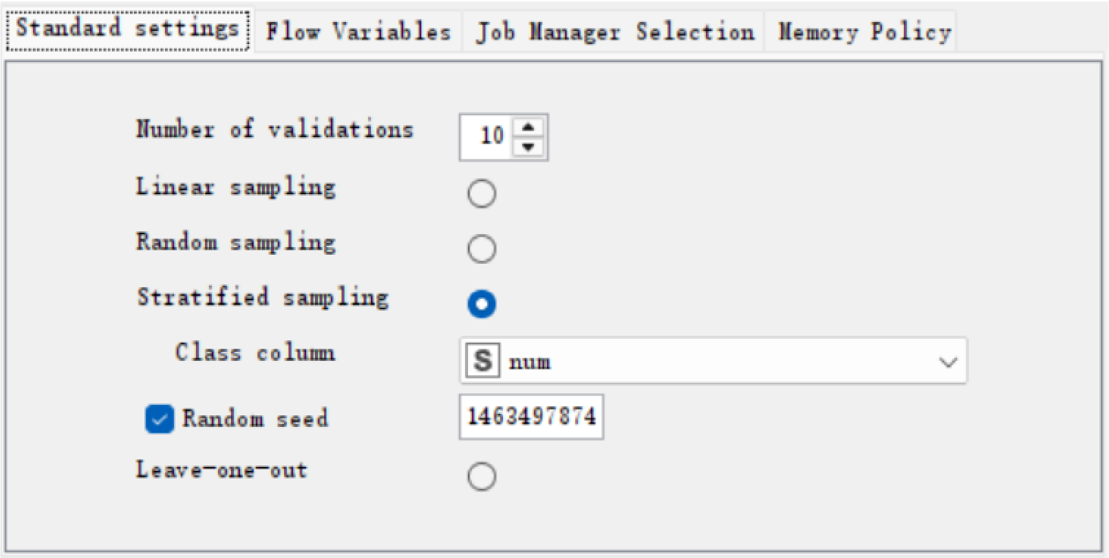
Model Training and Evaluation
The following algorithms are trained and evaluated with cross-validation:
Decision Tree
SVM (Support Vector Machines)
Neural Network
Double-click on the metanode to see the subworkflow

Model Comparison
All results such as accuracies, precision, recall and respective models are combined in one single table. Rows are then sorted by descending accuracy.

Graph 15

3.1 Model Validation

There are several methods for model validation. The re-substitution method uses all the data to fit and evaluate a model. But it may cause heavy overfitting and low generalization. The Holdout method split the data into a training set and a testing set. It is great at generalization. But considering the small number of samples, there is not enough data for training. Cross validation is one of the popular model validation techniques for assessing how the results can be generalized for an independent dataset. It randomly shuffles the dataset and splits the samples into groups. For each training, it takes one group as a testing set and the rest groups for training set. Cross validation gets less bias but takes more time. As the dataset is rather small, this method is suitable. The following graph (Graph 16) shows the configuration of cross validation X-Partitioner node in KNIME.



Number of validations 10

Linear sampling ○

Random sampling ○

Stratified sampling ●

Class column S num

☑ Random seed 1463497874

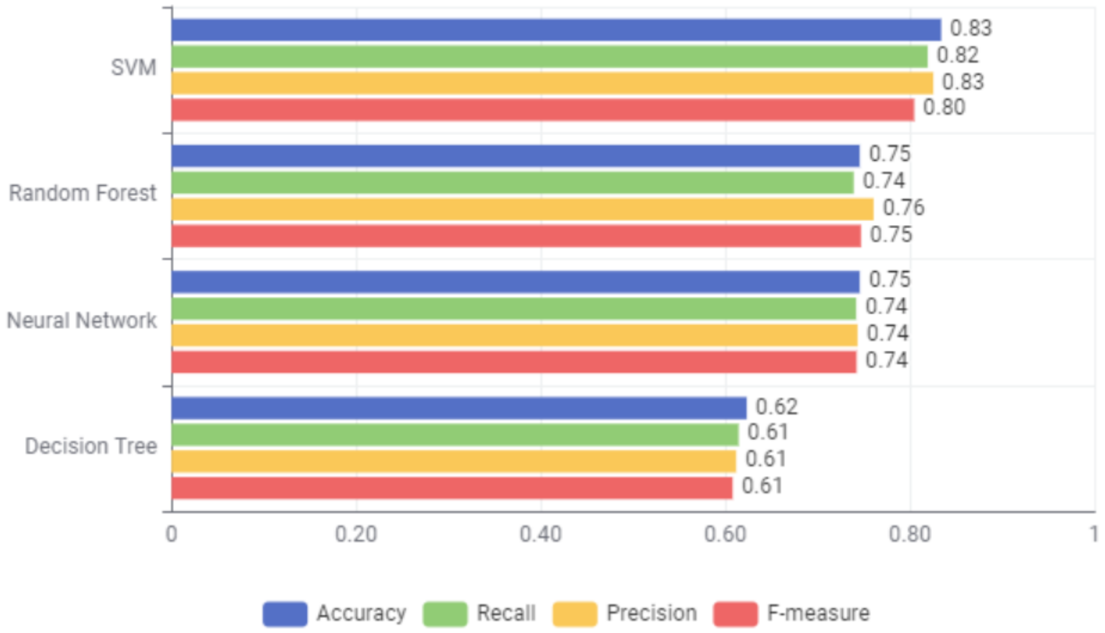
Leave-one-out ○

Graph 16

3.2 Model Evaluation, Results and Discussions

This project uses accuracy and F-measure assessing these classifying models. The accuracy of a classifier is the fraction of correct prediction over the testing set: $acc = \frac{1}{n}\sum_{i=1}^{n} I(y_i = \hat{y}_i)$. The precision of a classifier for class c_i is the fraction of correct predictions over all points predicted to be in class c_i: $prec_i = \frac{n_{ii}}{m_i}$ where m_i is the number of examples predicted as c_i. The recall of a classifier for class c_i is the fraction of correct predictions over all points in class c_i: $recall_i = \frac{n_{ii}}{n_i}$ where n_i is the number of points in class c_i. F-measure is a balance between precision and recall values [5], by computing their harmonic means for class c_i: $F_i = \frac{2}{\frac{1}{prec_i}+\frac{1}{recall_i}}$. The overall F-measure for a classifier is the mean of the class-specific values. It is better to hold high precision and recall. For a great classifier, the best value of F-measure is 1.

We see that SVM algorithm performed better than others (Graph 17), with 83% accuracy and 0.80 F-measure score. This might be because for this dataset, the distributions of samples in different classes mixed up together and the reasons for heart diseases are more complicated than expected and SVM, with an RBF kernel, has more flexibility to fit the classification by non-linear decision boundaries. Random Forest and Neural Network perform equally well, with 75% accuracy and about 0.75 F-measure score. Decision Tree is worse than others.



Graph 17

In more detail, the confusion matrices (Table 1) show a more comprehensive picture. Intuitively, greater accuracy is better. But, for this heart disease model, predicting a patient with heart disease to be healthy is much worse than predicting a healthy person to have heart disease. In other words, downplaying the severity of the disease will result in the disease not being treated promptly. So, it is meaningful to calculate the cost of the predictions by different weights. In terms of classification results, 0 means healthy, 1 to 4 means the severity of heart disease. Thus, the costs of predicting a patient to be healthy should be 100 and the costs of predicting severe illness to be less severe should be 10 (Table 2). To calculate the costs of different models, a different analysis result was obtained (Table 3). Though SVM has the highest accuracy and F-measure, the cost is higher than Neural Network (of 12068). For most classes of the predictions, SVM performed great, except that it tends to predict severity 1 as healthy (122 wrong predictions). In contrast, Neural Network, with lower accuracy and F-measure, had the minimum cost of 10641. Thus, Neural Network is better than other models for this heart disease dataset.

Model	Actual Class	0	1	2	3	4
Decision Tree	0	351	43	7	9	1
	1	102	115	25	20	3
	2	24	55	154	68	26
	3	33	48	67	145	28
	4	8	9	8	10	217
Random Forest	0	345	48	10	8	0
	1	99	135	18	12	1
	2	20	46	237	23	1
	3	17	51	21	230	2
	4	5	10	5	4	228
SVM	0	358	29	10	12	2

	1	122	76	26	31	10
	2	3	5	315	4	0
	3	3	3	2	313	0
	4	0	0	0	0	252
Neural Network	0	309	64	20	15	3
	1	71	126	32	29	7
	2	14	23	268	17	5
	3	14	31	23	248	5
	4	4	11	8	5	224

Table 1

Cost Matrix		Predicted Class				
		0	1	2	3	4
Actual Class	0	0	1	1	1	1
	1	100	-1	1	1	1
	2	100	10	-1	1	1
	3	100	10	10	-1	1
	4	100	10	10	10	-1

Table 2

Model	Cost
Decision Tree	18159
Random Forest	14763
SVM	12068
Neural Network	**10641**

Table 3

4. Conclusion

Heart disease poses a serious threat to human health worldwide. Accurately predicting heart diseases allows patients to receive appropriate treatment earlier. There are many diagnoses available in the medical industry. However, in terms of accuracy and efficiency, machine learning is a better choice. The proposed work used 4 different models for heart disease prediction. Overall, for this heart disease dataset, Support Vector Machine (SVM) performed well, with an 83% accuracy and 0.80 F-measure score. But considering the particularity of this dataset, its cost (12068) is slightly higher than Neural Network (10641). In addition, Neural Network also achieves relatively high accuracy of 75% and a F-measure score of 0.74.

Despite the good performance of these models, there are some limitations. For example, there are some missing values and outliers. Special processing used in this project may result in data being less objective and accurate. Additionally, incrementing some small number of classes may lead to some bias and reduce generalization ability. To address these issues, future work can consider collecting more complete data.

Although this work can help patients improve their efficiency and accurate prediction, the leakage of private data may cause trouble to patients. The data used in this work has been anonymized and widely used by researchers. Subsequent collaboration work can consider hiding the information of the collection agency to further reduce the risk of inferring patient information based on partial data.

References

[1] Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. Information Sciences, 513, 429–441. https://doi.org/10.1016/j.ins.2019.11.004

[2] Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. IEEE Xplore. https://ieeexplore.ieee.org/document/7724478

[3] Marc Peter Deisenroth, A Aldo Faisal, & Cheng Soon Ong. (2020). Mathematics for machine learning. Cambridge University Press.

[4] Scholkopf, B., Kah-Kay Sung, Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Transactions on Signal Processing, 45(11), 2758–2765. https://doi.org/10.1109/78.650102

[5] Zaki, M. J., & Wagner Meira. (2020). Data mining and machine learning: fundamental concepts and algorithms. Cambridge University Press.