Fundamentals of data analysis assessment 3 report

Student name: Zhonghe Wang

Student ID: 25744879

TABLE OF CONTENTS

1. Description of the data mining problem *2*

2. The data preprocessing and transformations *2*

2.1 data understanding **2**

2.2 Missing values **3**

2.3 Duplicate Row Filter **3**

2.4 Column Filter ("remove id") **3**

2.5 Transform age **3**

2.6 Calculate BMI **4**

2.7 Remove abnormal values **4**

2.8 Transform string (yes/no) to number (1/0) **4**

2.9 partitioning **5**

3. solving the problem method *5*

4. Classification techniques *5*

4.1 Decision Tree **5**

4.1.1 Gini index 6

4.1.2 Gain index 8

4.2 k nearest neighbours **9**

4.3 MLP **12**

4.4 SVM **14**

4.4.1 RBF 14

4.4.3 Polynoimal 18

4.5 Random Forest **21**

4.5.1 Information Gain Ratio 21

4.5.2 Information Gain 24

4.5.3 Gini Index 26

4.6 The best classifier **28**

5. Comparison of the pros and cons of classifiers *29*

6. Reflection *31*

6.1 Learnings about data mining 31

6.2 Learnings from problem solving 32

6.3 Approaching the problem differently Next Time 32

a. deeper feature engineering earlier and further 32

b. systematic hyperparameter optimization 33

c. investigate anomalous preprocessing effects 33

d. broader initial algorithm scan with baseline preprocessing 33

e. more fine-grained control error analysis 33

7. Kaggle and oral defense scorer 33

7.1 kaggle scorer 33

7.2 oral defence scorer 33

1. Description of the data mining problem

Data mining uses statistics, machine learning, and AI to uncover useful patterns in large datasets. First, pick a dataset and split it into training and testing sets. Then train a classifier on the training data and measure its accuracy on the test set. Once validated, that model predicts outcomes on new data. Organisations can use these predictions to make informed decisions and improve processes in areas like finance, healthcare, and marketing.

In our study, we apply this framework to a clinical dataset of 56,000 patients. Each record includes demographic and lifestyle features (age in days, gender, weight, height, smoking, alcohol use, physical activity) as well as key biomarkers (systolic and diastolic blood pressure, cholesterol and glucose levels). Specifically, we aim to use machine learning technology to find the best classifier and build the best performance model that predicts whether a patient has cardiovascular disease (cardio = 1) or not (cardio = 0). By capturing the complex interplay among these risk factors, our goal is to flag high-risk individuals early and support targeted preventive care.

2. The data preprocessing and transformations

2.1 data understanding

Our raw dataset contains 56,000 records and 13 attributes, with no missing values. A brief summary of each column follows:

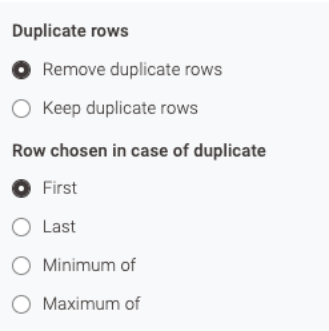
Attribute	Type	Description and action
id	Identifier	Unique patient ID (dropped before modeling)
age	Ratio	Age in days (10 859–23 692); converted to years (age_years) for interpretability
gender	Nominal	1 = female, 2 = male

height	Ratio	Height in centimetres (57–250)
weight	Ratio	Weight in kilograms (11–200)
ap_hi	Ratio	Systolic blood pressure in mmHg (–150–16020) –remove extreme outliers
ap_lo	Ratio	Diastolic blood pressure in mmHg (–70–11 000) – remove extreme outliers
cholesterol	Ordinal	1 = normal, 2 = above normal, 3 = well above normal
gluc	Ordinal	1 = normal, 2 = above normal, 3 = well above normal
smoke	Binary	"No" / "Yes" → recode to 0 / 1
alco	Binary	"No" / "Yes" → recode to 0 / 1
active	Binary	"No" / "Yes" → recode to 0 / 1
cardio	Binary	Target: "No" / "Yes" → recode to 0 / 1 (49.97 % positive, 50.03 % negative – approximately balanced)

2.2 Missing values

No missing values have been found.

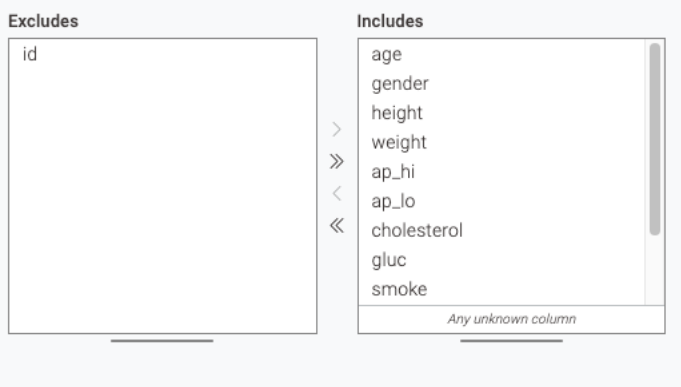
2.3 Duplicate Row Filter



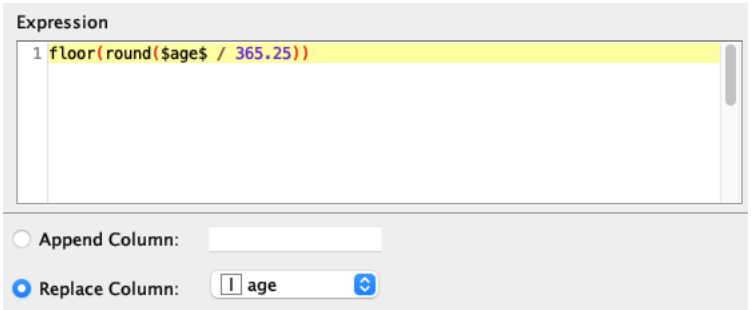
Identify and remove any duplicate patient records to ensure each row appears only once. No same row has been found.

2.4 Column Filter ("remove id")

Drop the ID column, since it is just a unique identifier and not predictive.

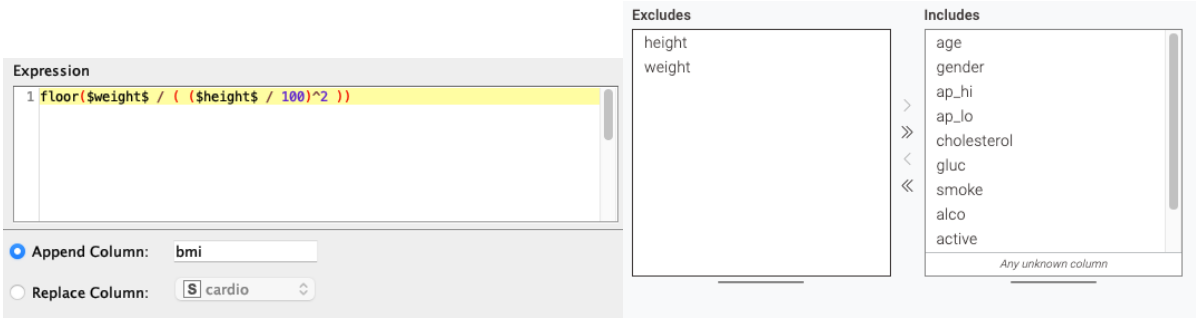


2.5 Transform age



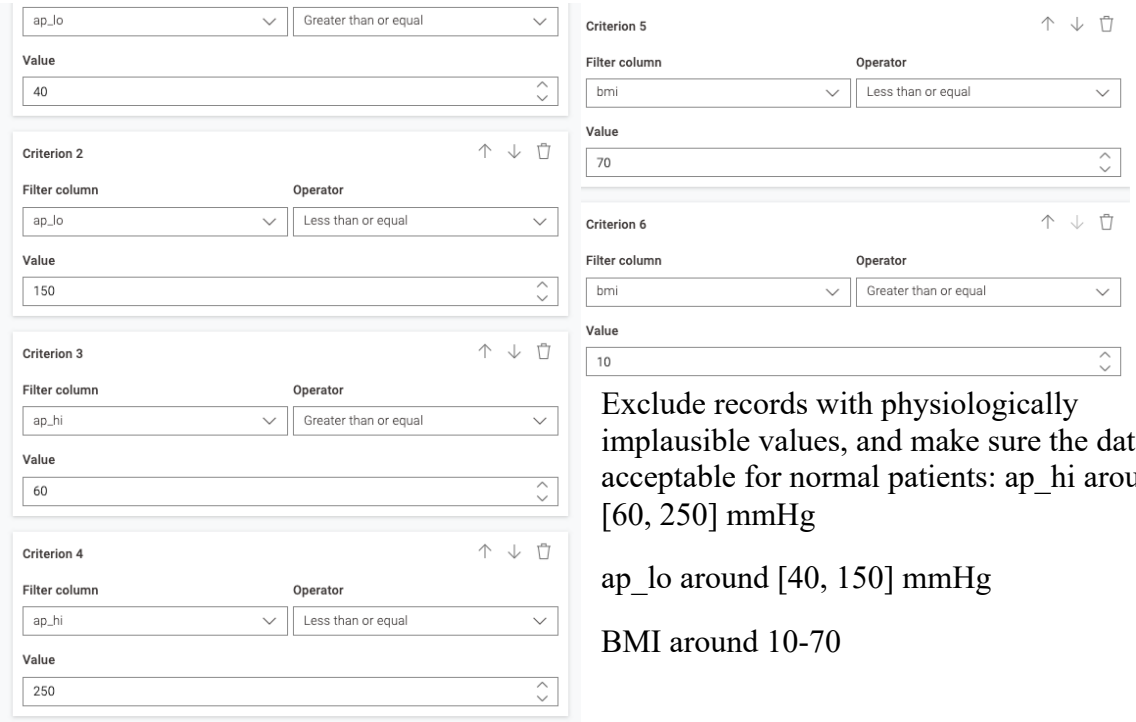
Use a Math Formula to convert age from days to years

2.6 CALCULATE BMI



Compute Body Mass Index (BMI) from weight (kg) and height (cm), and remove weight and height

2.7 REMOVE ABNORMAL VALUES



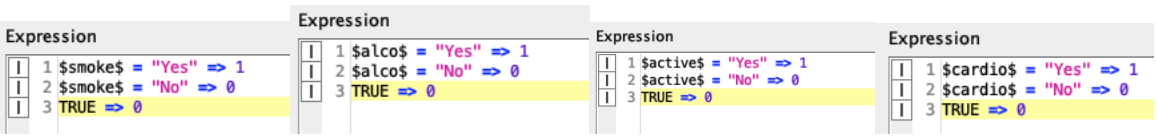
Exclude records with physiologically implausible values, and make sure the data is acceptable for normal patients: ap_hi around [60, 250] mmHg

ap_lo around [40, 150] mmHg

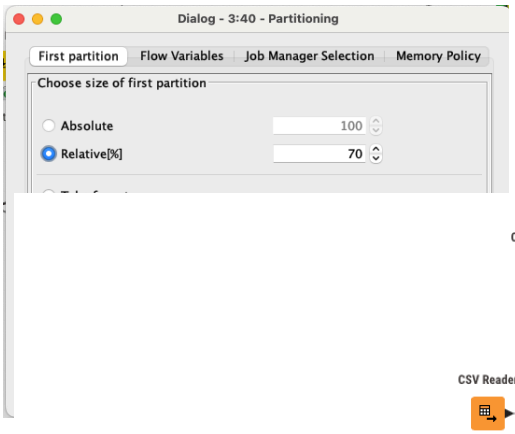
BMI around 10-70

2.8 TRANSFORM STRING (YES/NO) TO NUMBER (1/0)

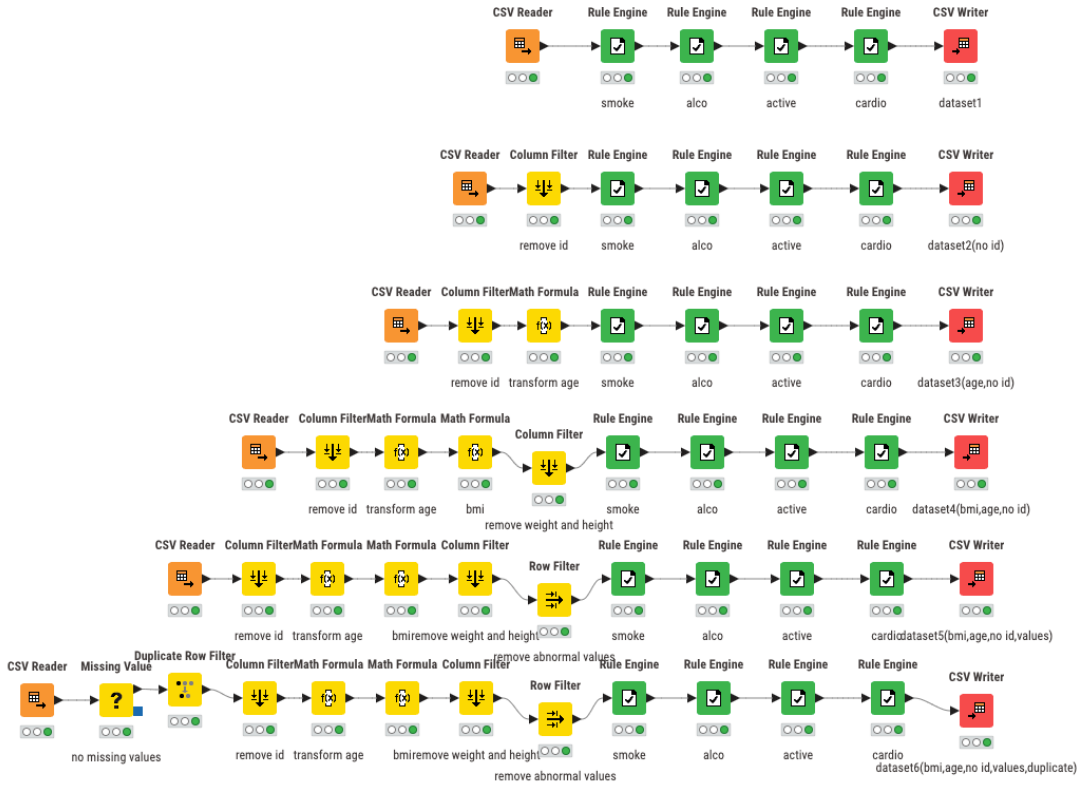
Convert the string "yes" or "no" in the target column（smoke, alco, active, cardio） to numeric values "1" or "0".



2.9 PARTITIONING



It split the data into 70% (training set) and 30% (test set). It uses a random draw method and uses the random seed 42 to test.



3. SOLVING THE PROBLEM METHOD

The approach uses KNIME, adopting the controlled experiment method, using different data preprocessing techniques and classification algorithms to find the most suitable and best prediction accuracy classifier to predict the risk of cardio.

In this report, various data preprocessing techniques and classification algorithms are used to test, use different parameter settings to improve the accuracy of prediction.

After training the model, the predictor of the best performance model would be used to predict the unknown dataset without the target label.

4. CLASSIFICATION TECHNIQUES

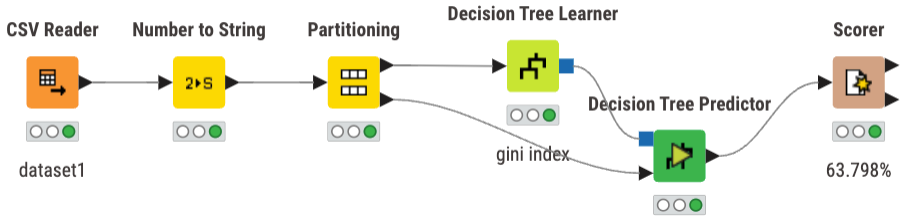
4.1 DECISION TREE

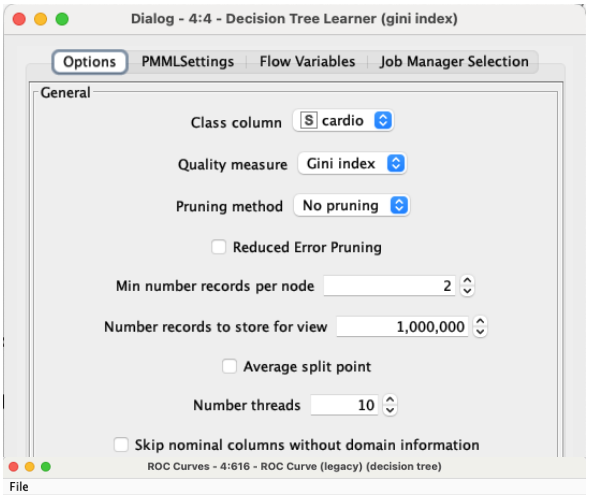
The Decision Tree Learner node then treats cardio as the target and applies the Gini index to choose the best splits among features like age (years), BMI, blood pressure, cholesterol, glucose, and lifestyle flags. Pruning remains disabled so the tree can learn detailed patterns,

while the default minimum records per node ensures nodes only split when there is enough data. Finally, the Predictor and Scorer nodes apply the model to the unseen test set and compute performance metrics—accuracy, precision, recall, and F1-score—to assess its predictive power.

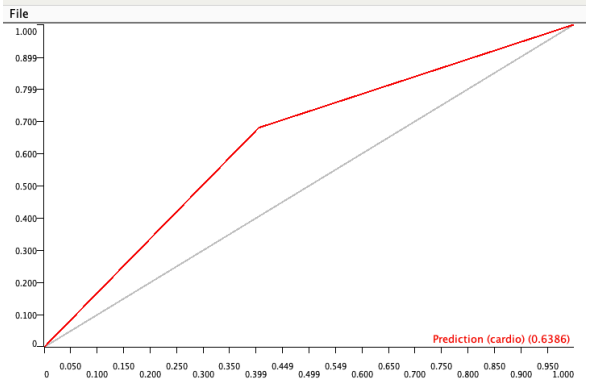
4.1.1 GINI INDEX

A. DEFAULT DATASET AND SETTING

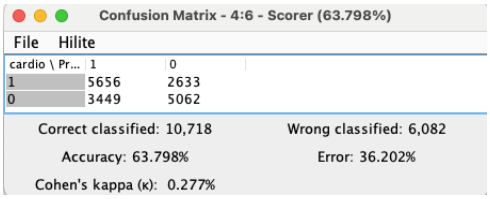




A KNIME workflow was built to train and evaluate a Decision Tree classifier on our cardiovascular dataset, using all default settings. First, the Partitioning node splits the cleaned data into 70 % training and 30 % test sets with random draw on the cardio column (random seed = 42) to preserve the disease/control balance.

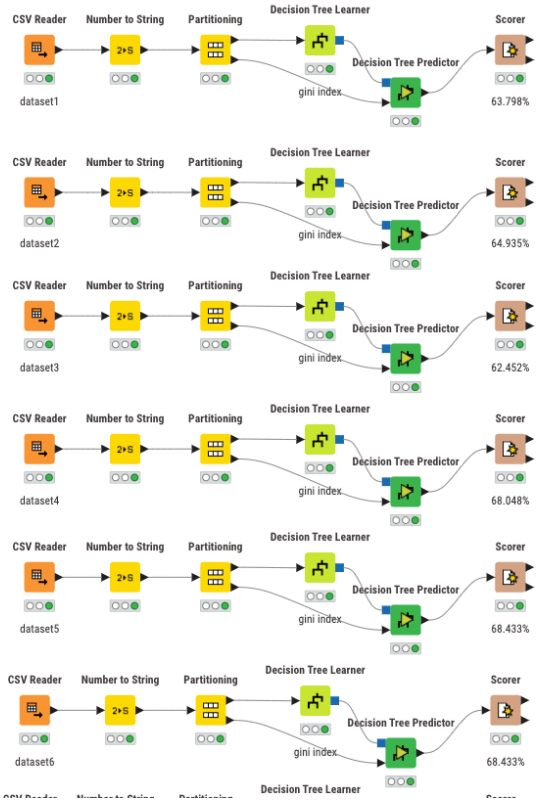


The red ROC curve lies just above the grey diagonal (random chance), with an AUC of 0.639. This indicates the decision tree has a moderate ability to distinguish between patients with and without cardiovascular disease, better than random guessing, but leaving room for improvement.



#	RowID	TruePositives	FalsePositiv...	TrueNegati...	FalseNegati...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kap...
		Number (integer)	Number (integer)	Number (integer)	Number (integer)	Number (double)	Number (double)	Number (double)	Number (double)	Number (double)	Number (double)	Number (double)
1	1	5656	3449	5062	2633	0.682	0.621	0.682	0.595	0.65		
2	0	5062	2633	5656	3449	0.595	0.658	0.595	0.682	0.625		
3	Overall										0.638	0.277

B. DIFFERENT PRE-PROCESSED DATASET

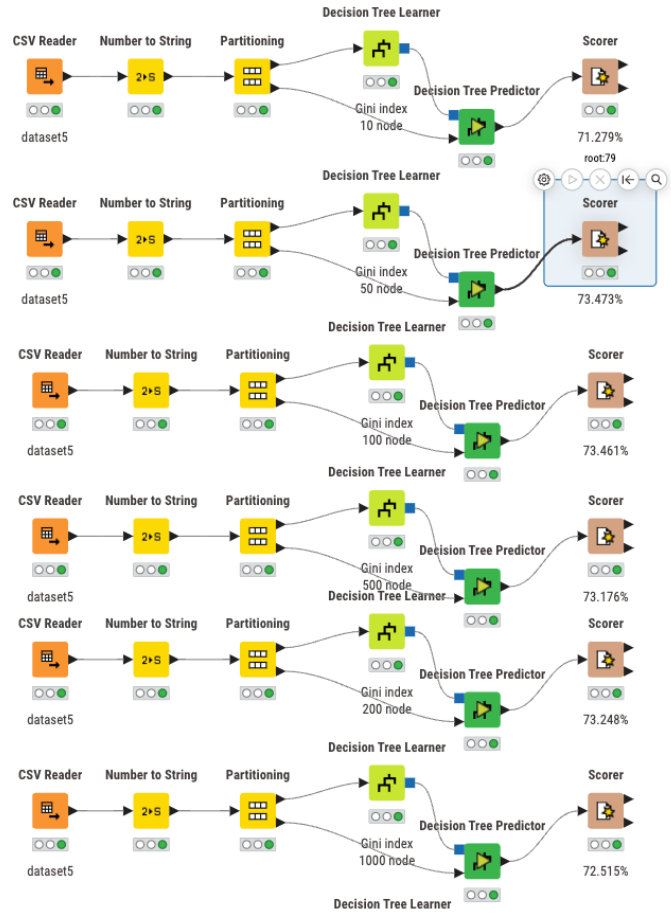


The figure compares test accuracies from the same Decision Tree model trained on six versions of dataset—each processed with progressively more advanced cleaning and feature engineering. A clear upward trend can be found:

dataset1 (no preprocessing) → 63.78 %and dataset6 (most thorough preprocessing) → 68.43 %

In other words, the more intensive the data-processing pipeline, the better the model's accuracy on unseen data.

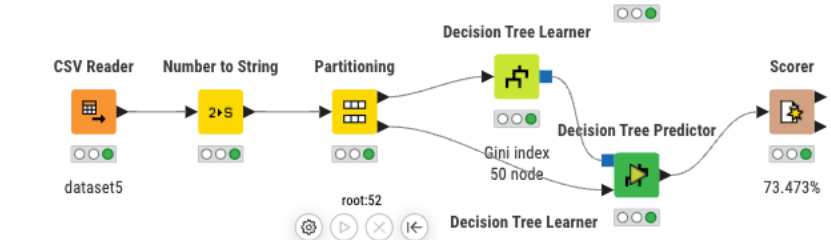
C. Different settings to find the best accuracy

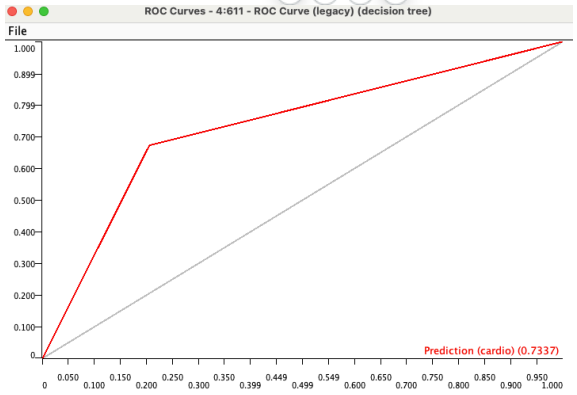


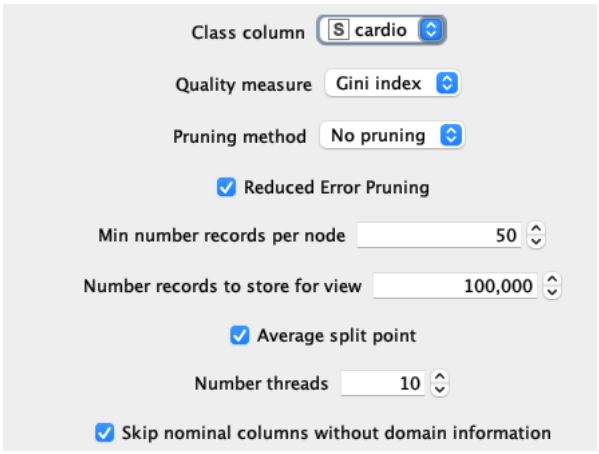
This figure evaluates how limiting the Decision Tree's maximum size affects its accuracy on dataset5.

Accuracy rises sharply from a very shallow tree (10 nodes) to around 50–100 nodes, then plateaus and even dips slightly as the tree grows larger. This suggests a moderate tree complexity best balances underfitting and overfitting for this cardiovascular dataset.

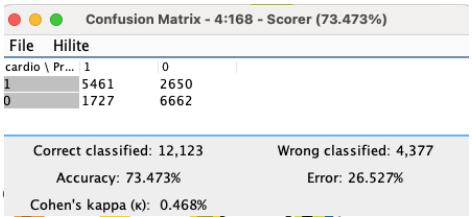
D. BEST RESULT SETTING

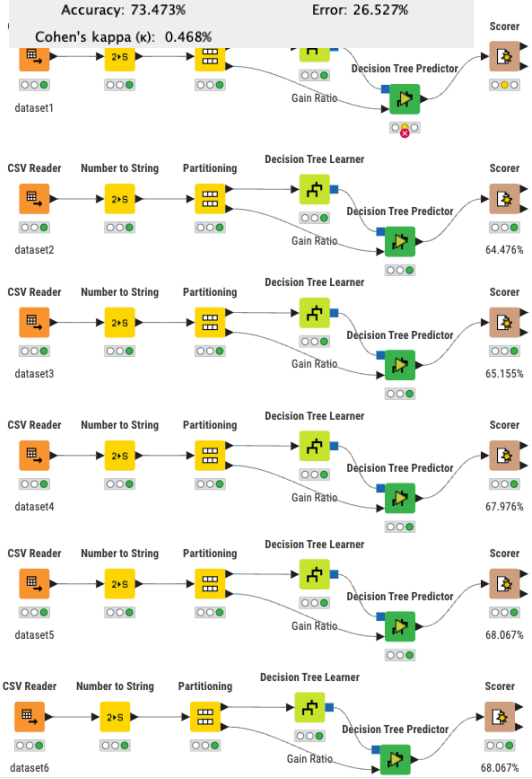






	#	RowID	TruePosit...	FalsePosi...	TrueNega...	FalseNeg...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's k...
			Number (inte...	Number (inte...	Number (inte...	Number (inte...	Number (dou...	Number (dou...	Number (dou...	Number (dou...	Number (dou...	Number (dou...	Number (dou...
☐	1	1	5488	1755	6634	2623	0.677	0.758	0.677	0.791	0.715	?	?
☐	2	0	6634	2623	5488	1755	0.791	0.717	0.791	0.677	0.752	?	?
☐	3	Overall	?	?	?	?	?	?	?	?	?	0.735	0.468





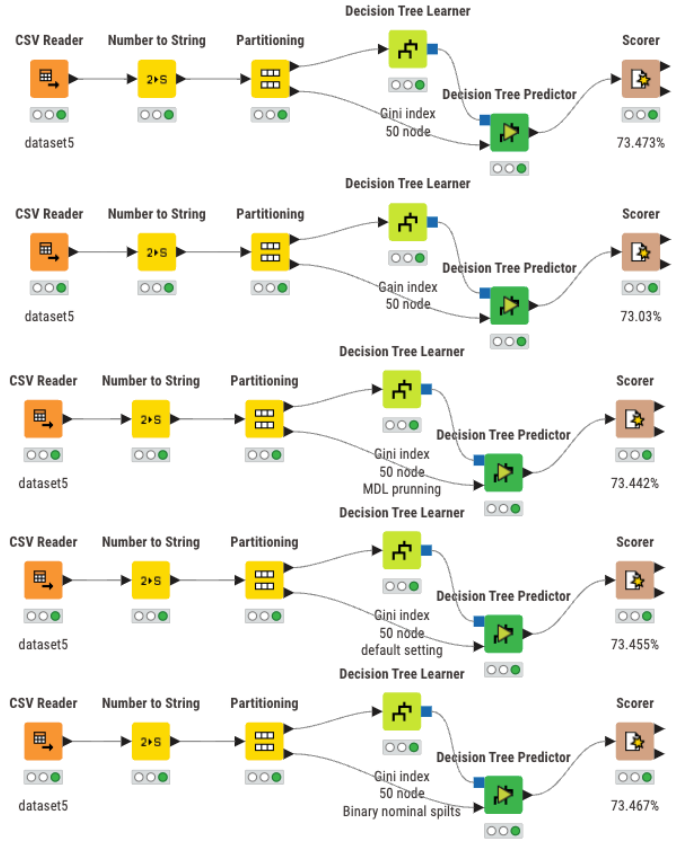
By experimenting with different parameter settings(like node, pruning, number threads, etc.) in the Decision Tree Learner, we identified the best parameter setting that delivered the highest accuracy-73.473%. Compared to the default setting, this setting increased the AUC to 0.7337.

4.1.2 GAIN INDEX

A. DIFFERENT PRE-PROCESSED DATASET

This figure repeats the same six-step KNIME pipeline on datasets 1–6, but switches the quality measure method from Gini to Gain Ratio. As with the Gini coefficient, there is a clear upward trend in accuracy as the data processing pipeline improves. Using the gain rate yields similar performance—with accuracy peaking at around 68% on the best-processed data.

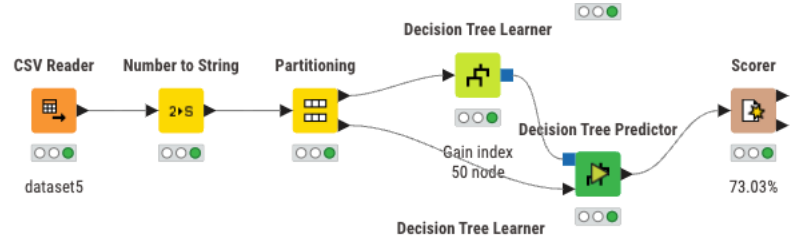
B. Different settings to find the best result



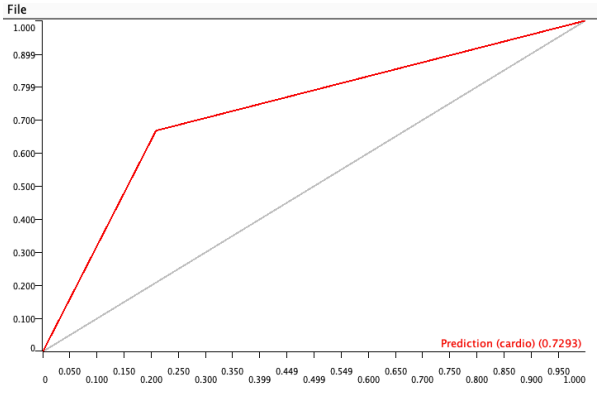
This workflow explores how different quality measures and parameter settings affect the Decision Tree's accuracy on dataset5.

The best performance still comes from the Gini-based workflow, while alternative measures or pruning options yield only marginally lower accuracy.

C. Best result setting



The best result of parameter setting is using Gain Index is consistent with Gini, but the accuracy is 73.03%, which is lower than Gini's 73.467%.



According to the ROC curve, AUC is 0.7293, which is lower than Gini's 0.7337.

Comparing the accuracy of the unprocessed and processed datasets also shows that the model trained after data processing performs better and has better prediction accuracy.

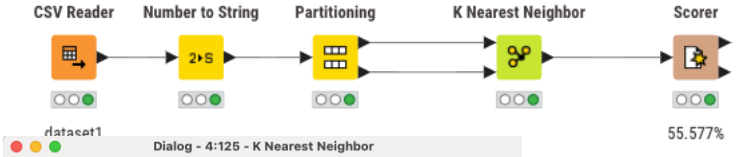
#	RowID	TruePosit...	FalsePosi...	TrueNega...	FalseNeg...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's k...
		Number (inte...	Number (inte...	Number (inte...	Number (inte...	Number (dou...	Number (dou...	Number (dou...	Number (dou...	Number (dou...	Number (dou...	Number (dou...
1	1	5461	1727	6662	2650	0.673	0.76	0.673	0.794	0.714		
2	0	6662	2650	5461	1727	0.794	0.715	0.794	0.673	0.753		
3	Overall										0.735	0.468

4.2 K nearest neighbours

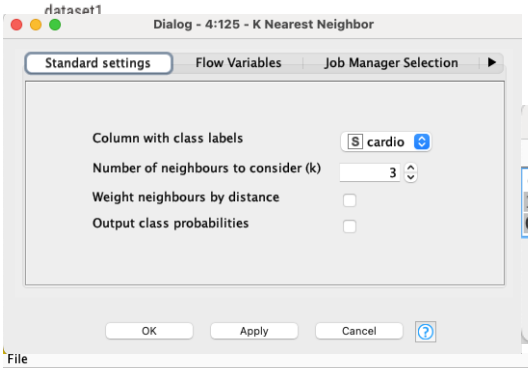
A. DEFAULT DATASET AND SETTING

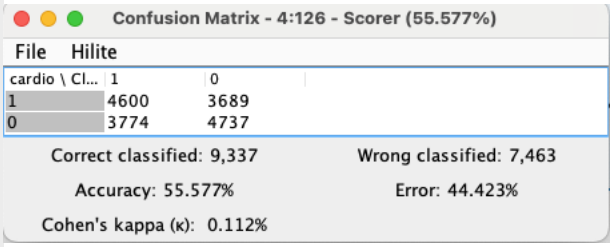
The K Nearest Neighbor node treats ***cardio*** as the target variable and classifies each test instance based on the majority label among its k closest training samples, using a distance metric such as Euclidean distance. KNN memorises the training data and performs instance-based learning. The number of neighbors (***k***) and whether to weight them (e.g., by distance) influence the decision boundary.

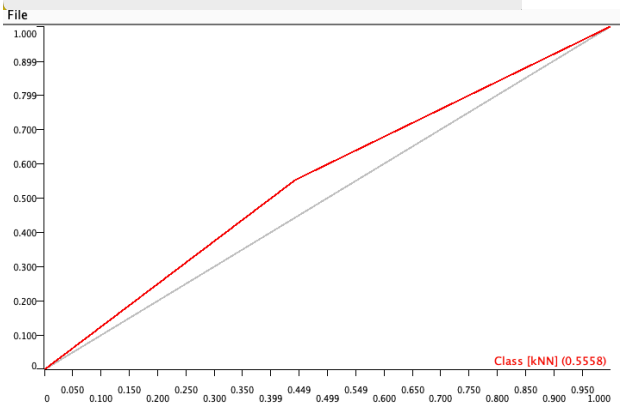
KNIME workflow was built to train and evaluate a K-nearest neighbour classifier on our dataset, using all default settings.



The accuracy is 55.577%, and it's not performing well, so it needs to be improved to gain more reliable predictions.



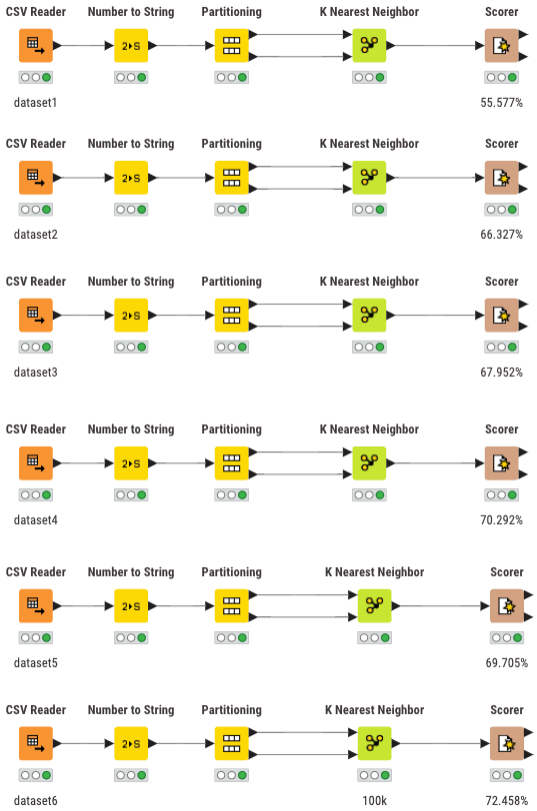




The red ROC curve for the KNN model hugs the grey diagonal (random chance) very closely, with an AUC of **0.556**. This indicates the K Nearest Neighbor classifier only slightly outperforms random guessing and has limited ability to distinguish between patients with and without cardiovascular disease across all thresholds.

#	RowID	TruePositi... Number (integ..	FalsePosi... Number (integ..	TrueNega... Number (integ..	FalseNeg... Number (integ..	Recall Number (dou..	Precision Number (dou..	Sensitivity Number (dou..	Specificity Number (dou..	F-measure Number (dou..	Accuracy Number (dou..	Cohen's k... Number (dou..
1	1	4600	3774	4737	3689	0.555	0.549	0.555	0.557	0.552	?	?
2	0	4737	3689	4600	3774	0.557	0.562	0.557	0.555	0.559	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.556	0.112

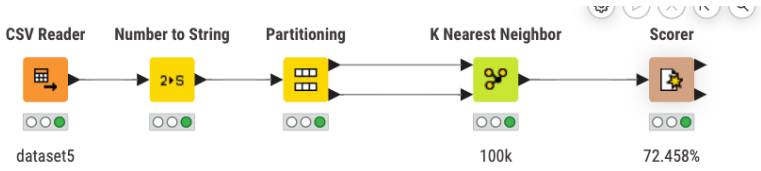
B. PRE-PROCESSED DATASET



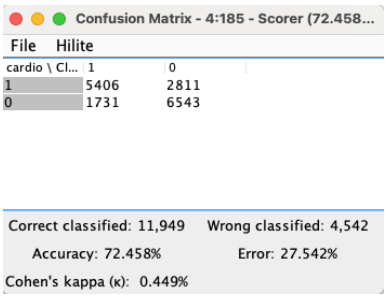
This figure applies the same KNN workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies rise steadily as preprocessing becomes more thorough: dataset1: 55.577%--dataset5: 69.705%, but dataset4 has the best performance-70.292%. So, about this classifier, it is possible that dataset 4 could have the best performance at least.

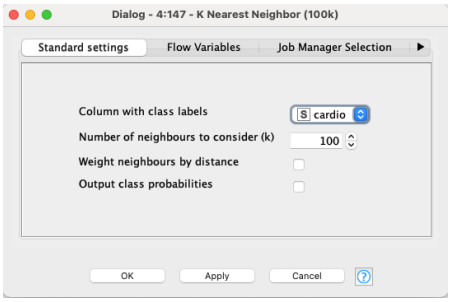
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS

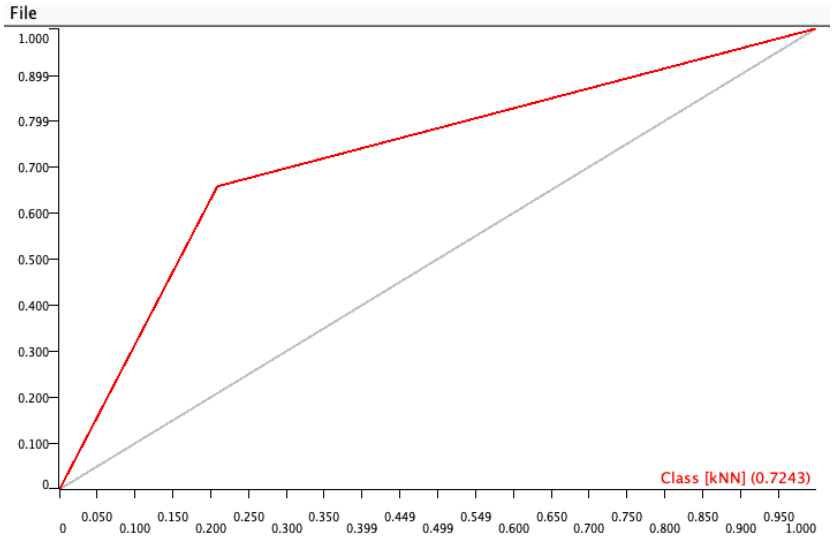
This workflow applies the same KNIME pipeline to **dataset5**, varying the number of neighbours to consider. The 100(k) neighbours deliver the highest accuracy, while using a higher or lower number of neighbours results in only marginally lower performance.



D. BEST RESULT SETTING





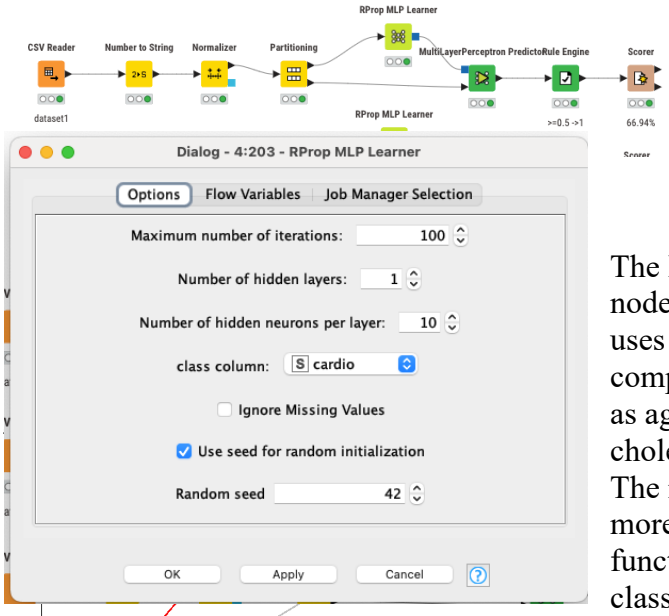


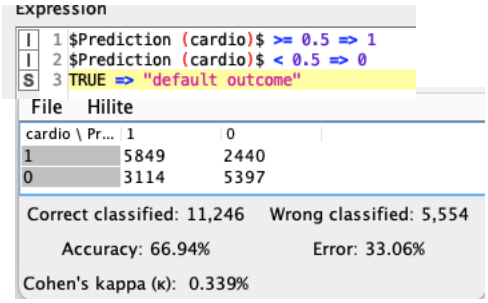
By experimenting with different parameter settings(number of neighbours to consider) in the K Nearest Neighbor, we identified the best parameter setting(100k) that delivered the highest accuracy-**72.458%.** Compared to the default setting, this setting increased the AUC to **0.7243**.

#	RowID	TruePositi...	FalsePosit...	TrueNegat...	FalseNega...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's k...
		Number (integ..	Number (integ..	Number (integ..	Number (integ..	Number (doub..	Number (doub..	Number (doub..	Number (doub..	Number (doub..	Number (doub..	Number (doub..
1	1	5406	1731	6543	2811	0.658	0.757	0.658	0.791	0.704		
2	0	6543	2811	5406	1731	0.791	0.699	0.791	0.658	0.742		
3	Overall										0.725	0.449

4.3 MLP

A. DEFAULT DATASET

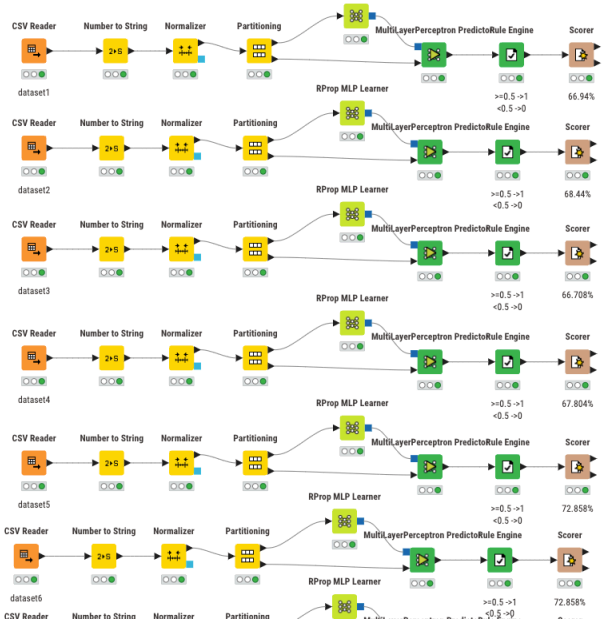




The Multi-Layer Perceptron (MLP) Learner node treats cardio as the target variable and uses a feedforward neural network to learn complex patterns from the input features, such as age (years), BMI, blood pressure, cholesterol, glucose, and lifestyle indicators. The network consists of an input layer, one or more hidden layers with nonlinear activation functions, and an output layer that performs classification. During training, the model adjusts weights using backpropagation and gradient descent to minimize prediction error.

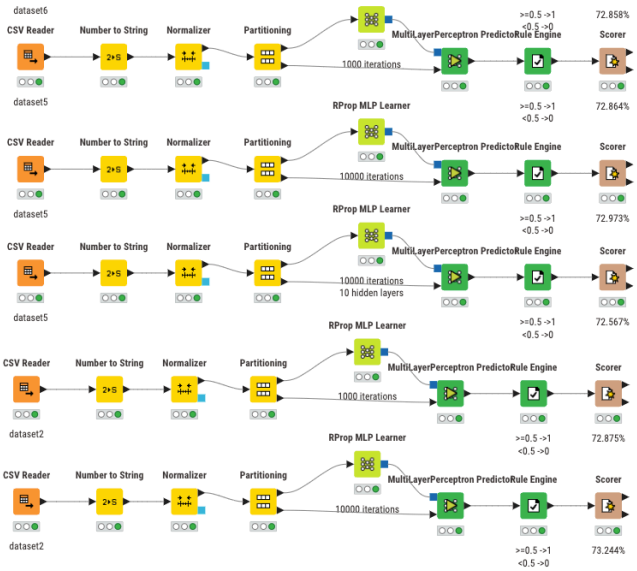
The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.7324**. This indicates the MLP has a great ability to distinguish between patients with and without cardiovascular disease, better than random guessing, but leaving room for improvement.

B. PRE-PROCESSED DATASET AND DIFFERENT METHODS



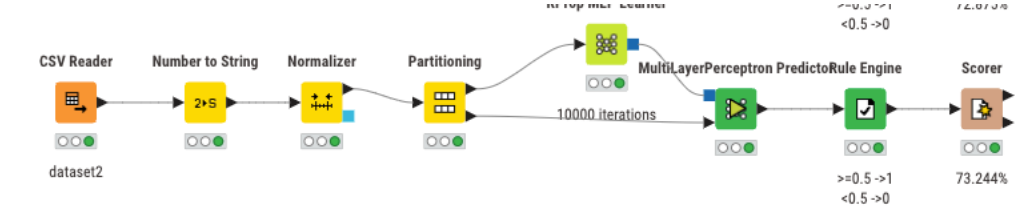
This figure applies the same MLP workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies rise steadily as preprocessing becomes more thorough: dataset1: 66.94%--dataset5: **72.858%.**

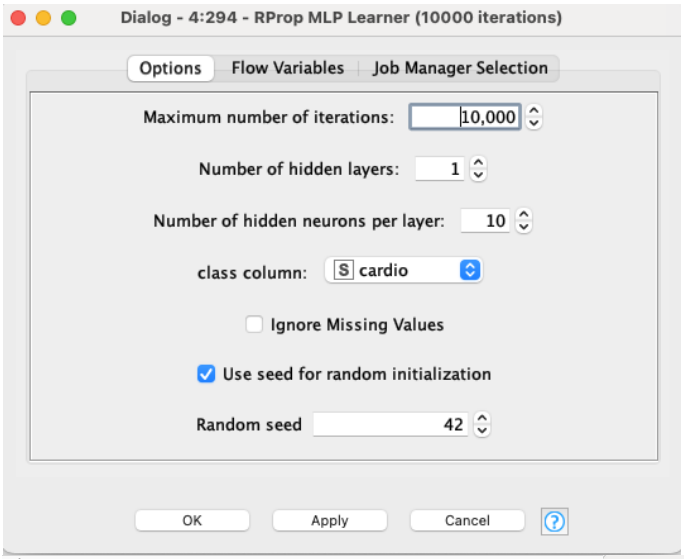
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS



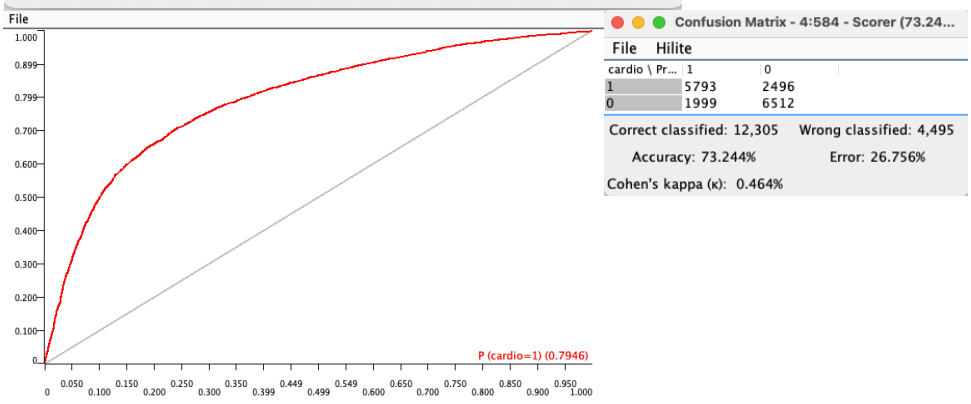
This workflow applies the same KNIME pipeline to **dataset 5 and dataset 2**, varying the number of iterations and hidden layers to consider. The **10000 iterations** deliver the highest accuracy, while using a lower iterations and higher hidden layers results in only marginally lower performance.

D. BEST RESULT SETTING





By experimenting with different parameter settings(maximum number of iterations) in the MLP, we identified the best parameter setting(100k) that delivered the highest accuracy- **73.244%.** Compared to the default setting, this setting increased the AUC to **0.7946**.



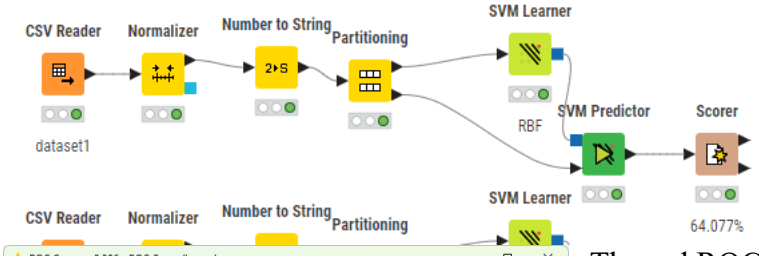
#	RowID	TruePositi...	FalsePosit...	TrueNegat...	FalseNega...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's ka...
1	1	5793	1999	6512	2496	0.699	0.743	0.699	0.765	0.72		
2	0	6512	2496	5793	1999	0.765	0.723	0.765	0.699	0.743		
3	Overall										0.732	0.464

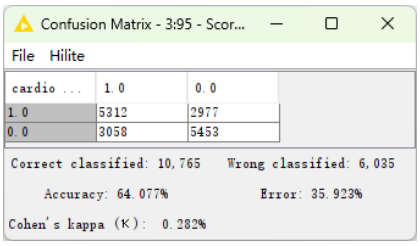
4.4 SVM

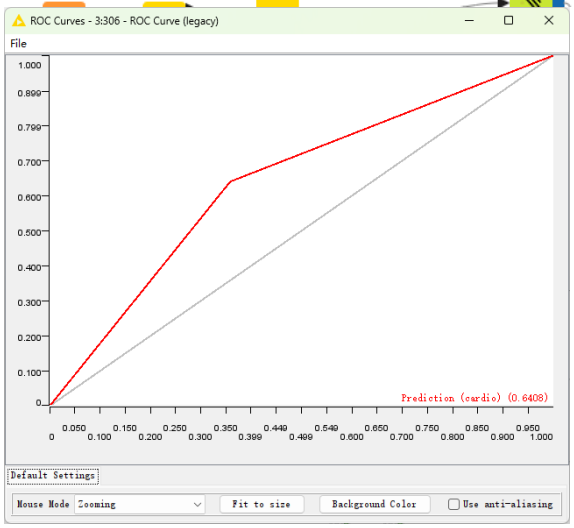
The Support Vector Machine (SVM) Learner node treats cardio as the target variable and aims to find the optimal hyperplane that best separates the data into two classes—those with and without cardiovascular risk. It does so by maximising the margin between the closest data points of each class (support vectors), and can utilise kernel functions to handle nonlinear relationships among features like age (years), BMI, blood pressure, cholesterol, glucose, and lifestyle indicators. During training, the model identifies the decision boundary that generalises well to unseen data.

4.4.1 RBF

A. DEFAULT DATASET AND SETTING



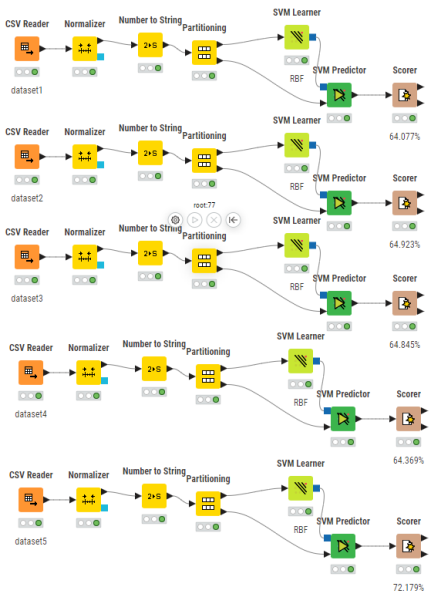




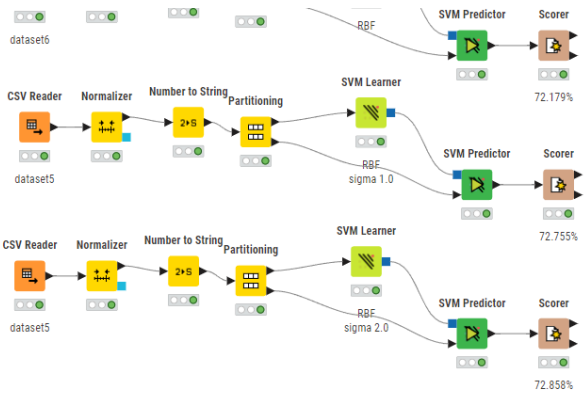
The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.6408**. This indicates the random forest has a lower ability to distinguish between patients with and without cardiovascular disease, a little better than random guessing, leaving room for improvement.

#	RowID	TruePositi... Number (in...	FalsePosi... Number (in...	TrueNega... Number (in...	FalseNeg... Number (in...	Recall Number (d...	Precision Number (d...	Sensitivity Number (d...	Specificity Number (d...	F-measure Number (d...	Accuracy Number (d...	Cohen's k... Number (d...
1	1.0	5312	3058	5453	2977	0.641	0.635	0.641	0.641	0.638	?	?
2	0.0	5453	2977	5312	3058	0.641	0.647	0.641	0.641	0.644	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.641	0.282

B. PRE-PROCESSED DATASET

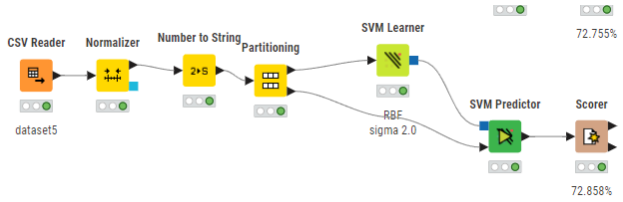


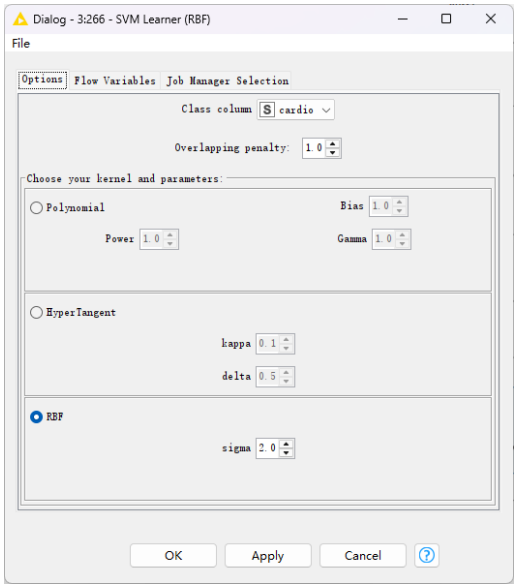
This figure applies the same SVM(RBF) workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies rise steadily as preprocessing becomes more thorough: dataset1: 64.077%--dataset5: **72.179%.**



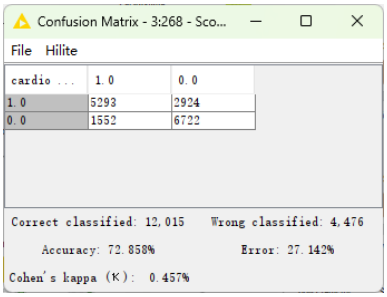
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS: This workflow applies the same KNIME pipeline to **dataset 5,** varying sigma. The **2.0 sigma** delivers the highest accuracy, while using a lower and higher sigma results in only marginally lower performance.

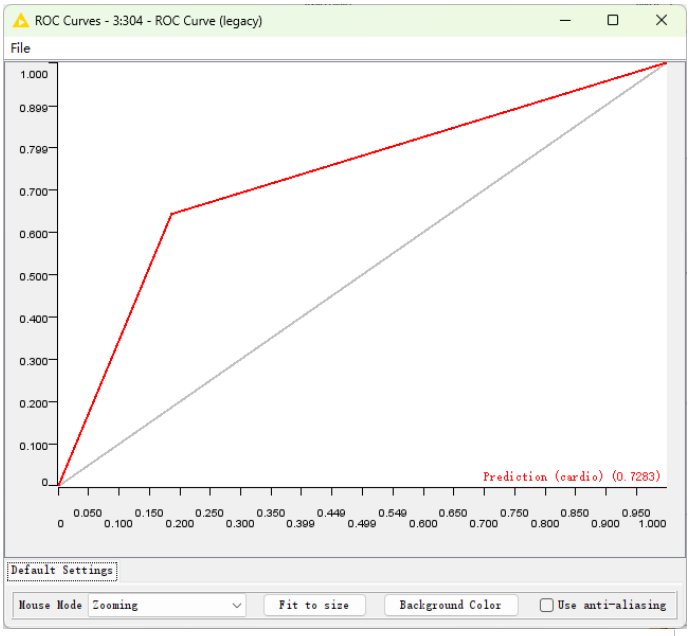
D. BEST RESULT SETTING





By experimenting with different parameter settings(sigma) in the SVM, we identified the best parameter setting (sigma 2.0) that delivered the highest accuracy-**72.858%.** Compared to the default setting, this setting increased the AUC to **0.7283**.

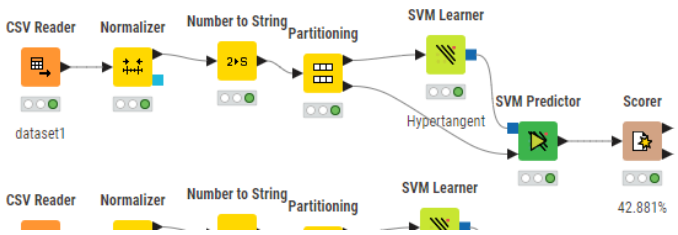


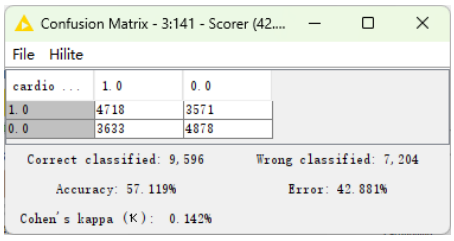


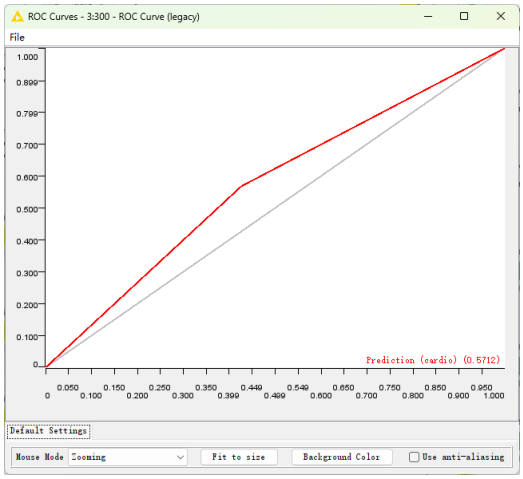
#	RowID	TruePositi... Number (in...	FalsePosit... Number (in...	TrueNegat... Number (in...	FalseNega... Number (in...	Recall Number (d...	Precision Number (d...	Sensitivity Number (d...	Specificity Number (d...	F-measure Number (d...	Accuracy Number (d...	Cohen's k... Number (d...
1	1.0	5293	1552	6722	2924	0.644	0.773	0.644	0.812	0.703	?	?
2	0.0	6722	2924	5293	1552	0.812	0.697	0.812	0.644	0.75	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.729	0.457

4.4.2 HYPERTANGENT

A. DEFAULT DATASET AND SETTING



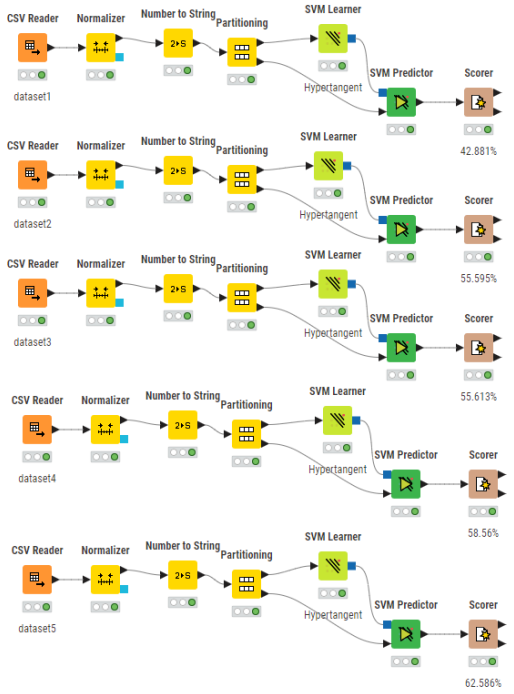




The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.5712**. This indicates the random forest has a poor ability to distinguish between patients with and without cardiovascular disease, just like random guessing, leaving room for improvement.

#	RowID	TruePositi... Number (int...	FalsePosit... Number (int...	TrueNegat... Number (int...	FalseNega... Number (int...	Recall Number (do...	Precision Number (do...	Sensitivity Number (do...	Specificity Number (do...	F-measure Number (do...	Accuracy Number (do...	Cohen's ka... Number (do...
1	1.0	4718	3633	4878	3571	0.569	0.565	0.569	0.573	0.567		
2	0.0	4878	3571	4718	3633	0.573	0.577	0.573	0.569	0.575		
3	Overall										0.571	0.142

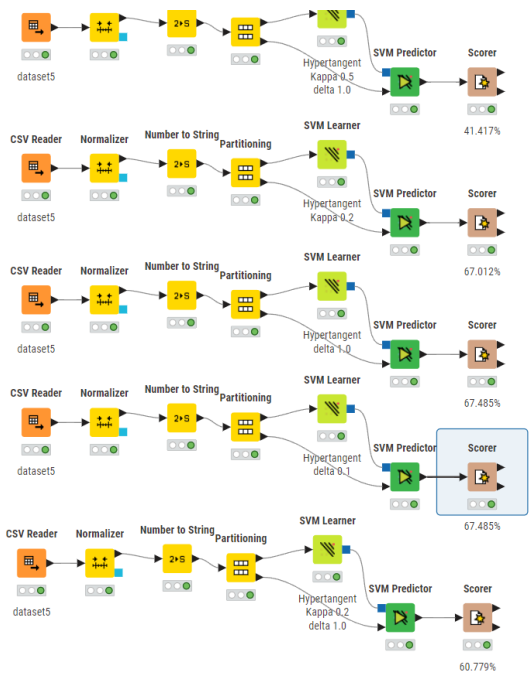
B. PRE-PROCESSED DATASET AND DIFFERENT METHODS



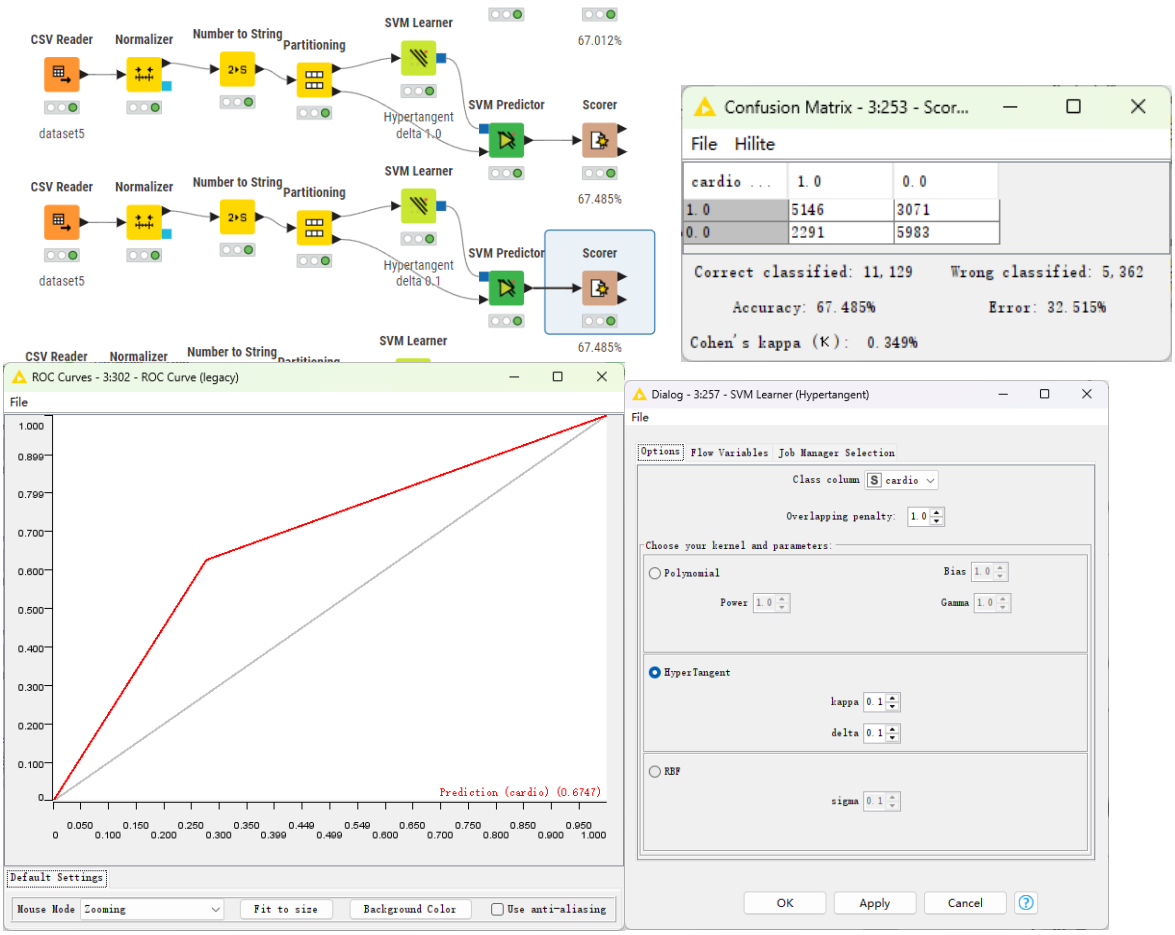
This figure applies the same SVM (hypertangent) workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies rise steadily as preprocessing becomes more thorough: dataset1: 42.881%--dataset5: **62.586%.**

C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS

This workflow applies the same KNIME pipeline to **dataset 5**, varying kappa and delta. The **delta 0.1** delivers the highest accuracy, while using a lower and higher kappa and delta results in only marginally lower performance.



D. BEST RESULT SETTING

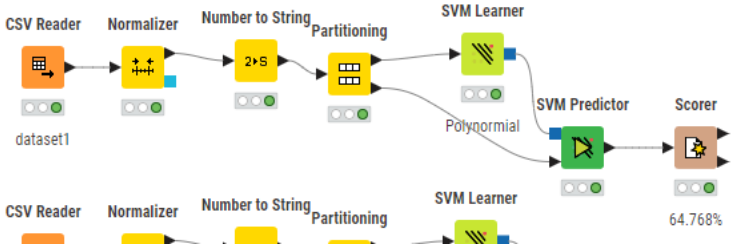


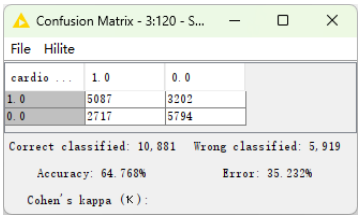
By experimenting with different parameter settings(kappa, delta) in the SVM, we identified the best parameter setting in HyperTangent (**kappa 0.1, delta 0.1**) that delivered the highest accuracy-**67.485**%. Compared to the default setting, this setting increased the AUC to 0.6747.

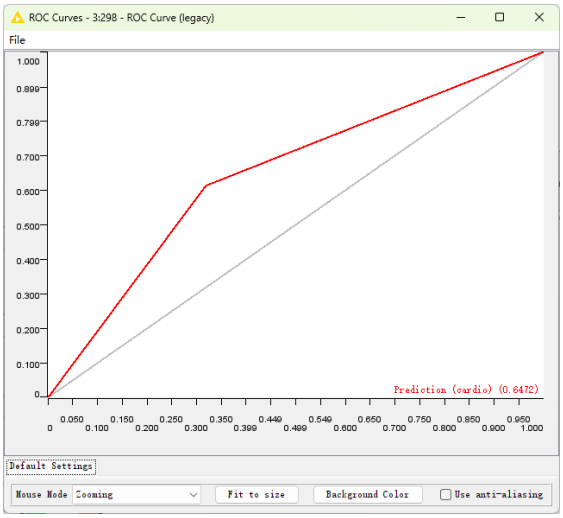
#	RowID	TruePositi... Number (in...	FalsePosit... Number (in...	TrueNega... Number (in...	FalseNeg... Number (in...	Recall Number (d...	Precision Number (d...	Sensitivity Number (d...	Specificity Number (d...	F-measure Number (d...	Accuracy Number (d...	Cohen's k... Number (d...
1	1.0	5146	2291	5983	3071	0.626	0.692	0.626	0.723	0.657	?	?
2	0.0	5983	3071	5146	2291	0.723	0.661	0.723	0.626	0.691	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.675	0.349

4.4.3 POLYNOIMAL

A. DEFAULT DATASET AND SETTING

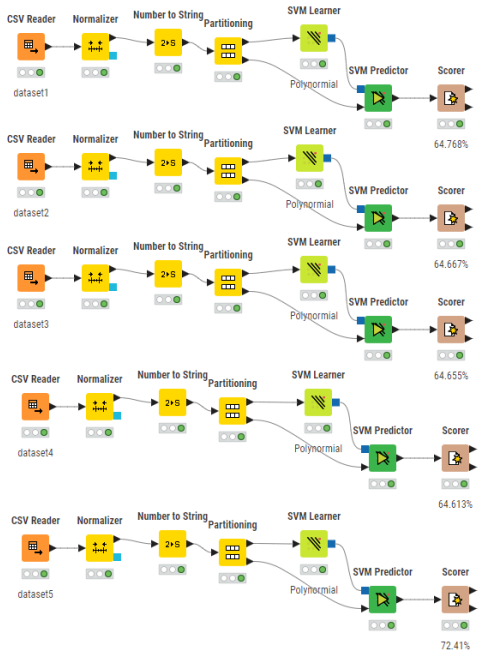






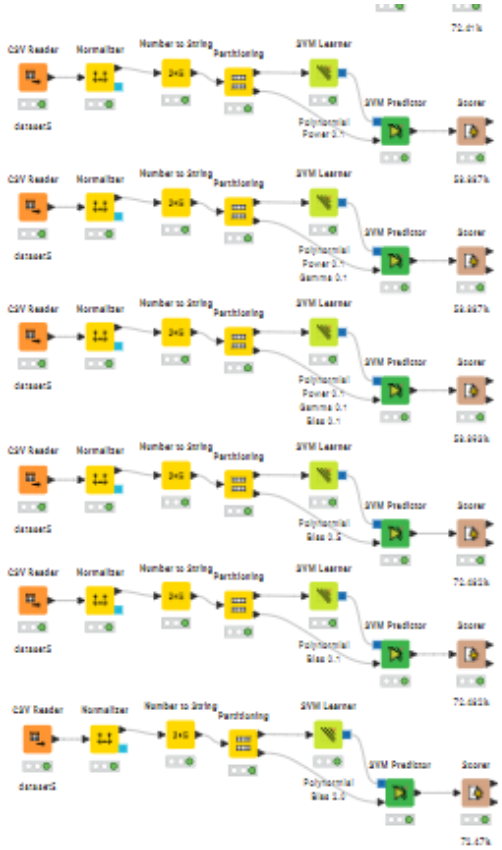
The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.6472**. This indicates the random forest has a low ability to distinguish between patients with and without cardiovascular disease, a little bit better than random guessing, leaving room for improvement.

B. PRE-PROCESSED DATASET AND DIFFERENT METHODS



This figure applies the same SVM (polynomial) workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies rise steadily as preprocessing becomes more thorough: dataset1: **64.768**%--dataset5: **72.41**%.

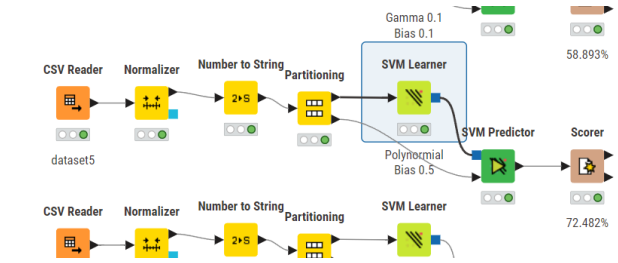
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS

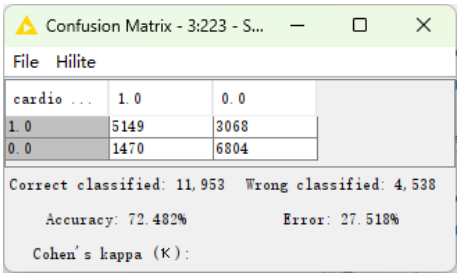


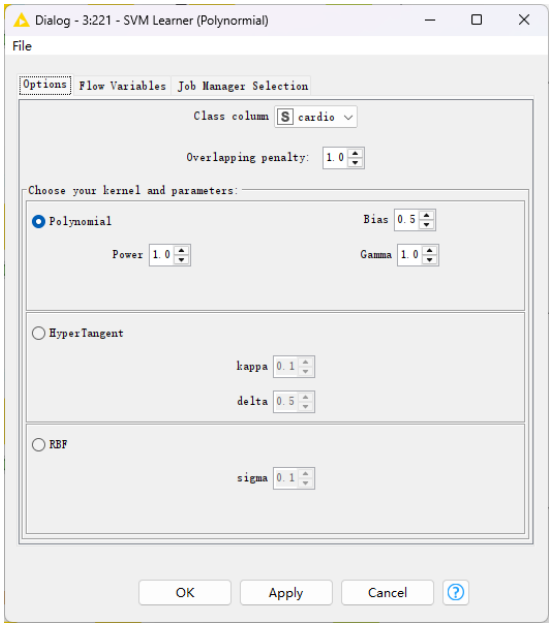
This figure applies the same SVM(Polynomial) workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies lower steadily as preprocessing becomes more thorough: dataset1: **73.321**%--dataset5: 72.912%.

Which means that more data pre-processing results in lower accuracy. So, the non-preprocessed data(**dataset1**) may have had a better result.

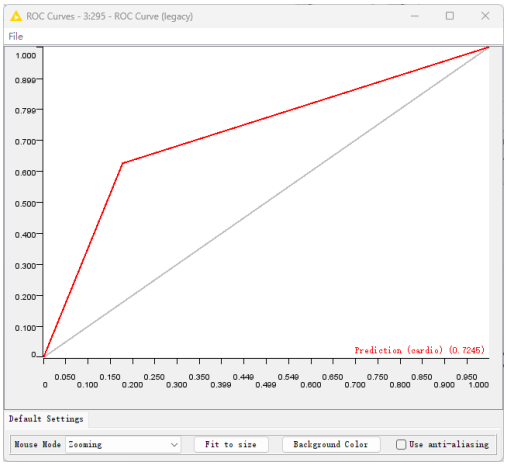
D. BEST RESULT SETTING







By experimenting with different parameter settings (power, bias, gamma) in the SVM (Polynomial), we identified the best parameter setting(power1.0, bias0.5, gamma1.0) that delivered the highest accuracy-**72.858**%. Compared to the default setting, this setting increased the AUC to **0.7245**.



Rows: 3 | Columns: 11

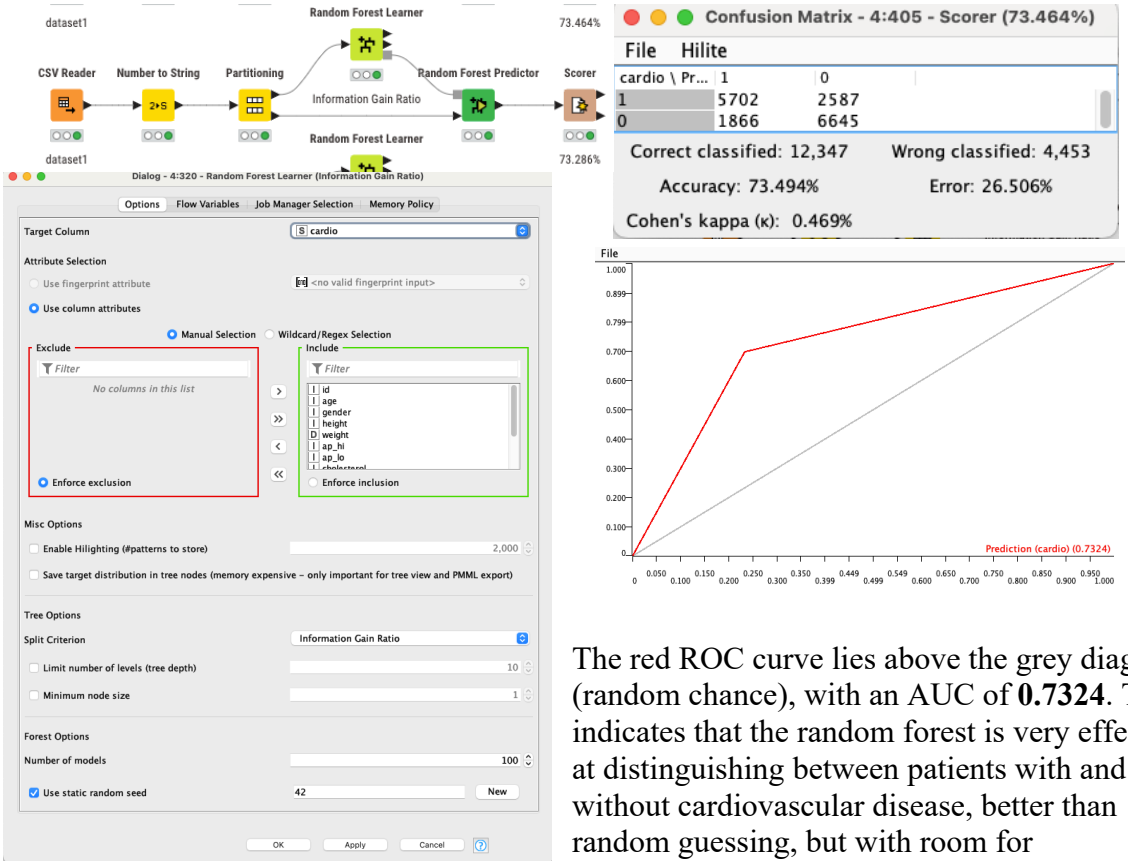
#	RowID	TruePosit... Number (i...	FalsePosi... Number (i...	TrueNega... Number (i...	FalseNeg... Number (i...	Recall Number (d...	Precision Number (d...	Sensitivity Number (d...	Specificity Number (d...	F-measure Number (d...	Accuracy Number (d...	Cohen's k. Number (d
1	1.0	5149	1470	6804	3068	0.627	0.778	0.627	0.822	0.694	?	?
2	0.0	6804	3068	5149	1470	0.822	0.689	0.822	0.627	0.75	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.725	0.449

4.5 Random Forest

The Random Forest Learner node treats cardio as the target variable and builds an ensemble of decision trees to improve predictive accuracy and reduce overfitting. Each tree is trained on a different random subset of the data and considers a random subset of features, such as age (years), BMI, blood pressure, cholesterol, glucose, and lifestyle factors, when determining splits. The final prediction is made by aggregating the outputs of all trees, typically through majority voting. This ensemble approach captures complex patterns while maintaining robustness.

4.5.1 Information Gain Ratio

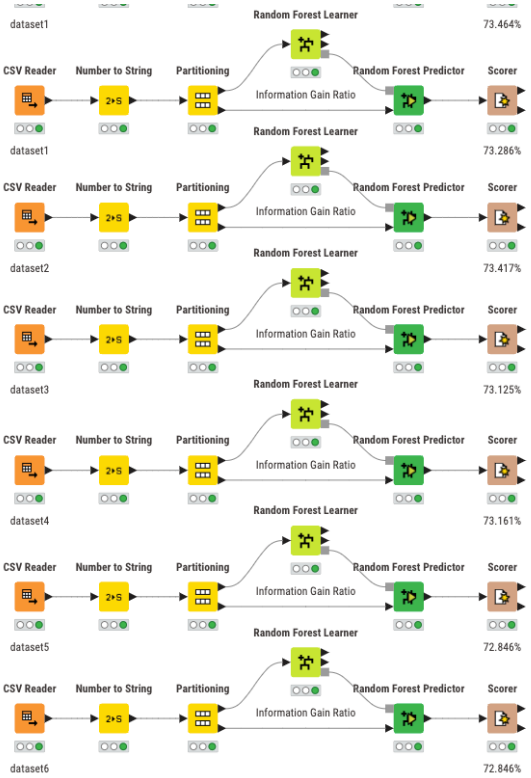
A. Default Dataset and Setting



The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.7324**. This indicates that the random forest is very effective at distinguishing between patients with and without cardiovascular disease, better than random guessing, but with room for improvement.

#	RowID	TruePositi... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	1	5702	1866	6645	2587	0.688	0.753	0.688	0.781	0.719		
2	0	6645	2587	5702	1866	0.781	0.72	0.781	0.688	0.749		
3	Overall										0.735	0.469

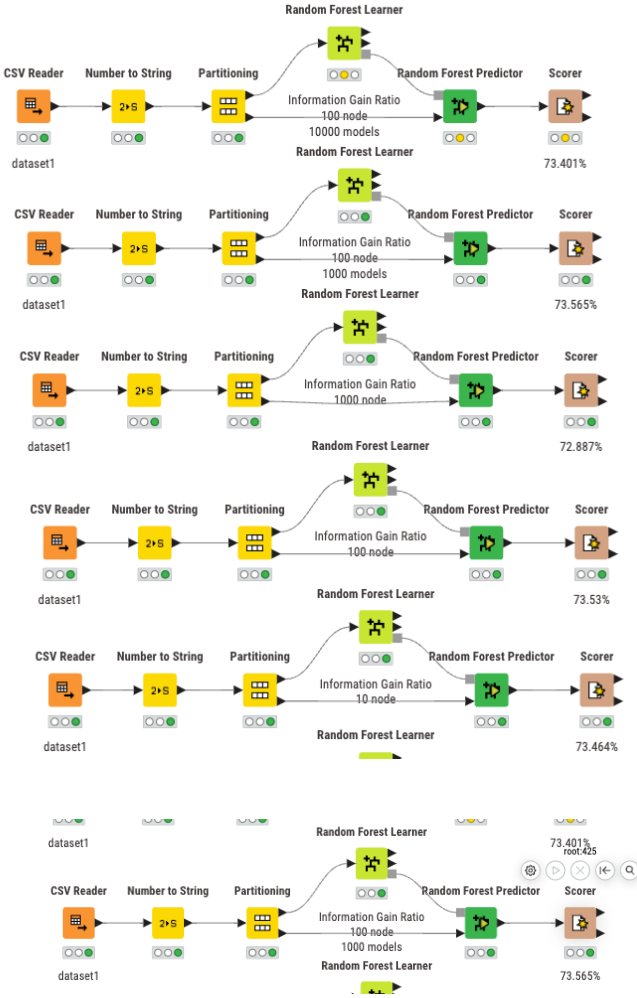
B. PRE-PROCESSED DATASET AND DIFFERENT METHODS



This figure applies the same Random Forest(information gain ratio) workflow to six increasingly processed datasets (dataset1–dataset6). The test accuracies lower steadily as preprocessing becomes more thorough: dataset1: **73.286**%--dataset5: **72.846**%.

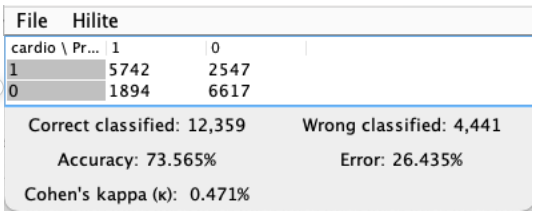
Which means that more data pre-processing results in lower accuracy. So maybe the non-preprocessed data(dataset1) would have a better result.

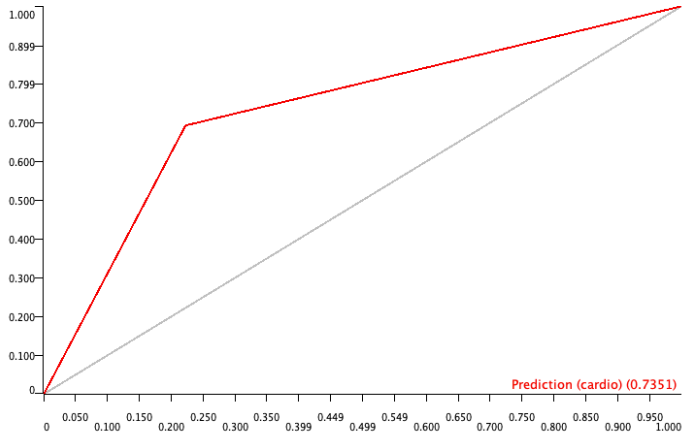
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS



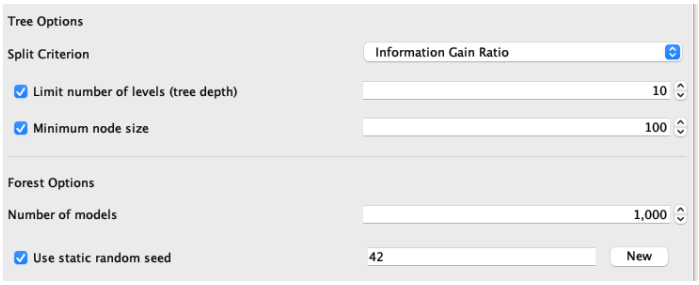
This workflow applies the same KNIME pipeline to **dataset1**, varying the number of nodes and models. The **100 nodes** and **1000 models** deliver the highest accuracy, while using a higher or lower number of nodes and models results in only marginally lower performance.

D. BEST RESULT SETTING





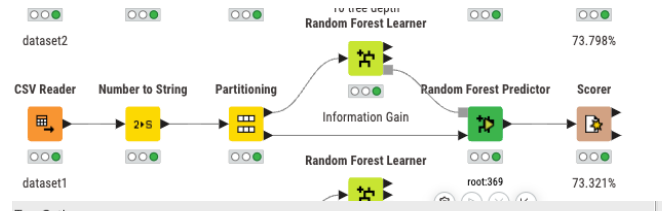
By experimenting with different parameter settings (tree depth, node size, models) in the random forest, we identified the best parameter setting in Information Gain Ratio (**100 node size, 1000 models**) that delivered the highest accuracy: **73.565%**. Compared to the default setting, this setting increased the AUC to **0.7351**.

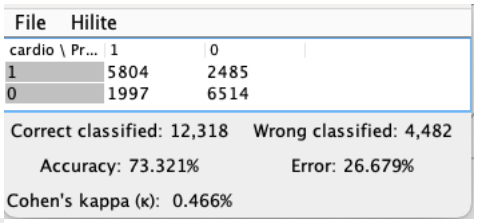


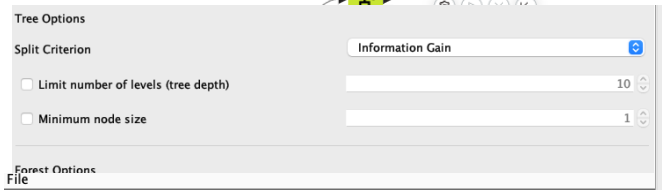
#	RowID	TruePositi...	FalsePosit...	TrueNegat...	FalseNega...	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's k...
1	1	5742	1894	6617	2547	0.693	0.752	0.693	0.777	0.721	?	?
2	0	6617	2547	5742	1894	0.777	0.722	0.777	0.693	0.749	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.736	0.471

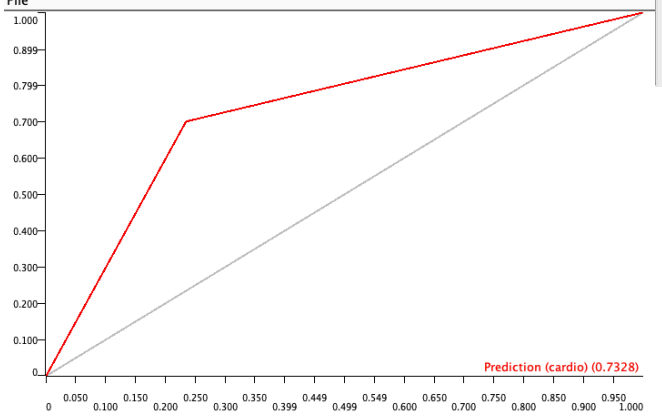
4.5.2 Information Gain

A. Default Dataset and Setting





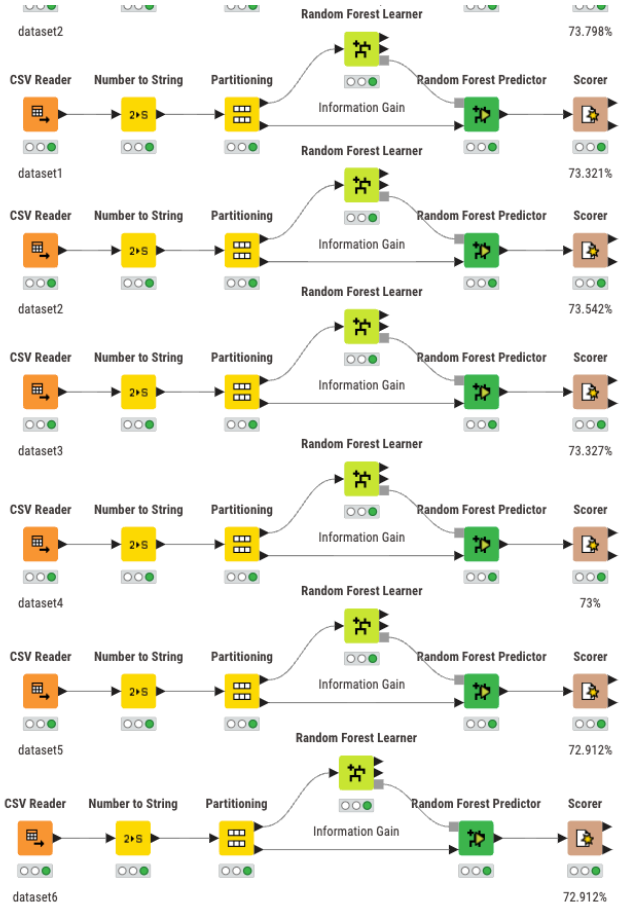




The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.7328**. This indicates the random forest (information gain) has a great ability to distinguish between patients with and without cardiovascular disease, better than random guessing, but leaving room for improvement.

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	1	5804	1997	6514	2485	0.7	0.744	0.7	0.765	0.721		
2	0	6514	2485	5804	1997	0.765	0.724	0.765	0.7	0.744		
3	Overall										0.733	0.466

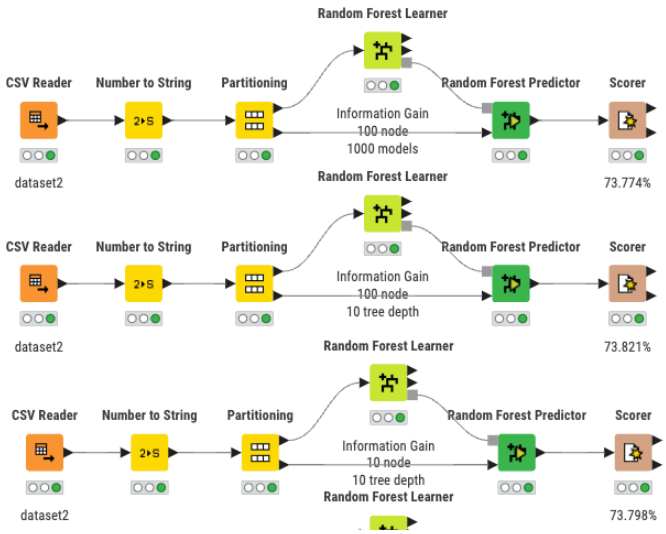
B. PRE-PROCESSED DATASET



This figure applies the same Random Forest workflow (information gain) to six increasingly processed datasets (dataset1–dataset6). The test accuracies lower steadily as preprocessing becomes more thorough: dataset1: **73.321%**--dataset5: 72.912%.

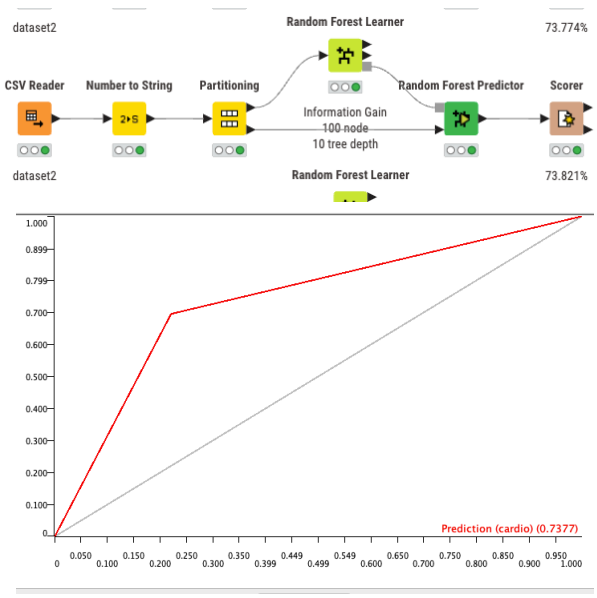
Which means that more data pre-processing results in lower accuracy. So, the non-preprocessed data(**dataset1**) may have had a better result.

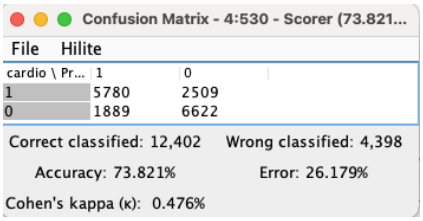
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS



This workflow applies the same KNIME pipeline to **dataset2**, varying the number of nodes, models and the limits of tree depth. The **100 nodes** and **10 tree depth** deliver the highest accuracy, while using a higher or lower number of nodes and tree depth results in only marginally lower performance.

D. BEST RESULT SETTING



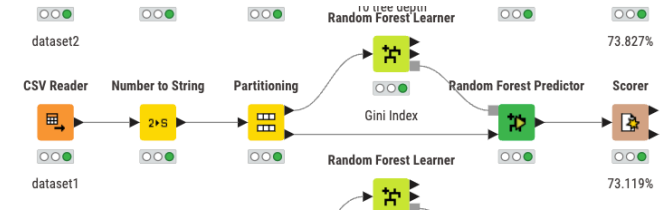


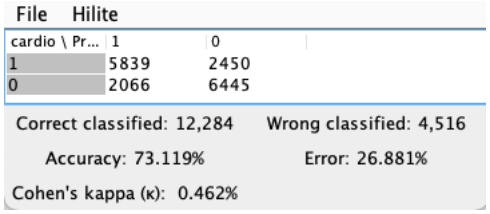
By experimenting with different parameter settings (tree depth, node size, models) in the random forest, we identified the best parameter setting in Information Gain (**100 node size, 10 tree depth**) that delivered the highest accuracy: **73.821%**. Compared to the default setting, this setting increased the AUC to **0.7377**.

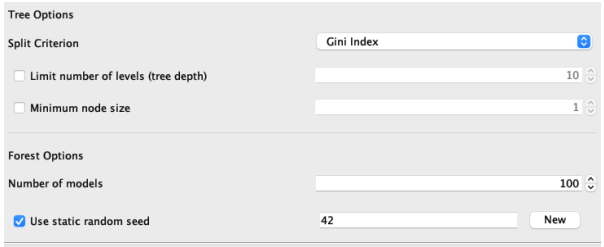
	#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
☐	1	1	5780	1889	6622	2509	0.697	0.754	0.697	0.778	0.724	?	?
☐	2	0	6622	2509	5780	1889	0.778	0.725	0.778	0.697	0.751	?	?
☐	3	Overall	?	?	?	?	?	?	?	?	?	0.738	0.476

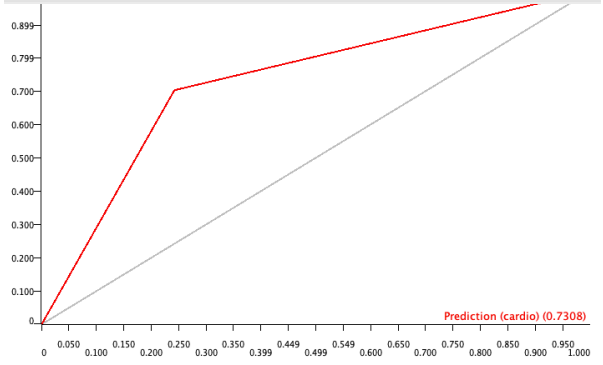
4.5.3 GINI INDEX

A. DEFAULT DATASET AND SETTING





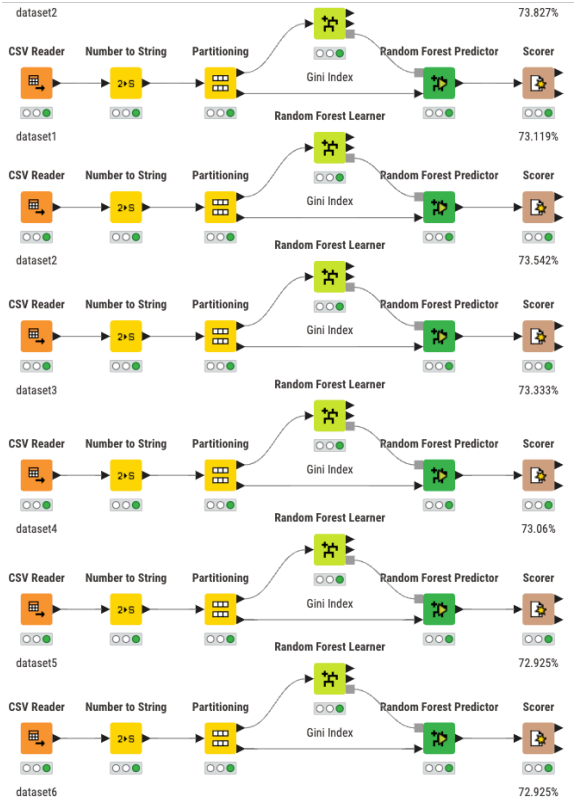




The red ROC curve lies above the grey diagonal (random chance), with an AUC of **0.7308**. This indicates the random forest (Gini index) has a great ability to distinguish between patients with and without cardiovascular disease, better than random guessing, but leaving room for improvement.

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	1	5839	2066	6445	2450	0.704	0.739	0.704	0.757	0.721		
2	0	6445	2450	5839	2066	0.757	0.725	0.757	0.704	0.741		
3	Overall										0.731	0.462

B. PRE-PROCESSED DATASET AND DIFFERENT METHODS

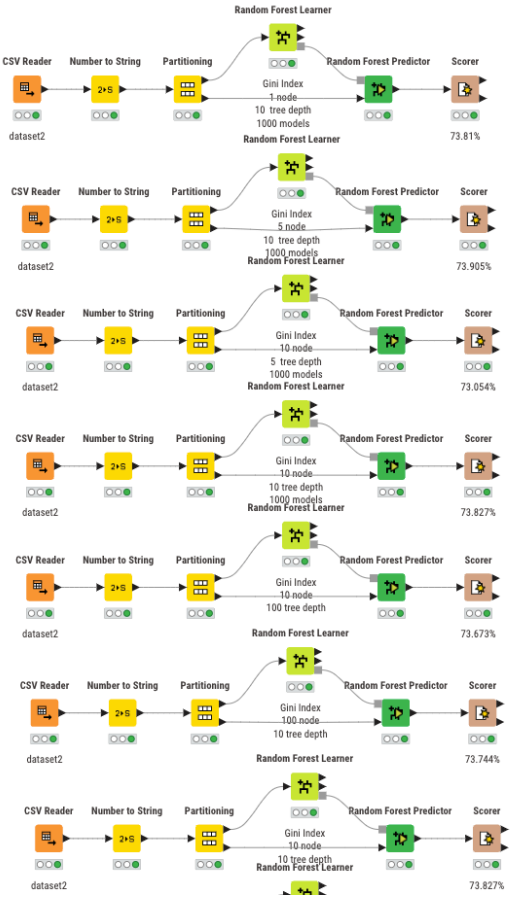


This figure applies the same Random Forest workflow (Gini index) to six increasingly processed datasets (dataset1–dataset6). The test accuracies lower steadily as preprocessing becomes more thorough: dataset1: **73.119**%-- dataset5: **72.25**%.

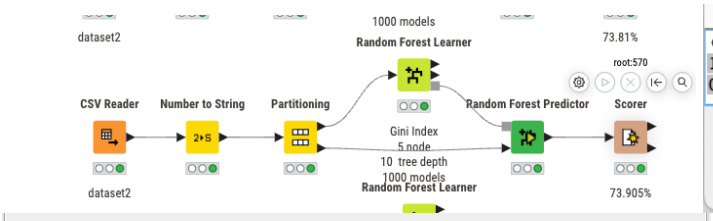
But the **dataset2** has the best accuracy: **73.542**%. Which means that more data pre-processing results in lower accuracy. So, the removal of ID data(dataset2) may have had a better result.

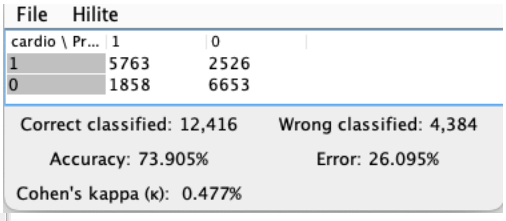
C. DIFFERENT EXPERIMENTS TO FIND THE BEST SETTINGS

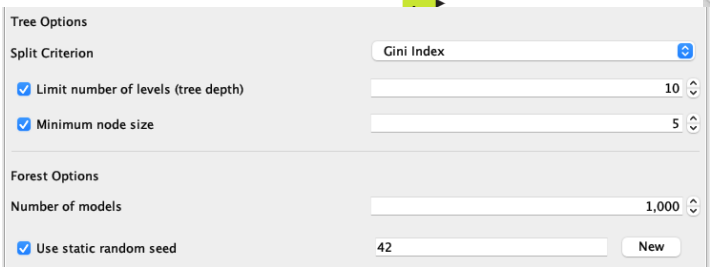
This workflow applies the same KNIME pipeline to **dataset2**, varying the number of nodes, limits of tree depth and models. The **5 nodes, 10 tree depth** and **1000 models** deliver the highest accuracy, while using a higher or lower number of nodes, tree depth, and models results in only marginally lower performance.

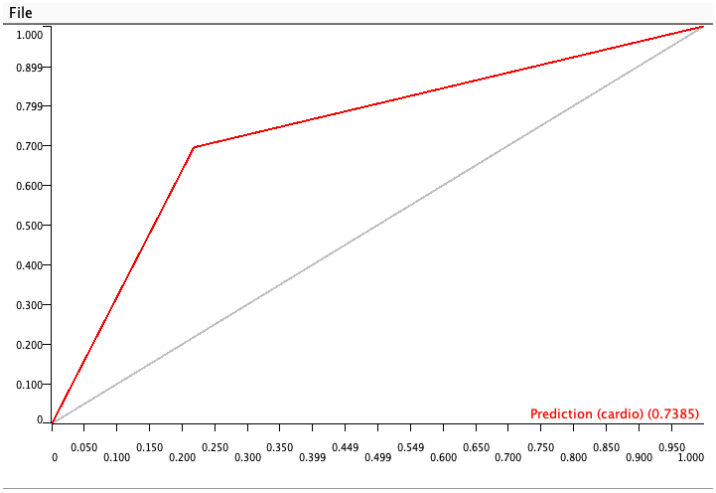


D. BEST RESULT SETTING





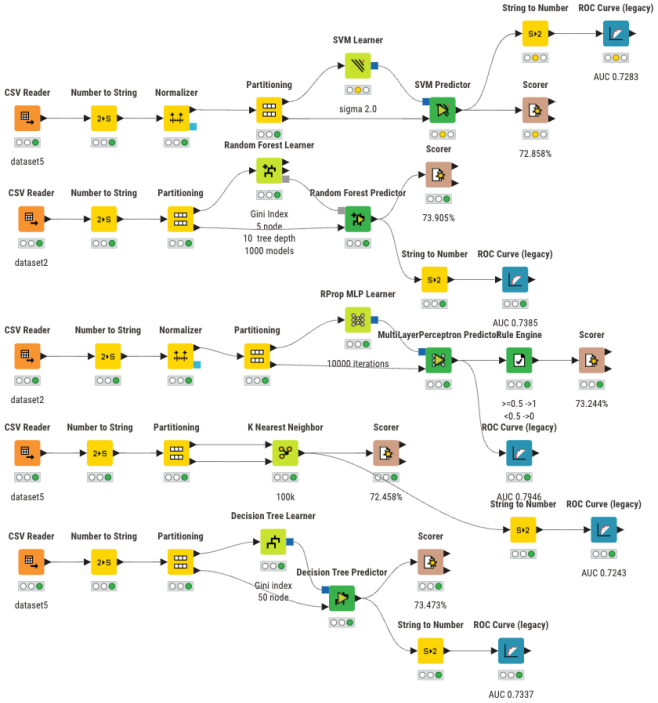




By experimenting with different parameter settings (tree depth, node size, models) in the random forest, we identified the best parameter setting in Gini Index (**5 node size, 10 tree depth, 1000 models**) that delivered the highest accuracy: **73.905**%. Compared to the default setting, this setting increased the AUC to **0.7385**.

#	RowID	TruePosit... Number (inte...	FalsePosi... Number (inte...	TrueNega... Number (inte...	FalseNeg... Number (inte...	Recall Number (dou...	Precision Number (dou...	Sensitivity Number (dou...	Specificity Number (dou...	F-measure Number (dou...	Accuracy Number (dou...	Cohen's k... Number (dou...
1	1	5763	1858	6653	2526	0.695	0.756	0.695	0.782	0.724	?	?
2	0	6653	2526	5763	1858	0.782	0.725	0.782	0.695	0.752	?	?
3	Overall	?	?	?	?	?	?	?	?	?	0.739	0.477

4.6 THE BEST CLASSIFIER



This image shows the best prediction accuracy results from five different machine learning models.

The **Random Forest model** achieves the **highest accuracy at 73.905%**, making it the top performer in terms of predictive accuracy.

The **Decision Tree model** follows with **73.473%**, and the **RProp MLP neural network** achieves **73.244%**.

The **Support Vector Machine (SVM)** model records an accuracy of **72.858%**,

The **K-Nearest Neighbor (KNN)**model has the lowest among the five, with **72.458%**. Overall, the Random Forest model provides the most accurate predictions in this comparison.

5. Comparison of the pros and cons of classifiers

Metric Group	Metric	Class	Decision Tree	KNN	Random Forest	MLP	SVM
Counts	True Positives (TP)	1	33.09%	32.78%	34.30%	34.48%	32.10%
		0	40.38%	39.68%	39.60%	38.76%	40.76%
	False Positives (FP)	1	10.47%	10.50%	11.06%	11.90%	9.41%
		0	16.06%	17.05%	15.04%	14.86%	17.73%
	True Negatives (TN)	1	40.38%	39.68%	39.60%	38.76%	40.76%
		0	33.09%	32.78%	34.30%	34.48%	32.10%
	False Negatives (FN)	1	16.06%	17.05%	15.04%	14.86%	17.73%

		0	10.47%	10.50%	11.06%	11.90%	9.41%
Derived	Recall (Sensitivity)	1	0.673	0.658	0.695	0.699	0.644
Metrics		0	0.794	0.791	0.782	0.765	0.812
	Precision	1	0.76	0.757	0.756	0.743	0.773
		0	0.715	0.699	0.725	0.723	0.697
	Specificity	1	0.794	0.791	0.782	0.765	0.812
		0	0.673	0.658	0.695	0.699	0.644
	F-measure	1	0.714	0.704	0.724	0.72	0.703
		0	0.753	0.742	0.752	0.743	0.75
Overall	Accuracy	Overall	0.735	0.725	0.739	0.732	0.729
Performance	Cohen's Kappa	Overall	0.468	0.449	0.477	0.464	0.457

Model	Overall Performance Rank (Accuracy/Kappa)	Strengths	Weaknesses	Key Characteristics / Notes
Random Forest	1 (Highest)	Best overall accuracy and Cohen's Kappa. Good Recall (0.695) and F-measure (0.724) for Class 1. Strong Recall (0.782) and F-measure (0.752) for Class 0. Relatively balanced performance between classes.	Not the absolute highest in every single per-class metric, but consistently performs very well overall.	Robust, generally the top-performing model. Suitable for general classification tasks where balanced performance is desired.
Decision Tree	2 (High)	Good overall accuracy and Kappa. Very good Recall (0.794) and F-measure (0.753)	Recall for Class 1 (0.673) is notably lower than for Class 0, and lower than Random	Good overall, particularly excels at identifying Class 0. Potentially more

		for Class 0. Good Precision for Class 1 (0.760).	Forest/MLP. Performance is less balanced than Random Forest.	interpretable than Random Forest or MLP (though not shown in metrics).
MLP (Multi-Layer Perceptron)	3 (Medium-High)	Good overall accuracy and Kappa. Highest Recall for Class 1 (0.699), meaning it's best at finding true Class 1 instances. Good F-measure for Class 1 (0.720).	Precision for Class 1 (0.743) is lower than Random Forest/Decision Tree. Recall for Class 0 (0.765) is lower than Random Forest/Decision Tree/KNN.	Strongest at detecting Class 1 instances. Might be preferred if missing a Class 1 instance is more costly.
SVM (Support Vector Machine)	4 (Medium-Low)	Highest Precision for Class 1 (0.773) - when it predicts Class 1, it's very likely correct. Highest Recall for Class 0 (0.812) - excellent at identifying Class 0.	Lowest Recall for Class 1 (0.644) - misses many true Class 1 instances Lowest Precision for Class 0 (0.697) Performance shows a significant trade-off between classes.	Best choice if the priority is extremely high confidence for Class 1 predictions (Precision) or exhaustive detection of Class 0 (Recall). Less balanced.
KNN (K-Nearest Neighbors)	5 (Lowest)	Decent Precision for Class 1 (0.757)Good Recall for Class 0 (0.791).	Lowest overall accuracy and Cohen's Kappa Low Recall for Class 1 (0.658). Low Precision for Class 0 (0.699) Generally outperformed by other models.	Simplest model conceptually. Its lower performance in this case suggests it may not be capturing the data's complexity as well as other models.

6. Reflection

This assessment provided a practical and insightful journey into the world of data mining, specifically in the context of predicting cardiovascular disease. It helped me to underscore several key learnings about the data mining process itself and offered valuable self-reflection on my approach to problem-solving.

6.1 Learnings about data mining

My primary takeaway is the critical importance of data preprocessing. The document consistently demonstrates that more intensive data preprocessing generally leads to better model accuracy. For instance, with the Decision Tree (Gini index), accuracy improved from 63.78% on unprocessed data (dataset1) to 68.43% on the most thoroughly preprocessed data (dataset6). Similar trends were observed for KNN, MLP, and SVM models. This highlights that the quality of data preparation directly impacts the ability of a classifier to uncover meaningful patterns.

Secondly, I learned that there's no one-size-fits-all classifier. Each model has its unique strengths and weaknesses. For example, while the Random Forest model achieved the highest overall accuracy (73.905%), the MLP was best at finding true Class 1 instances (highest Recall for Class 1 at 0.699), and the SVM had the highest precision for Class 1 (0.773), meaning when it predicted Class 1, it was very likely correct. This emphasises the need to select a classifier based on the specific goals of the problem, for instance, whether minimizing false negatives (like missing a high-risk individual) is more critical than minimizing false positives.

The process also illuminated the significant impact of parameter tuning. Default settings are rarely optimal. Experimenting with parameters like node size and tree depth in Decision Trees, the number of neighbors (k) in KNN, iterations and hidden layers in MLP, sigma in SVM RBF, or the number of models in Random Forest consistently led to improved performance, often substantially increasing accuracy and AUC values.

Finally, the assessment showcased the iterative and experimental nature of data mining. It's not a linear process but one of building, evaluating, refining, and repeating, using tools like KNIME to systematically test different approaches and settings.

6.2 Learnings from problem solving

This assessment revealed my appreciation for methodical experimentation. I found the process of systematically testing different datasets, algorithms, and parameter settings quite engaging. Following a controlled approach, as outlined in the methodology, helped in understanding the incremental benefits of each change.

I also learned the value of patience and persistence. Achieving the "best" model wasn't immediate. It required trying various configurations, analyzing results (like ROC curves and accuracy scores), and iteratively seeking improvements. For instance, the initial KNN model had an accuracy of only 55.577%, but through preprocessing and parameter adjustments, it eventually reached 72.458%.

Moreover, I discovered a certain creative satisfaction in data transformation. Taking raw data, like age in days, and converting it into a more interpretable or useful feature, like age in years or BMI, and then seeing how these changes affected the model's predictive power was quite rewarding. It felt like shaping the raw material to better reveal the underlying story of cardiovascular risk.

6.3 Approaching the problem differently Next Time

If I were able to do this assessment again, I would implement a few changes in my approach:

a. Deeper feature engineering earlier and further

While BMI was calculated and proved useful, I would explore more complex feature engineering from the outset. This could involve creating interaction terms between variables

(e.g., age and cholesterol levels, or smoking status and blood pressure) or exploring polynomial features, as these might capture more nuanced relationships relevant to cardiovascular disease.

B. SYSTEMATIC HYPERPARAMETER OPTIMIZATION

Instead of manually iterating through different parameter settings for each model, I would employ more automated hyperparameter tuning techniques. Tools like KNIME offer optimization loops, or if using Python, libraries like Scikit-learn provide functionalities like GridSearchCV or RandomizedSearchCV. This would allow for a more exhaustive search of the parameter space and potentially uncover even better configurations more efficiently.

C. INVESTIGATE ANOMALOUS PREPROCESSING EFFECTS

For some models, like SVM with a Polynomial kernel and certain Random Forest configurations, more preprocessing sometimes led to slightly lower accuracy on specific dataset versions. I would dedicate more time to understanding why these anomalies occurred. Perhaps certain transformations introduced noise or obscured patterns that these particular algorithms or configurations were sensitive to.

D. BROADER INITIAL ALGORITHM SCAN WITH BASELINE PREPROCESSING

While multiple classifiers were tested, I might start with an even broader range of algorithms after a robust baseline preprocessing. This initial scan could quickly identify promising candidates before diving deep into fine-tuning the top few.

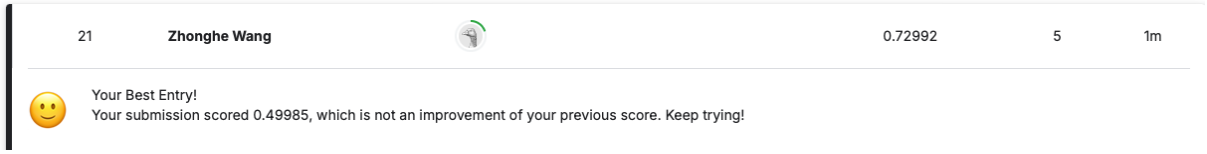
E. MORE FINE-GRAINED CONTROL ERROR ANALYSIS

Beyond overall accuracy and AUC, I would perform a more detailed error analysis earlier in the process for each model. This means looking closely at the confusion matrices to understand the types of errors being made (false positives vs. false negatives) for the "cardio" attribute and how different preprocessing steps or parameter changes affect these specific error types, which is crucial for a health-related prediction task.

By incorporating these changes, I believe the process could become even more efficient and potentially yield even more robust and insightful classifiers for predicting cardiovascular disease.

7. KAGGLE AND ORAL DEFENSE SCORER

7.1 KAGGLE SCORER



7.2 ORAL DEFENCE SCORER

The tutor Yoshiano Hartanto gave me 40 on the oral defence.

Note: Please give me some feedback about this report so I can improve my skill of writing report, thank you so much.