Introduction to Data Analytics
Assignment 3
Name: Yongyan Liu
Student ID: 14214338

Content

1. Data mining Problem

Data mining refers to the identification of data by classification to predict future trends. In this assignment, my task is to explore the correlation between the "qualified" attribute and other attributes, deal with missing data, clean the data, build an optimal prediction classifier to classify the qualified attributes by adjusting parameters and trying different models to obtain high prediction accuracy, and finally predict whether the house is qualified or not through the new house attribute.

Data mining also means the process of algorithmically extracting and predicting the useful information and knowledge present in a large amount of incomplete, noisy, fuzzy and random data. Therefore, in the prediction of "qualified" attributes, the house attribute data needs to be logically judged and tried, and the optimal combination and parameters are selected to train the model.

Based on my previous exploration of the house dataset, I will preprocess the data, including filter nodes, missing value nodes, auto-start nodes, normalization nodes, binning nodes, etc. I'll also use K Nearest Neighbor model(KNN), decision tree model, random tree model, ensemble tree model, Gradient Boosted Trees, SVM model, MLP Model the model to explore the best classifier.

At the same time, in order to consolidate my knowledge and try more rich data analysis tools, I used Python, Excel and Knime for data exploration in the process of analysis.

2. Input

There will be two input sets:

About training set: Assignment-HousingDataset.csv

About test set: Assignment-UnknownDataset.csv

One of which is the training set used to build the model, and the indicators of the model are preliminarily evaluated through verification. In the second dataset, I don't know if the house is qualified or not, I can predict the pass of the house by the best trained model, check the ROC graph to check the performance of the model, and finally upload the prediction result to kaggle.

3. Output

The output of the result will be a new CSV file with only two lines of attributes. The first-row attribute is ROW, and the second row will be the qualified prediction value, 1 if qualified and 0 if unqualified. I uploaded this new CSV file to kaggle to detect accuracy.

For the output results, I used the “Histogram” chart to visualize the distribution of qualified and unqualified. Use “Scorer” and “ROC curve” to view the correctness of model prediction and ROC performance curve. The closer the AUC is to a classifier in the ROC chart, the

better the performance. The abscissa represents the false positive rate, The response model converts negative examples into positive examples. The ability of the vertical axis represents the true positive rate, also known as sensitivity or recall rate, the ability of the response model to correctly classify positive cases into positive cases.

4. Data Preparation

4.1 Missing value

Data loss may be caused by human error or neglect, imperfect data collection mechanism, technical problems, etc. Excessive missing data can lead to results bias, low accuracy and reliability of analysis results, overfitting, low model performance stability, and data distortion.

Usually, I use mean or median or other filled values based on machine learning algorithms, or use binning to deal with missing values. However, in the process of data analysis, I realized that blindly filling in the exact values can lead to data distortion, obscuring the association of missing values with certain variables, misleading analysis, and increasing noise.

Therefore, before training the model, I first use 'Statistic' to look at the distribution of the data and analyze and determine the method of dealing with missing values. Figure 1 and Table 2 show the attributes and quantities with missing values, respectively.

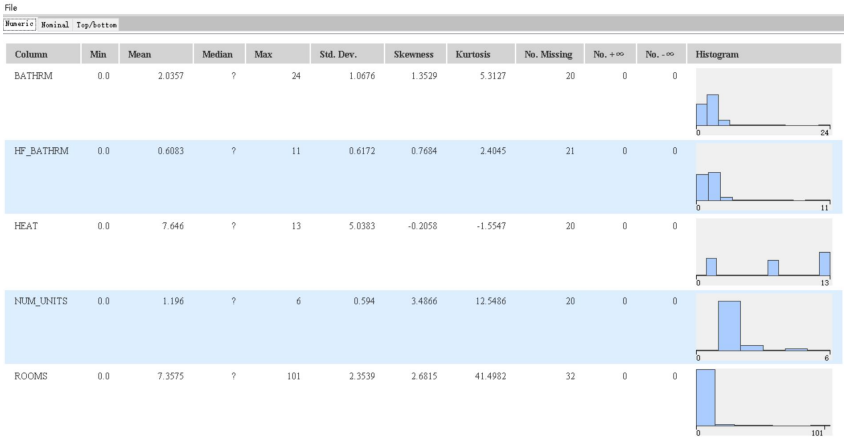


Figure 1 Statistic all attribute

Attribute	Number of missing values
BATHRM	20
HF_BATHRM	21
HEAT	20
NUM_UNITS	20
ROOMS	32
BEDROM	24
AYB	10
YR_RMDL	40473
STORIES	52

PRICE	13506
STYPLE	20
STRUCT	20
GRADE	20
CNDTN	20
EXTWALL	20
ROOF	20
INTWALL	20
KITCHENS	21
FIREPLACES	21

Table 2 Outlines

At the same time, we used the filter function in the Excel table to see if the attribute with the number of missing values is 20 related. The specific operation results are as follows:

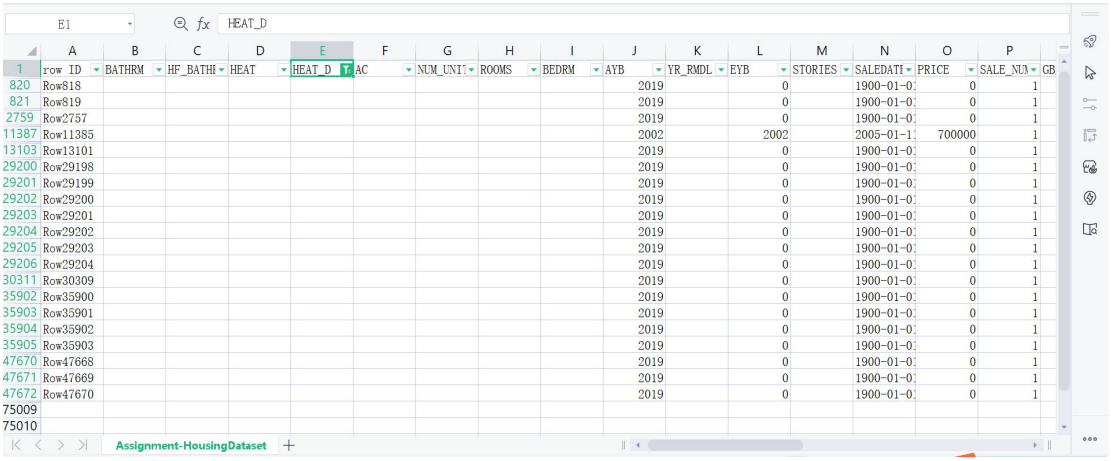


Figure 3 Find relationships between missing values

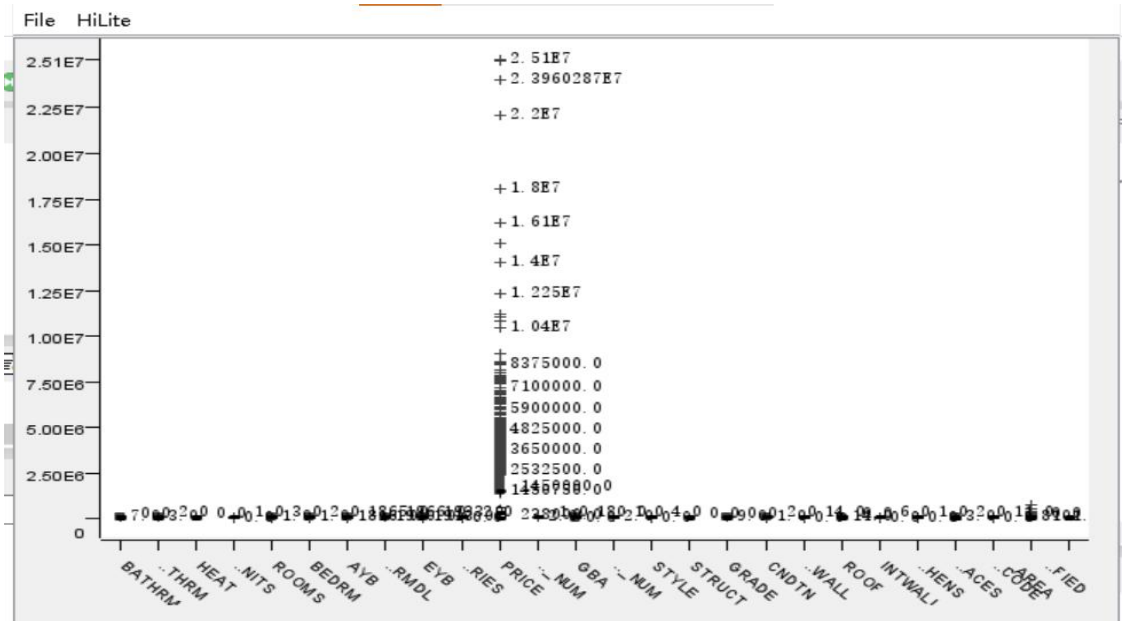
Conclusion, YR_RMDL and PRICE attributes have the most missing values. And there are 20 house attribute data missing the BATHRM, HF_BATHRM, HEAT, NUM_UNITS, ROOMS, BEDROM, YR_RMDL, STORIES, STYPLE, STRUCT, GRADE, CNDTN, EXTWALL, ROOF, INTWALL, KITCHENS, FIREPLACES attributes: at the same time, it can be considered that these 20 attributes may be incomplete or wrong due to some factors, so they need to be processed in data cleaning.

4.2 Outlines

Outliers are extreme data that exceeds the normal data expected by its type. The causes of outliers are similar to the causes of missing values, and data entry bias may be caused by data entry errors, incomplete human factors, incomplete collection mechanisms, and technical problems. The main impact of outliers on data analysis is that they have a significant impact on the mean variance and the statistical criteria for correlation coefficients and may reduce the fit.

Therefore, we can use visualization “Box Plot” and “Conditional Box Plot” nodes to

distinguish outliers that have a large impact on the data and the data. The specific operation results are as follows:



Conclusion and Reflection: Although some values are field values, not all of them are wrong values. For example, if the bedroom attribute is a value of 24, it may be real, but it may be too different from the data, resulting in a very strong accident, which may cause the model to be inaccurate and affect the accuracy. As well as LANDAREA number is 691817.0.

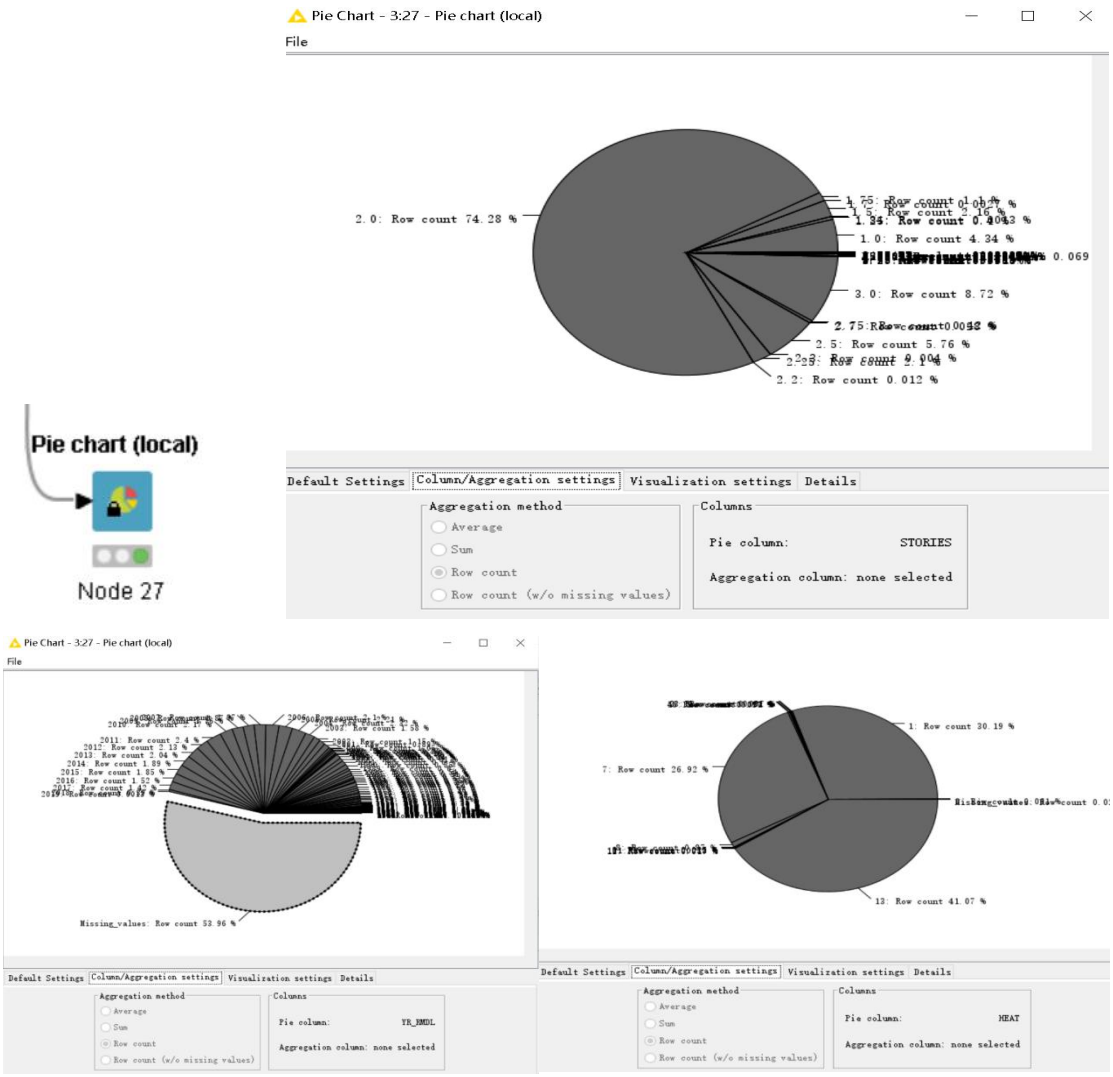
In this data, the outliers that can be prejudged are some values that are zero, such as A Y B and EYB years are extremely unlikely to be built in AD, and there are 20,740 data with zero price.

4.3 Data Exploration

4.3.1 Pie chart

The pie chart is to view the proportion of data distribution through visualization. We can quickly view the outliers and main ranges of some data through the pie chart.

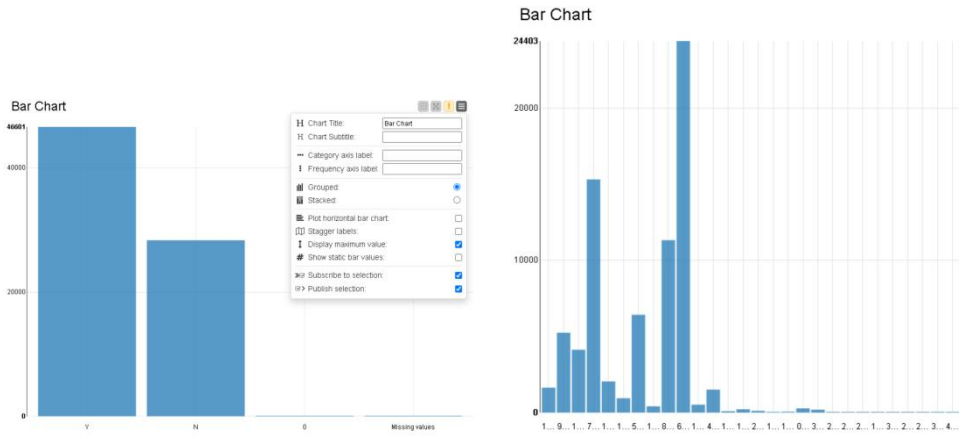
I selected the screenshots of some properties and pasted them below, and we can find that when the values are too many to be evenly divided or the angles are small, it is not conducive to visual viewing.



4.3.2 Bar chart

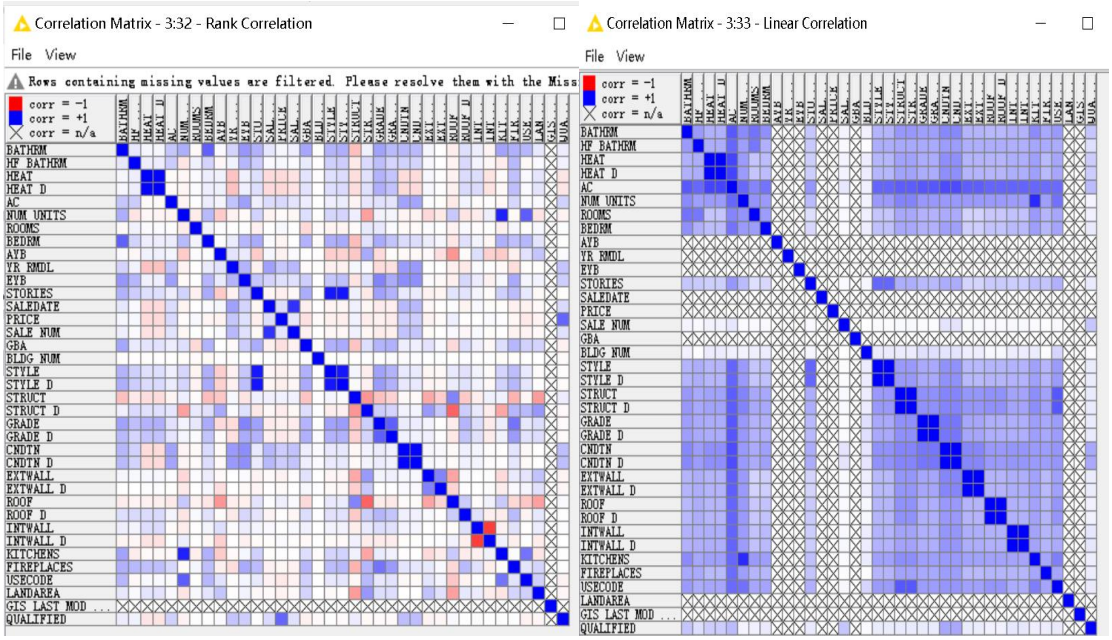
Bar charts can show comparisons and distribution types between different categories and organizations, as well as emphasize differences between categories. In some properties, a bar chart shows the distribution more clearly than a pie chart.

Through the histogram, we can identify that there are some attributes that can be removed, and we can adjust the nodes to make some oversized values delete.

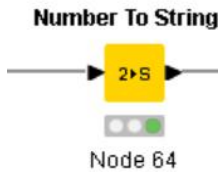
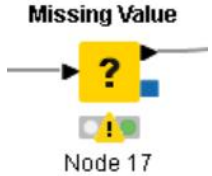


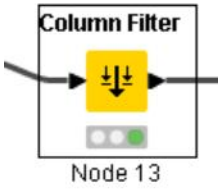
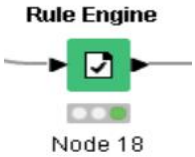
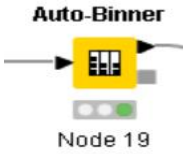
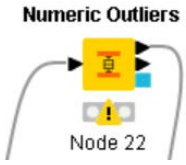
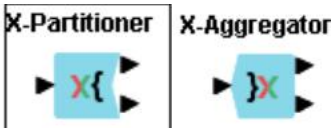
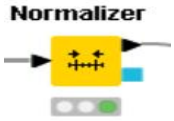
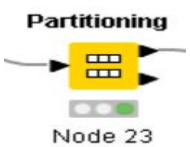
4.3.3 "Linear Correlation" and "Rank Correlation"

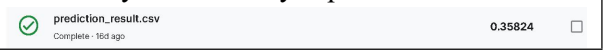
Both correlation nodes can be used to quantify the relationship between variables, with linear correlation more suitable for measuring linear relationships between continuous variables and ranking correlations more suitable for measuring monotonic or sequential relationships.
As can be seen from the following chart, except for a slightly linear relationship between price and quality, the correlation between other attributes and quality is not strong. But we can clearly see from the chart that Stories and style have a strong correlation, Kitchen and Num Unit also have correlation, and we can try to remove one of the attributes or both in the process of training the model to see the final accuracy.



4.4 Nodes used for data preprocessing

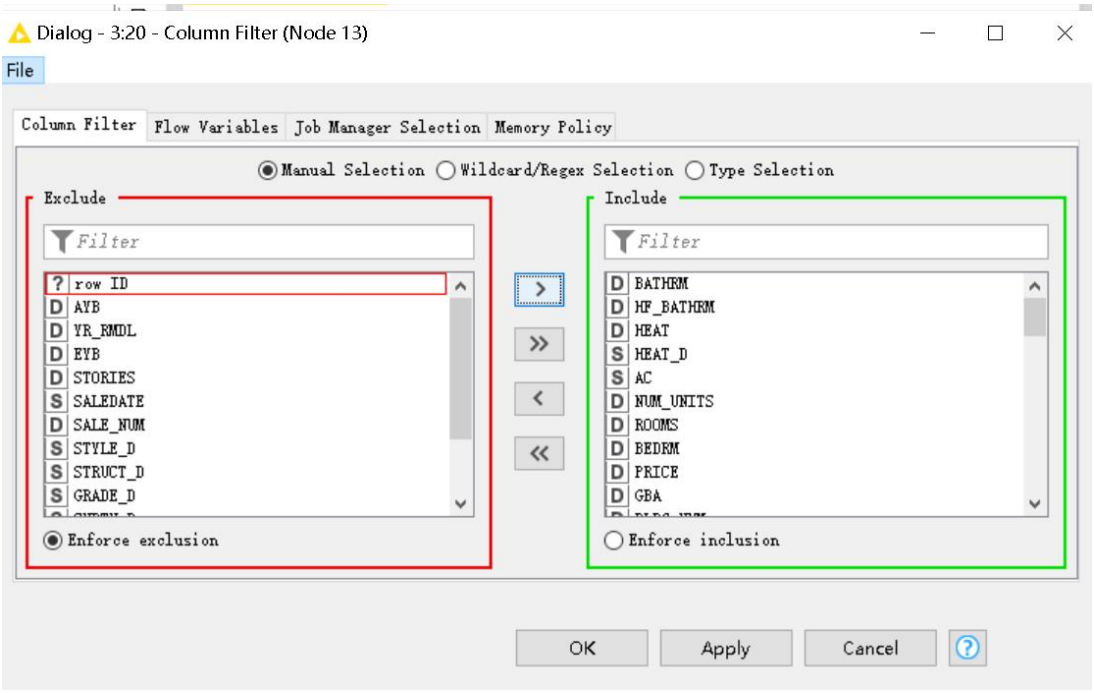
1. 	This node can convert numeric data into character data, and I use this node to process the target after classification as the prediction object of the classifier model.
2. 	Nodes are used to handle missing values, where I commonly use the median mean, and delete to try to train the model with the highest accuracy. At the same time, I set this node before the **type** conversion node in order to try different missing value filling methods for numeric types and string

	types, respectively.
3. 	This node can filter unnecessary columns in an attempt to improve the accuracy of the model. Some columns do not help much in the prediction of the model and may cause overfitting and model performance degradation. I'll describe in detail in the sections 4.4 below why some attributes are filtered.
4. 	This node can be used as a feature selection and generation node, we can customize a list of rules and try to match each row of data in the input list, and if this is met, its results will be added to the new column. This function can simply filter out the row data that meets our expectations.
5. 	This junction bins the data, which can improve the generalization degree of the model, avoid overfitting, and filter some outliers. We increase the speed of model training by sharing that can reduce the range of data, thereby reducing the complexity of the model.
6. 	This node can directly delete outliers and set the range. We set the multiple to 3 times. When it is greater than this multiple, deleting the data can control the data range within a certain range.
7. 	The function of this node is to realize K-fold cross-validation method. If the data is unbalanced, it may lead to low prediction ability of classification model and weak generalization ability. Therefore, cross-validation method is used to improve model accuracy.
8. 	This node normalizes the values of all columns, which can improve the difference between the index levels and reduce the expected value pair model. At this node, I try to use the maximum and minimum normalization, Z-score normalization and decimal normalization to try to tune the best model.
9. 	The function of this node is to divide the input data into two areas into training set and test. You can adjust the parameters of training and proportion to avoid overfitting. I usually set the proportion of training to 70%. Excessive proportion may cause Leading to over-fitting, too

	small may lead to insufficient training of the classifier model, the accuracy is not high, and the features cannot be captured. In the process of training, I tried to adjust the ratio to about 30. The accuracy rate can reach about 89%. However, if the result is put into kaggle, the result is about 35.82%. The possible reason is because of the training. Insufficient, resulting in the inability to accurately capture features. 

4.5 Data judgment

I finally chose to delete 'SALE_NUM' ， 'HEAT_D' ， 'STYLE_D' ， 'GRADE_D' ， 'STRUCT_D"CNDTN_D' ， 'SALEDATE' ， AYB ， EYB ,GIS_LAST_MOD_DTTM, 'EXTWALL_D'，'ROOF_D'，'INTWALL_D'， USER_CODE. The screenshot of the specific operation is as follows:



The reason is as follows:

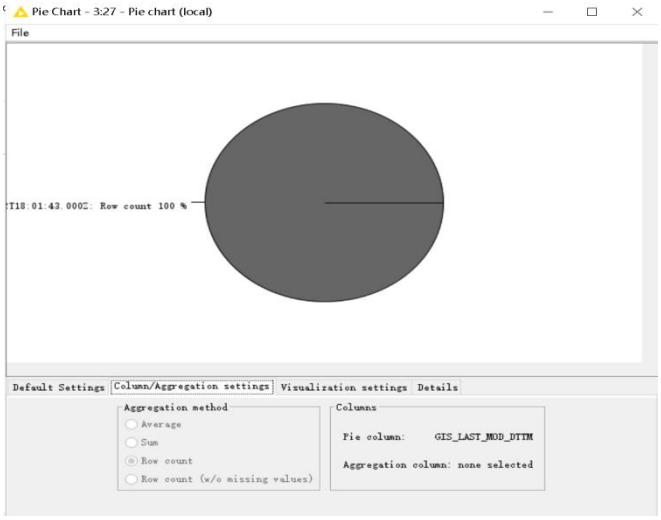
1. The attributes used as descriptions are deleted. For example, 'HEAT_D'，'STYLE_D'，'GRADE_D'，'STRUCT_D','CNDTN_D, 'EXTWALL_D'，'ROOF_D'，'INTWALL_D' . The role of these attributes in the data set is for explanation, but there is also a column of attributes with the same meaning but the type is numeric, so we deleted such data to reduce the burden of model operation, and the attributes using numeric values are more easy to quantify and analyze,.
2. Data with little meaning, such as AYB represents the earliest part of the construction of the main body of the building, EYB represents the number of years of

improvement of the main body of the building, SALE_NUM, USER_CODE

3. Delete attributes with more vacant values. The YR_RMDL proportion of vacant values is 54%. Therefore, whether we use the average value or the median method, it may affect the real value of the attribute. If you delete the empty value, it will affect other attributes.

4. Use the graph node to view the distribution of a certain attribute, and delete the data that is not meaningful.

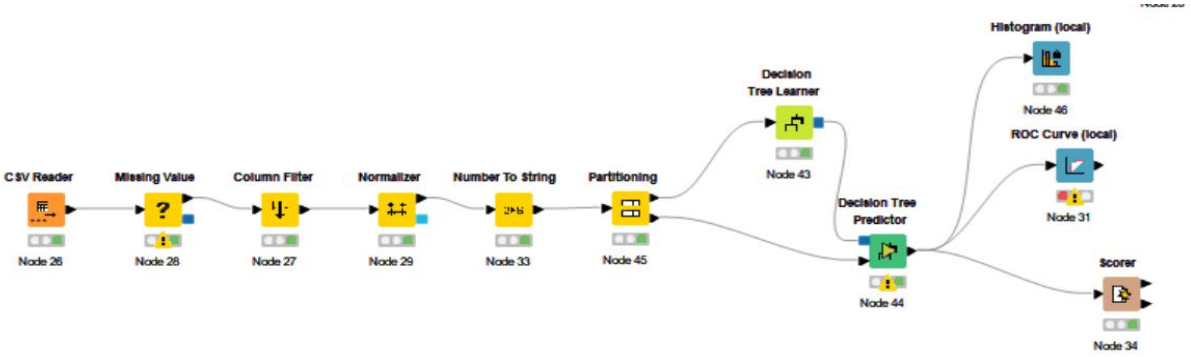
In GIS_LAST_MOD_DTTM attribute, all the values are 2018-07-22T18:01:43.000Z , so it does not have much effect on adjusting the model, but it will reduce the performance of the model, so this attribute needs to be deleted.



5. Classifiers

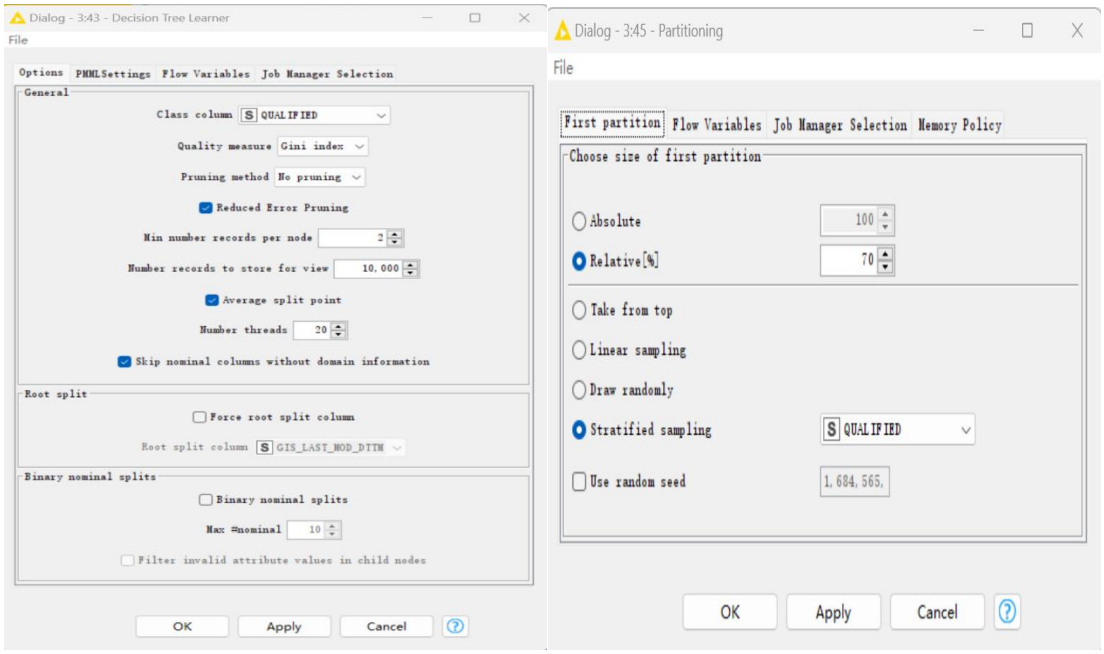
5.1 Decision Tree Classifier

5.1.1 Using the Decision Tree node

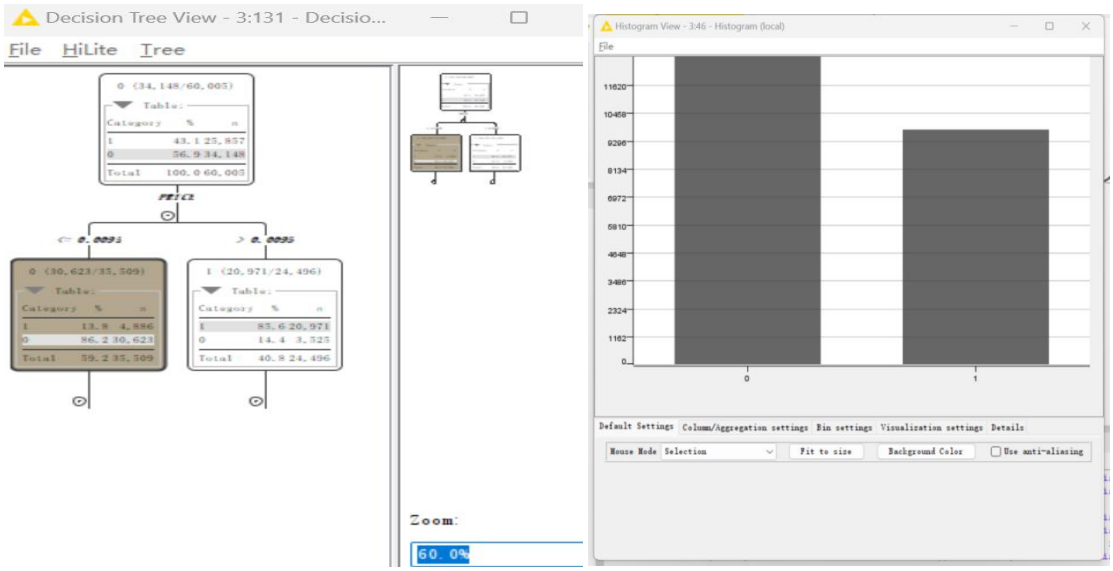


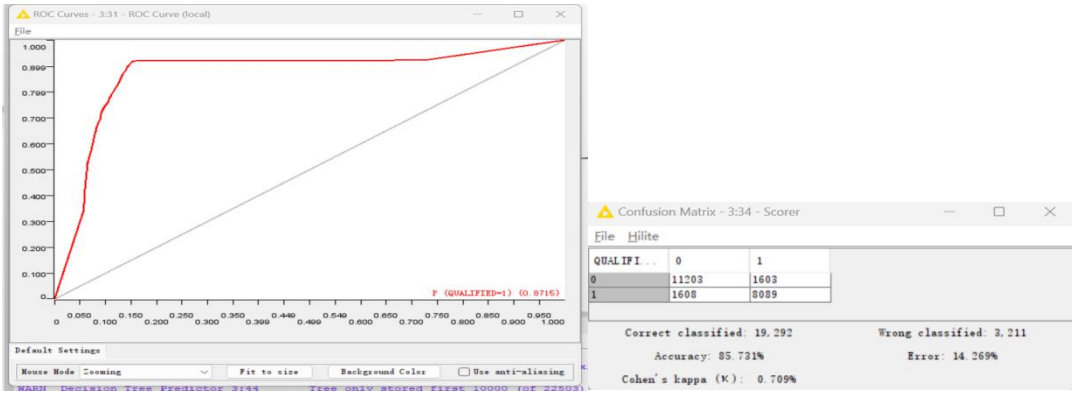
Decision tree refers to the construction of a judgment rule for each path from the root node to the leaf node. The characteristics of the internal nodes correspond to the conditions of the rules, and the class of the leaf nodes corresponds to the conclusion of the rules. Finally, a

decision analysis visualization graph like a tree structure is formed.



The partitioning setting is 70% of the training set, 30% of the test set. The Decision tree leaner setting quality measure is "**Gini index**" and pruning method is "**no pruning**". The Gini index refers to the index of data and purity, which can make the same category of samples easier to aggregate, with good robustness, intuitiveness, interpretability, low complexity and suitable for binary division. Since the output value of this prediction of whether the house is qualified or unqualified, only '0' and '1' belong to the binary division, so we first try "Gini index" parameter.

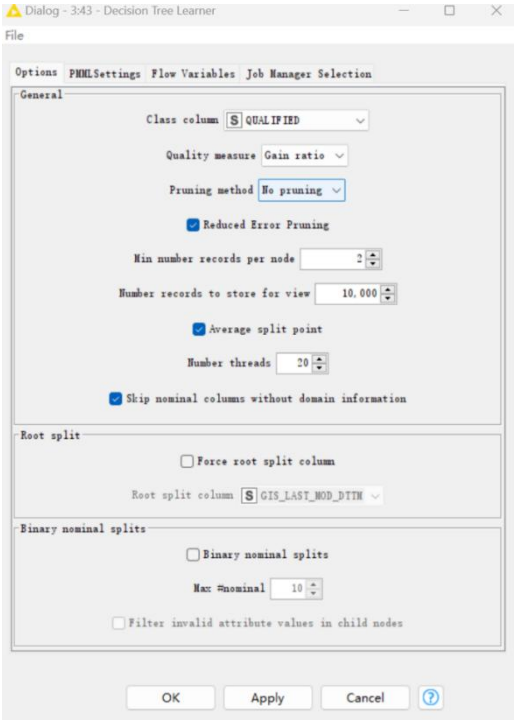




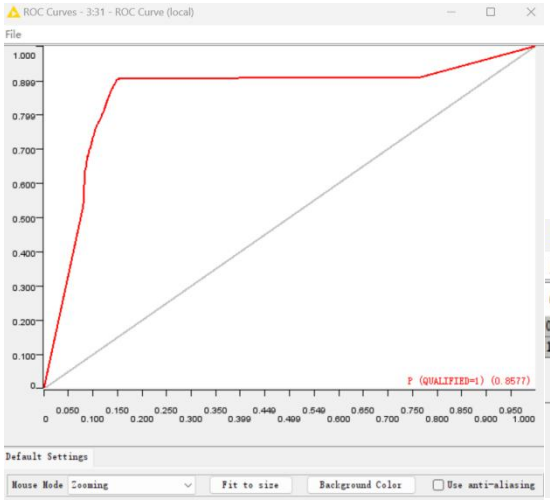
According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 85.731% , and the value of the ROC curve AUC is 87.15%. But we used the **historgram** to observe the distribution of the data and found that the distribution of the data was unbalanced, so we continued to try to adjust the optimal value of the parameter goods.

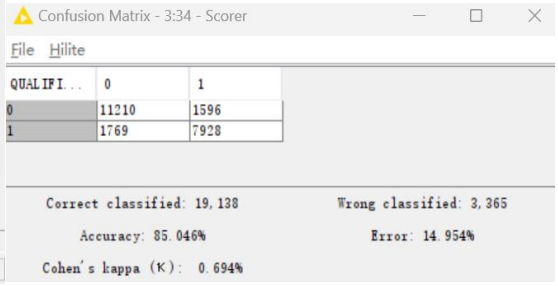
5.1.2 Adjust the quality measure

We try to adjust the parameters, as well as adjust the quality measure. The Decision tree leaner setting quality measure is "**Gain ratio**" and pruning method is "**no pruning**". Gain ratio considers the characteristic value distribution. If some features have large values, it may lead to a large information gain, so this method is adopted for normalization, which can eliminate the bias of data, improve fairness, and improve the consistency of feature selection.



We used the same data preprocessing method, only adjusting this parameter. The results are as follows:

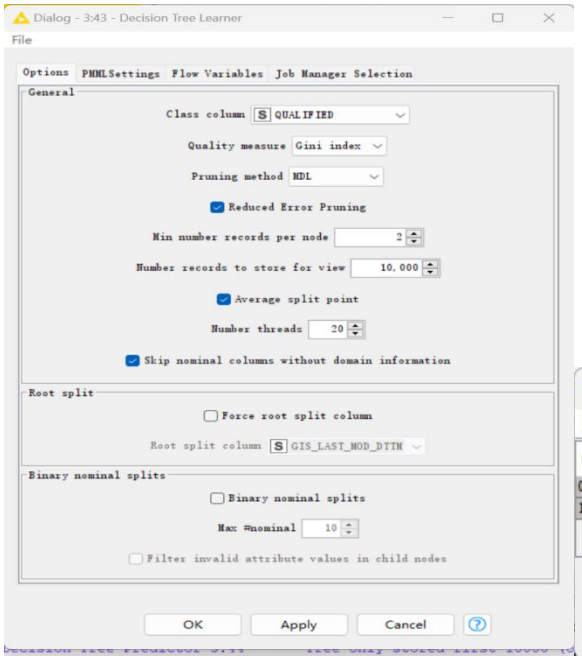


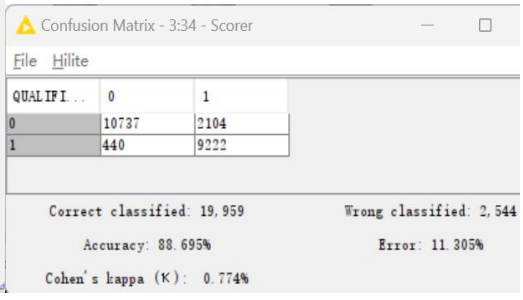


According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 85.046% , and the value of the ROC curve AUC is 85.77%.

5.1.3 Adjust the pruning measure

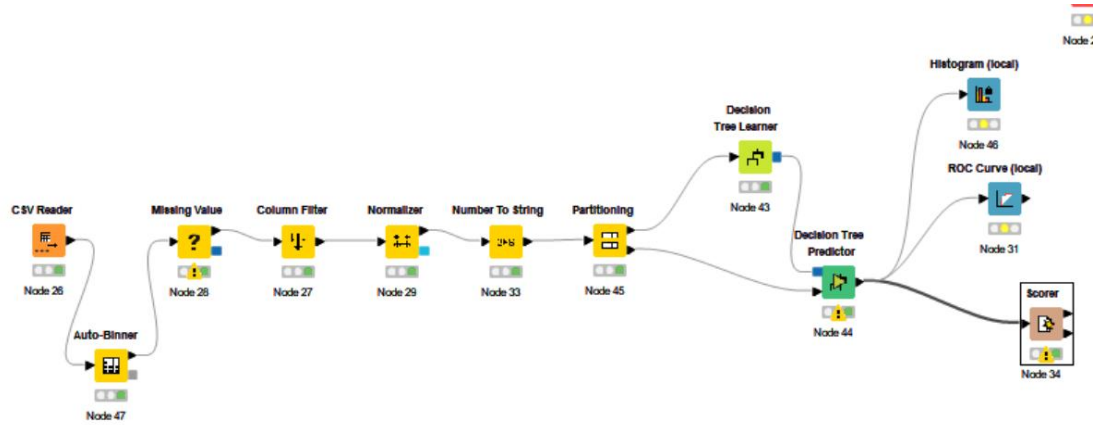
Without decision tree pruning, which refers to the division of certain leaf nodes in the construction of a decision tree, which is not statistically significant, it shows good visibility, and can show all the internal models and laws and potentially high accuracy. Therefore, we chose MDL pruning, which is not only to ensure the accuracy of the model and reduce the complexity of the model.



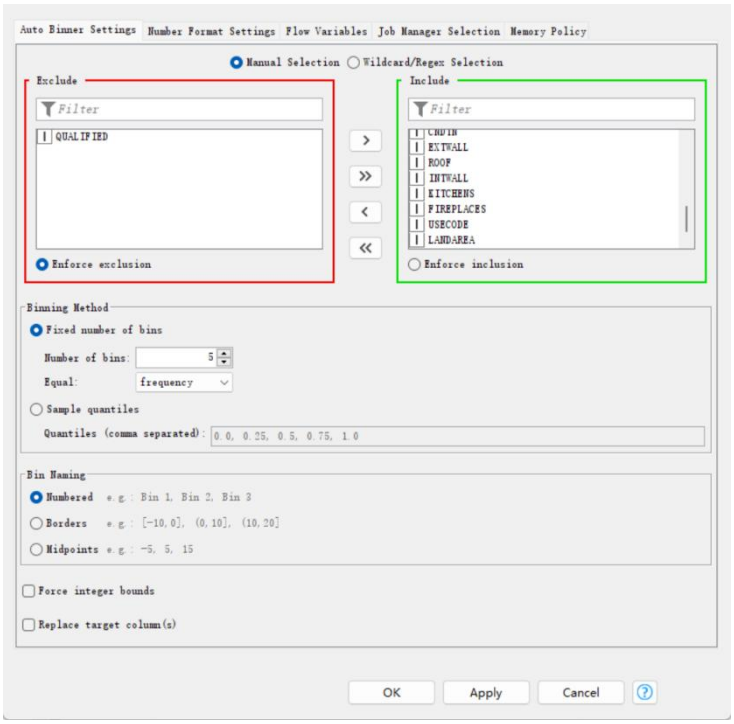


According to the confusion matrix, the prediction accuracy of this classifier is 88.695%. From this result, we can clearly see that the method of MDL pruning will improve the accuracy of the model and avoid over-fitting.

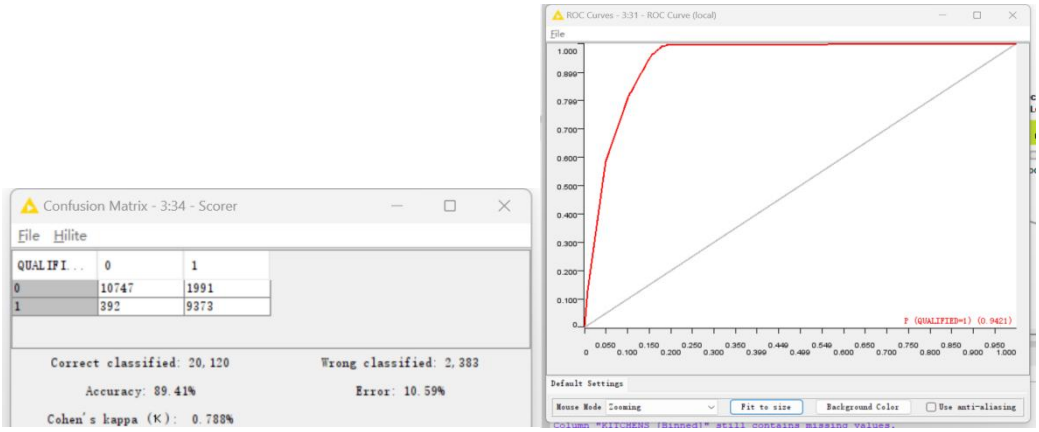
5.1.4 Use automatic binning to preprocess the classifier

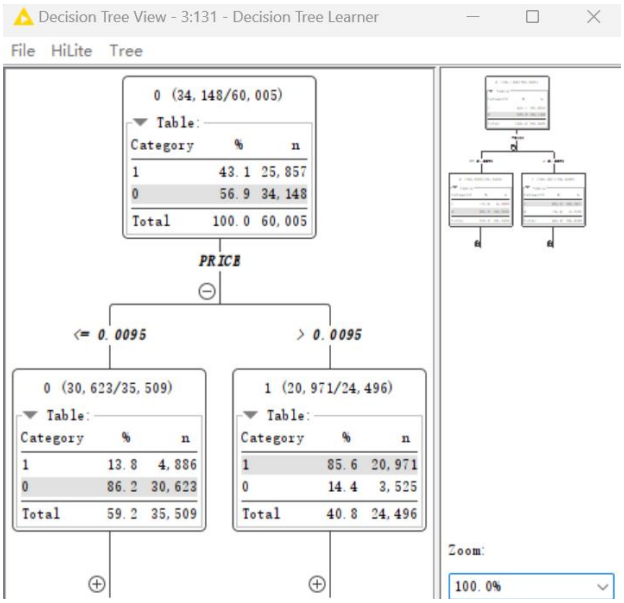


In the observation data distribution, we find that there are many missing values and outlines in the attribute, but this attribute is of great significance for predicting the quality of the house, so it cannot be deleted arbitrarily. The **automatic binning** function can classify outliers into adjacent bins, so that the values in each bin are more uniform, reduce the impact of outliers, and smooth the data to reduce data overfitting.



We set the number of boxes to five, and frequency measurement is used. Using width binning simplifies the model of data, dividing continuous variables into a finite number of discrete spaces. Too large correlation can lead to loss of information, too small box widths may over-fit.

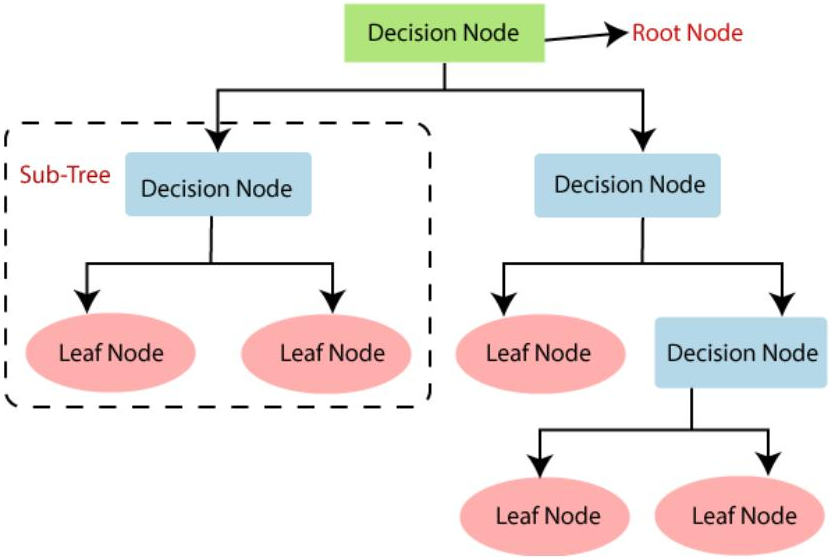




According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.41% , and the value of the ROC curve AUC is 94.21%. The **automatic binning** is used to improve the accuracy and performance of the model.

5.1.5 Conclusion

Decision trees use decision tree algorithms to segment data from the root to obtain the maximum characteristic information gain. It makes the samples in each leaf node belong to the same class, as shown in the figure for the decision tree algorithm (Arain, 2021):



By adjusting the parameters and comparing the ROC and the correct rate, we can conclude that the Gini index to gain ratio has little effect on the data accuracy rate, but the Gini index has a better AUC of ROC, and the guess is that the Gini index is more suitable for binary classification. The use of MDI pruning method can prevent model overfitting, improve the accuracy of the model and the performance of ROC. Adding automatic binning of nodes can eliminate the influence of missing values and outlier data and improve the accuracy of the model to a certain extent.

For decision trees, we discuss the decision tree models ID3, C4.5, CART, which have been used historically. ID3 uses Gain criterion as the node split attribute, C4.5 uses the gain ratio, and CART uses the Gini index. A comparison of the three models is attached below (Sharma & Kumar, 2016):

Features	*ID3*	*C4.5*	*CART*
Type of data	Categorical	Continuous and Categorical	continuous and nominal attributes data
Speed	Low	Faster than ID3	Average
Boosting	Not supported	Not supported	Supported
Pruning	No	Pre-pruning	Post pruning
Missing Values	Can't deal with	Can't deal with	Can deal with
Formula	Use information entropy and information Gain	Use split info and gain ratio	Use Gini diversity index

According to the accuracy of the model, the ROC curve and the description of the literature, the use range and comprehensive ability of CART are strong, which can reduce the calculation amount of the model and process continuous values, which is very convenient for dealing with binary classification problem prediction.

Score for best Decision tree:

Accuracy	89.41%
Precision	96.47%
AUC	94.21%
Error	10.59%
Recall	84.27%
Cohen's kappa	0.788%
F1 score	90.03%

Precision=(TP)/(TP+FP)

Error=1-Accuracy

Recall=TP/(TP+FN)

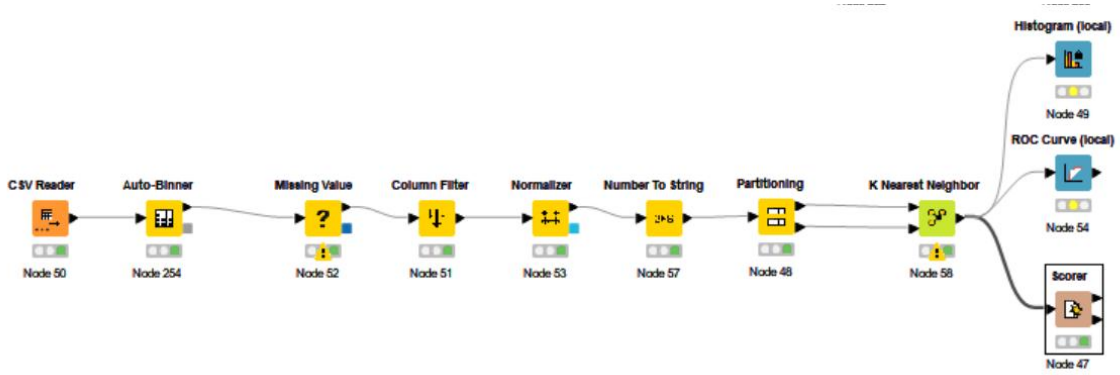
F1 score = 2 * (Precision * Recall) / (Precision + Recall)

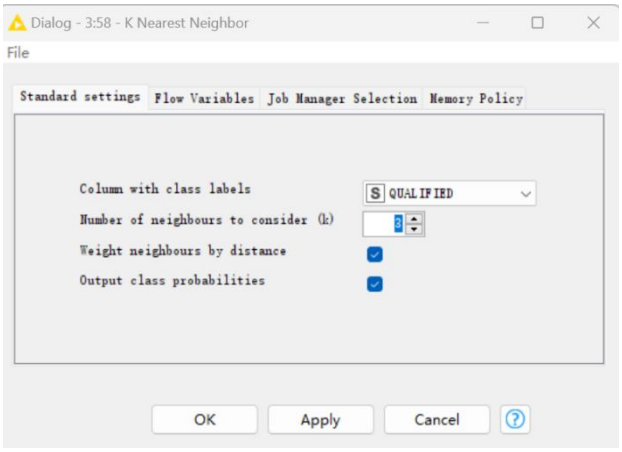
where TP represents the true example, TN represents the true negative example, FN represents the false negative example, and FP represents the false positive example. The confusion matrix is shown below (Wikipedia Contributors, 2019):

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

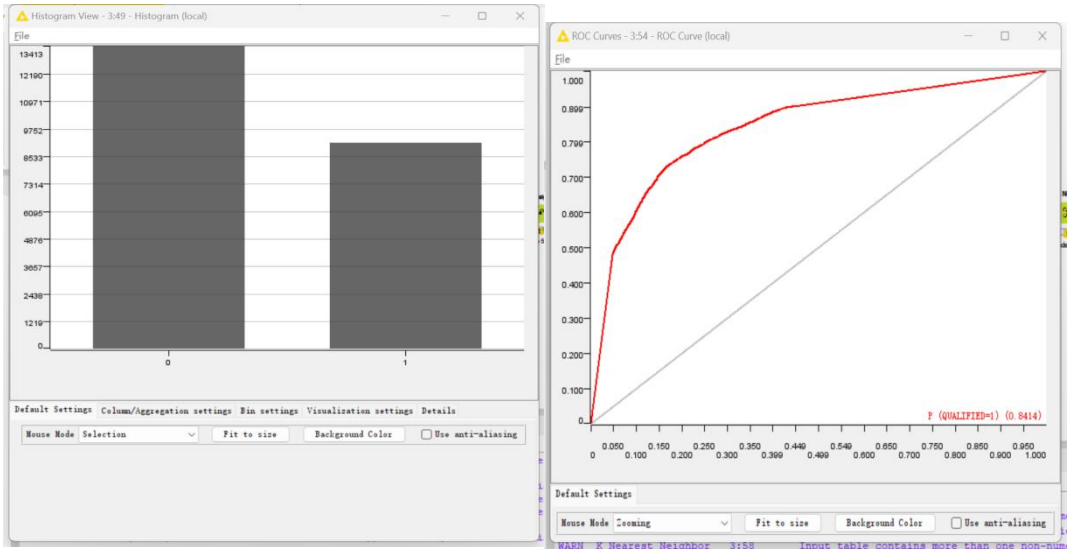
5.2 K Nearest Neighbor Classifier

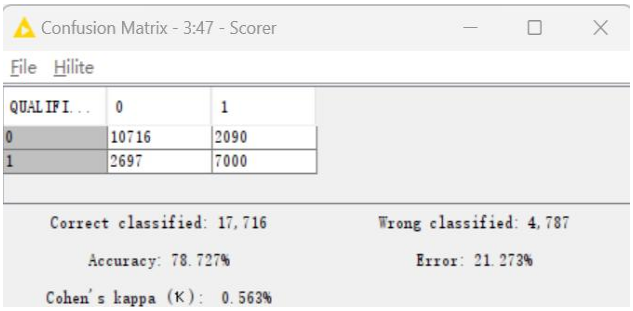
5.2.1 Using K Nearest Neighbor node





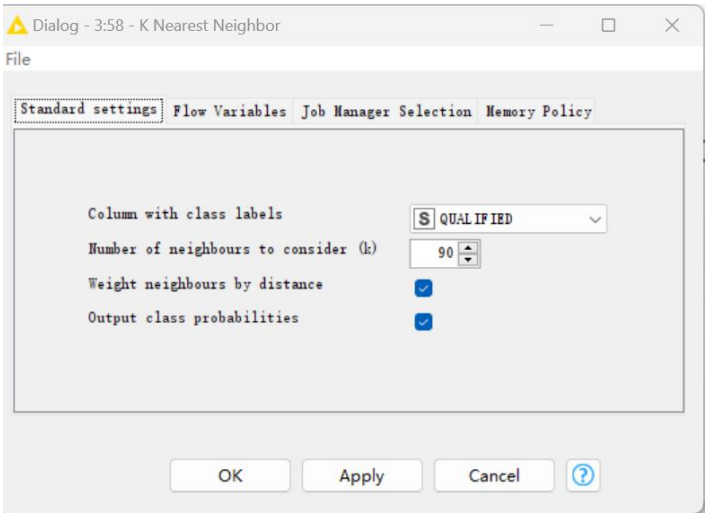
K nearest neighbor calculates the distance of new cases from each case, counts the most similar cases, and puts new cases into a category containing the nearest neighbors. Its principle is similar to the minority obeying the majority distributing data in adjacent taxa. The KNN algorithm is very suitable for nonlinear classification and multivariate classification problems, but it also has a huge amount of computation, lazy learning methods, low tolerance rate for training data, and low prediction rate for rare categories. We set the neighbor's parameters to 3. Here are the results of the model:



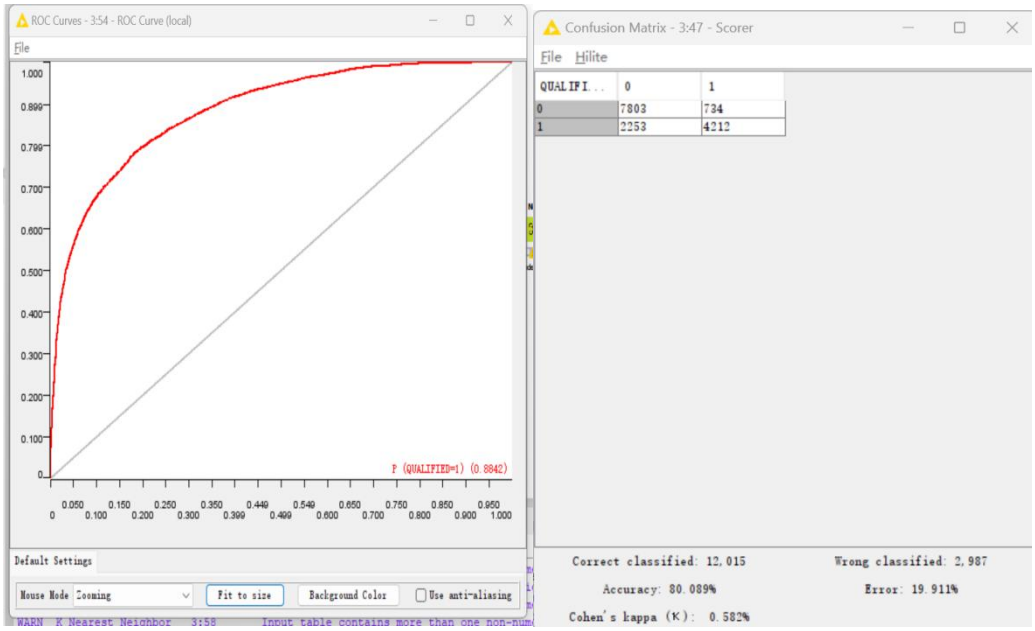


According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 78.727% , and the value of the ROC curve AUC is 84.14%. But we used the **historgram** to observe the distribution of the data and found that the distribution of the data was unbalanced, so we continued to try to adjust the optimal value of the parameter goods.

5.2.2 Adjust the number neighbor

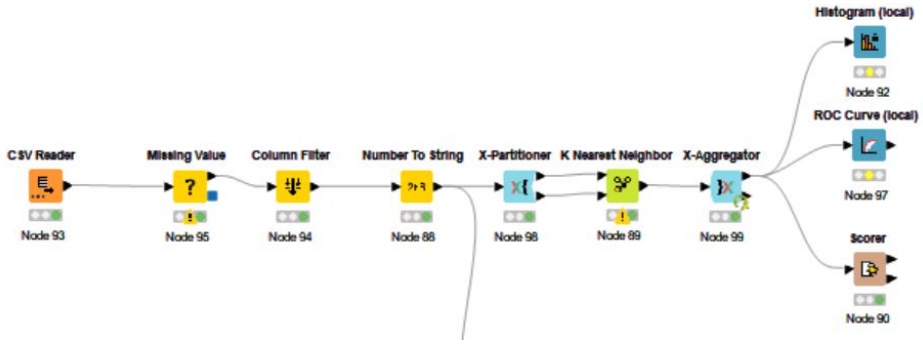


The selection of the appropriate number of neighbors determines the edge of the decision and better adapts to the local quantity characteristics. Smaller K-values are less computationally efficient and produce more complex decision boundaries, so they may cause the model to be more sensitive, overly rely on outliers, and not display well in smaller categories. So,we set the neighbor's parameters to 90. In the attempt, it got the highest accuracy.

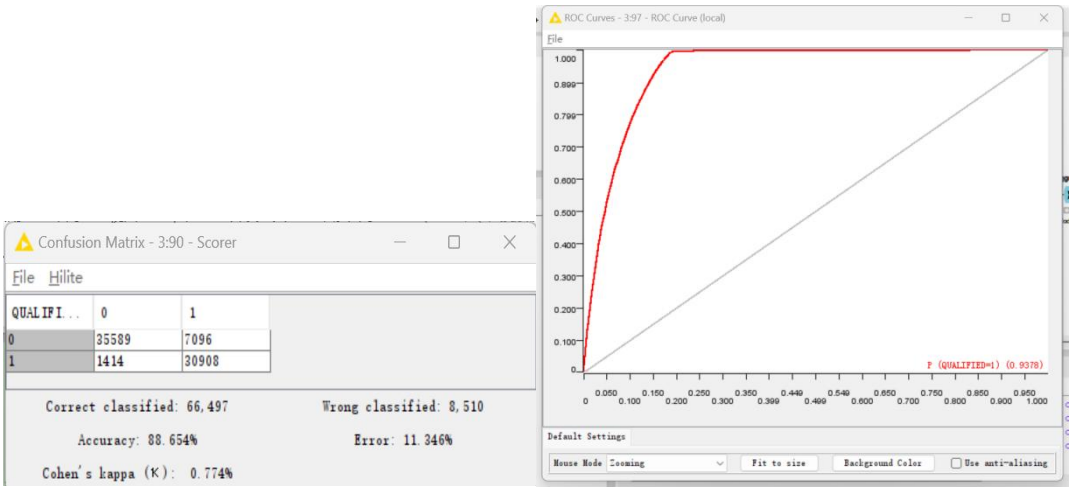


According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 80.089% , and the value of the ROC curve AUC is 88.42%.

5.2.3 Use x-partition and x-aggregator to preprocess the classifier

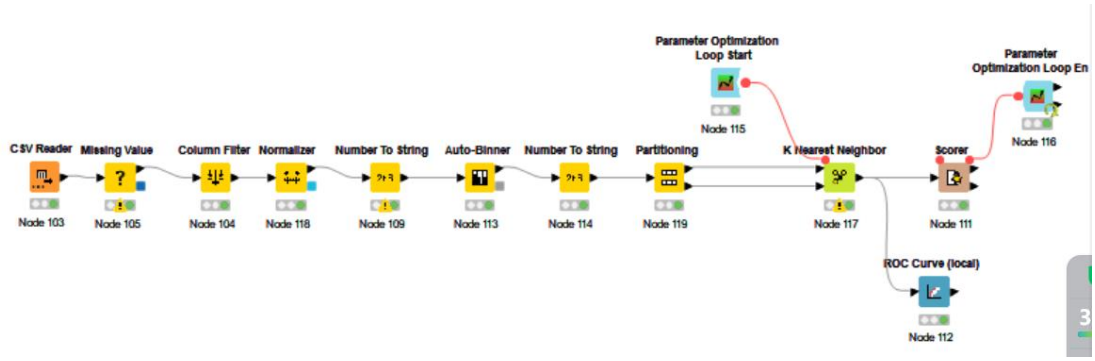


Use these x-partition and x-aggregator nodes to split and process data in parallel, which can facilitate the calculation of data, statistically dispersed data is called a global result, divide computing tasks into sub-tasks for merging and summarizing, and facilitate data storage and retrieval. The application of these x-partition and x-aggregator nodes often improves the computational efficiency of the model as well as large data processing.



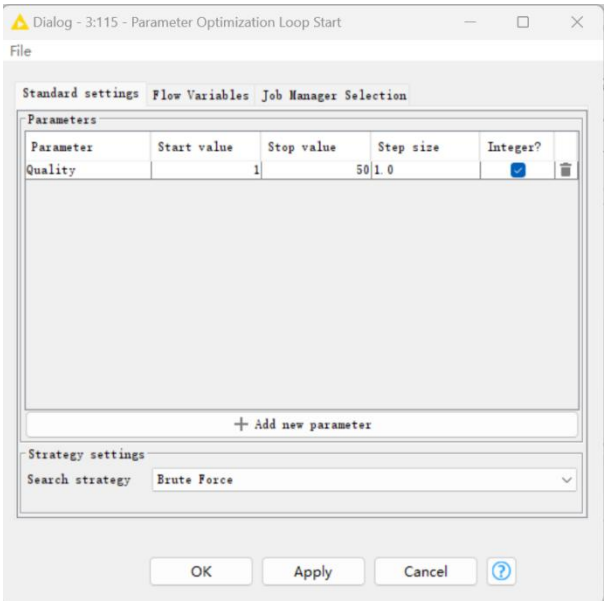
According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 88.654%, and the value of the ROC curve AUC is 93.78%. Compared with the model without this node, the accuracy rate and AUC of ROC are higher.

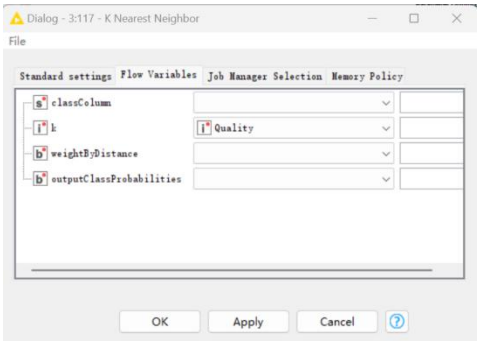
5.2.4 Use Parameter Optimization to preprocess the classifier

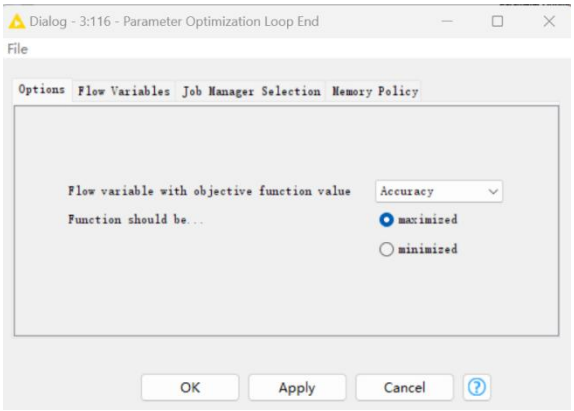


Parameter optimization is the process of searching only for the best parameter value, and the node can run multiple times to obtain the best accuracy parameter, improve the performance

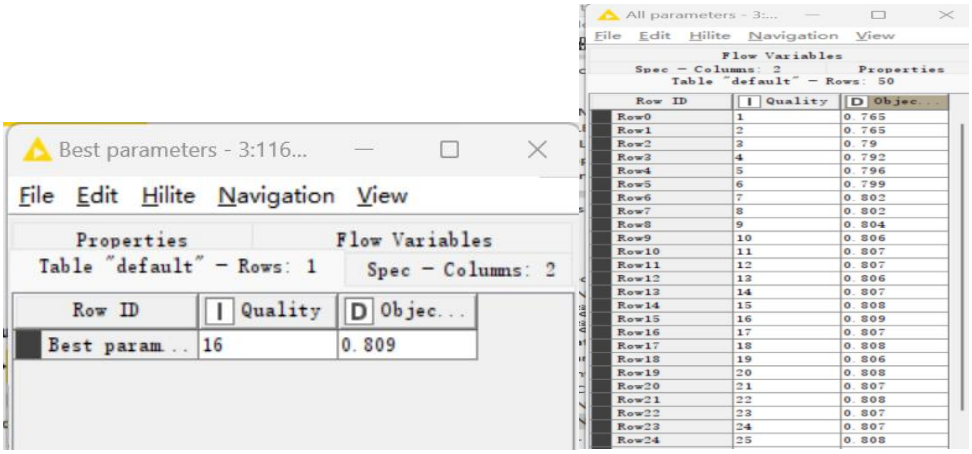
of the model, and reduce overfitting, and reduce labor costs and interference because the parameter adjustment process is automated, repetitive operation.







We set the stop parameter to 50 with a step size of one. Setting flow variable with objective function value is “Accuracy” and function should be maximized. After adjusting the relevant parameters, the following is the output result:

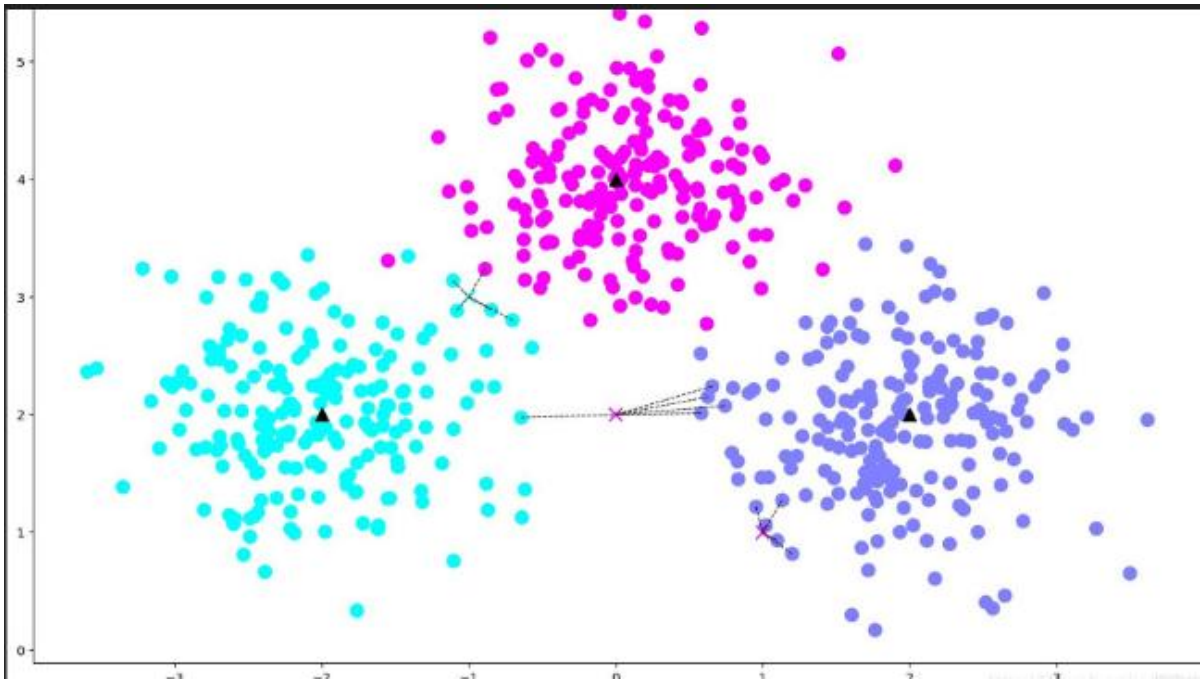


Based on the results we can see that the best parameter is 16 with an accuracy of 80.9, and we get the best result based on multiple attempts. We can also see the accuracy from zero to 50

from all the parameter plots, we can see that the parameter and the accuracy are not linear, and using this tool can reduce the time of multiple attempts to get the best value.

5.2.5 Conclusion

The KNN model does not need to adjust too many parameters, is easy to operate, and is suitable for an automatic classification supervision algorithm for a large number of sample areas. It can use the limited number of adjacent samples around to classify overlapping sample sets. The model of the KNN algorithm shown below (K-Nearest Neighbor (KNN) Classifier - Wolfram Demonstrations Project, n.d.):



In the exercise, we tried to adjust the number of neighboring neighbors of the KNN, and after trying, we found that 90 is a very suitable parameter, he has good robustness, dilutes the influence of noise and outliers, provides more information for most categories.
From the area of AUC in the ROC curve, it can be seen that the performance of the model can be improved by using x-partition and x-aggregator.

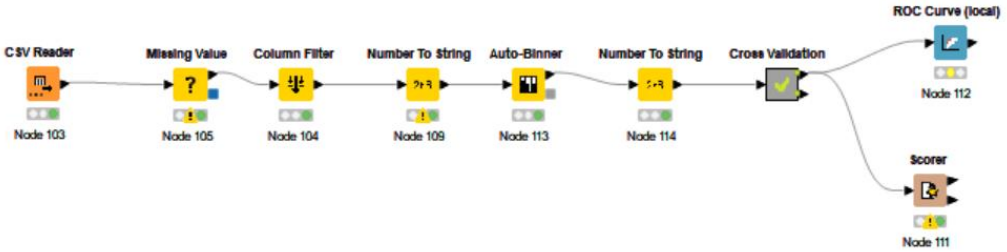
Score for best Decision tree:

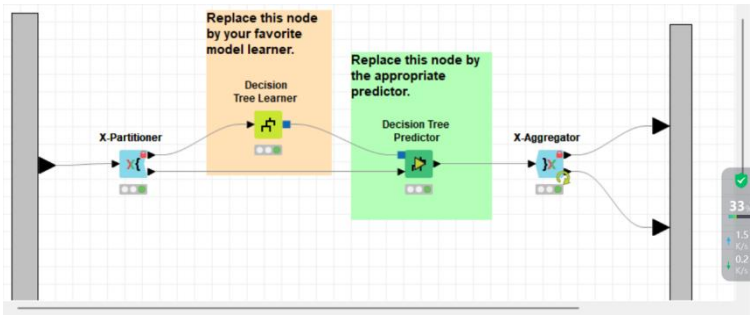
Accuracy	88.654%
Precision	96.16%
AUC	93.78%
Error	11.346%
Recall	83.36%
Cohen's kappa	0.774%
F1 score	89.22%

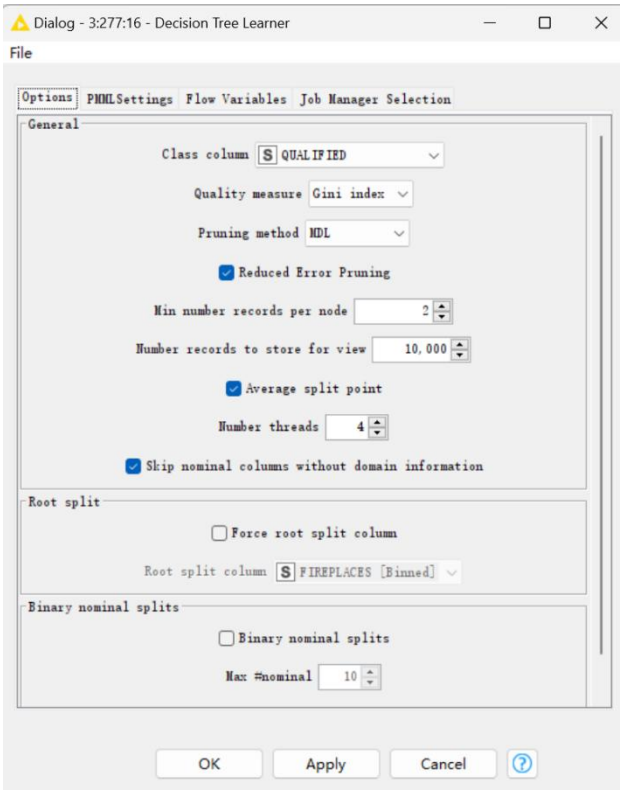
5.3 K-Fold Cross Validation

5.3.1 Using Decision Tree in Cross Validation

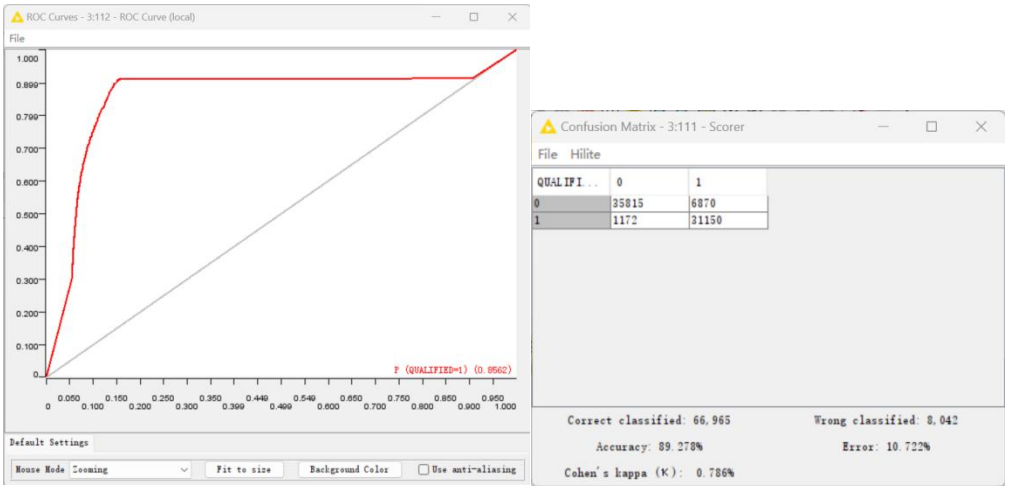
K cross is a model evaluation method, data and divided into subsets of K values and equal size, each time select one of the subsets as the validation set and the remaining subsets as the training set, repeat the process K times, so that each subset has the opportunity to become a test set, the model makes full use of the data to improve the confidence of the model, by adjusting parameters to compare the model performance in different configurations, can prevent and detect overfitting.







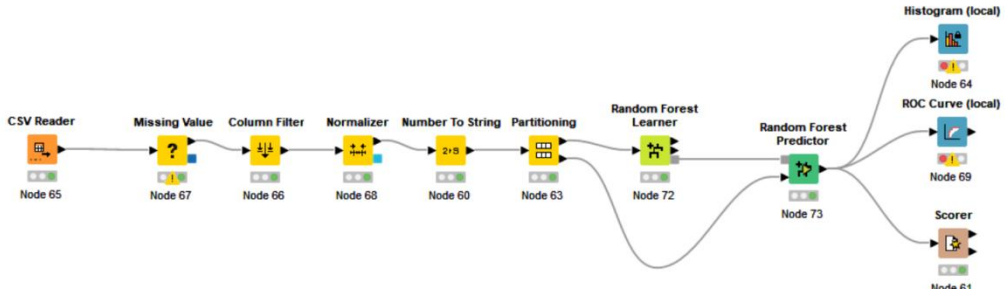
Random sampling was used in the cross movement and the number of validations was 10 according to my test exercises, which would have resulted in better accuracy.



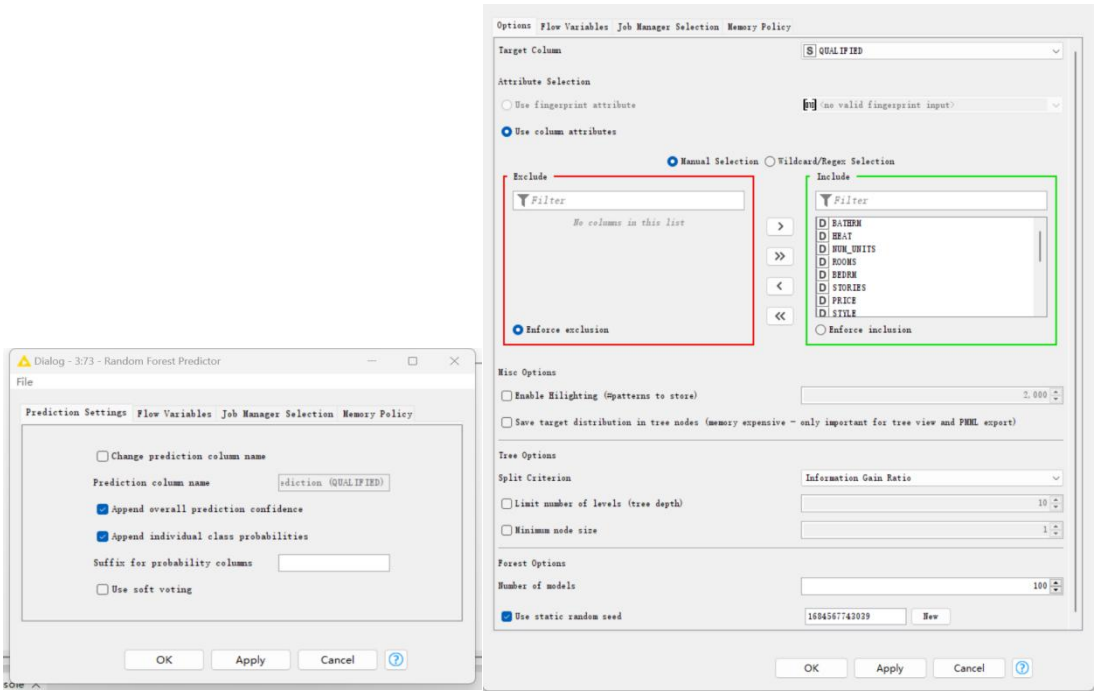
According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.278% , and the value of the ROC curve AUC is 94.23%. We observe that the area of ROC using the crossover algorithm is not optimal.

5.4 Random forest

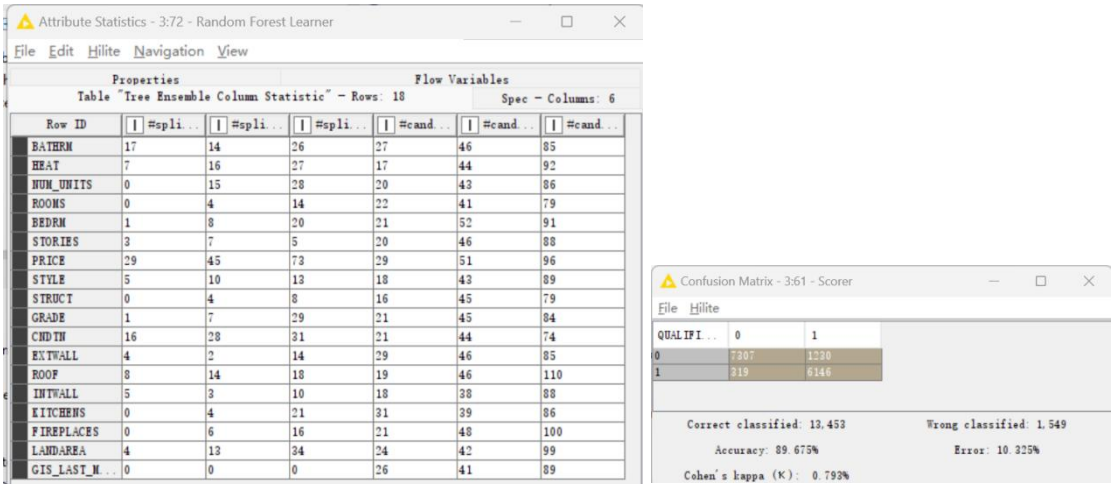
5.4.1 Using Random Forest node

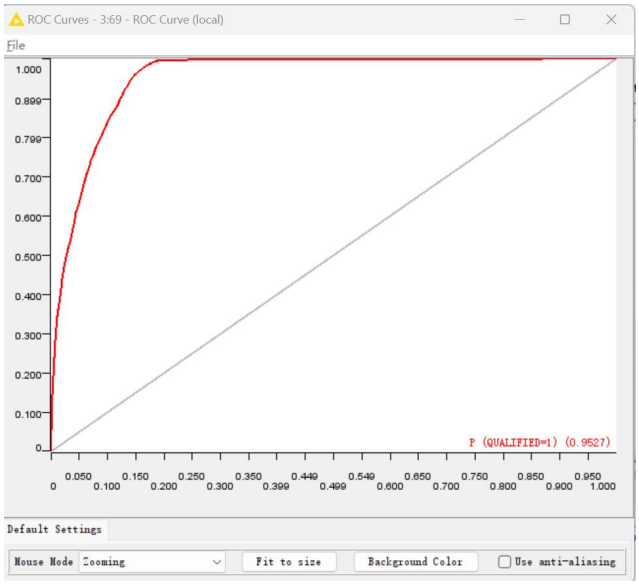


Random forest is composed of multiple decision trees, before setting the random forest model, we need to set the node size, the number of trees, the number of sampled features. Random forest can handle regression and classification tasks with high accuracy, and deal with missing values to evaluate the importance of variables while ensuring data accuracy. We can use random forests to process large-scale data and see important features. Here are the parameters we set:



By adjusting the parameters, we obtain optimal results:

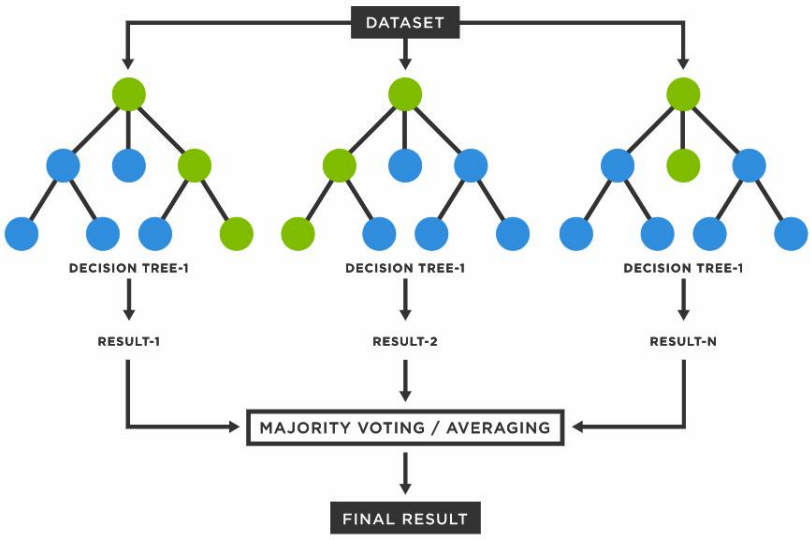




According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.675% , and the value of the ROC curve AUC is 95.27%.

5.4.2 Conclusion

Random forest is an ensemble learning method that predicts the outcome of multiple decisions by constructing multiple decision numbers in parallel. It integrates bagging thinking and random selection features to construct multiple decision trees, and uses the voting method to select the final result according to the prediction results (What Is a Random Forest?, n.d.). Forest is then better at dealing with classification problems and less good at regression problems because it cannot make predictions beyond the training set without continuous output, which can lead to outliers overfitting the data modeling.



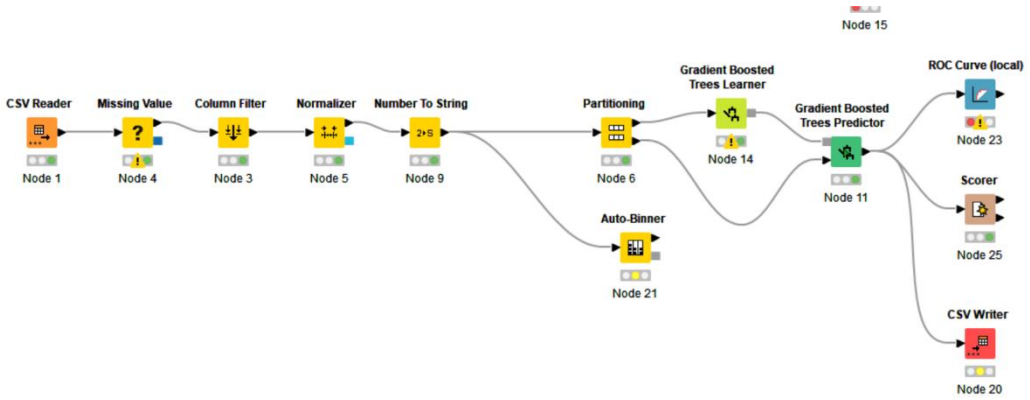
In the exercise, we can find that the random forest can obtain a high AUC and the correct rate, indicating that the model has good robustness.

Score for best Decision tree:

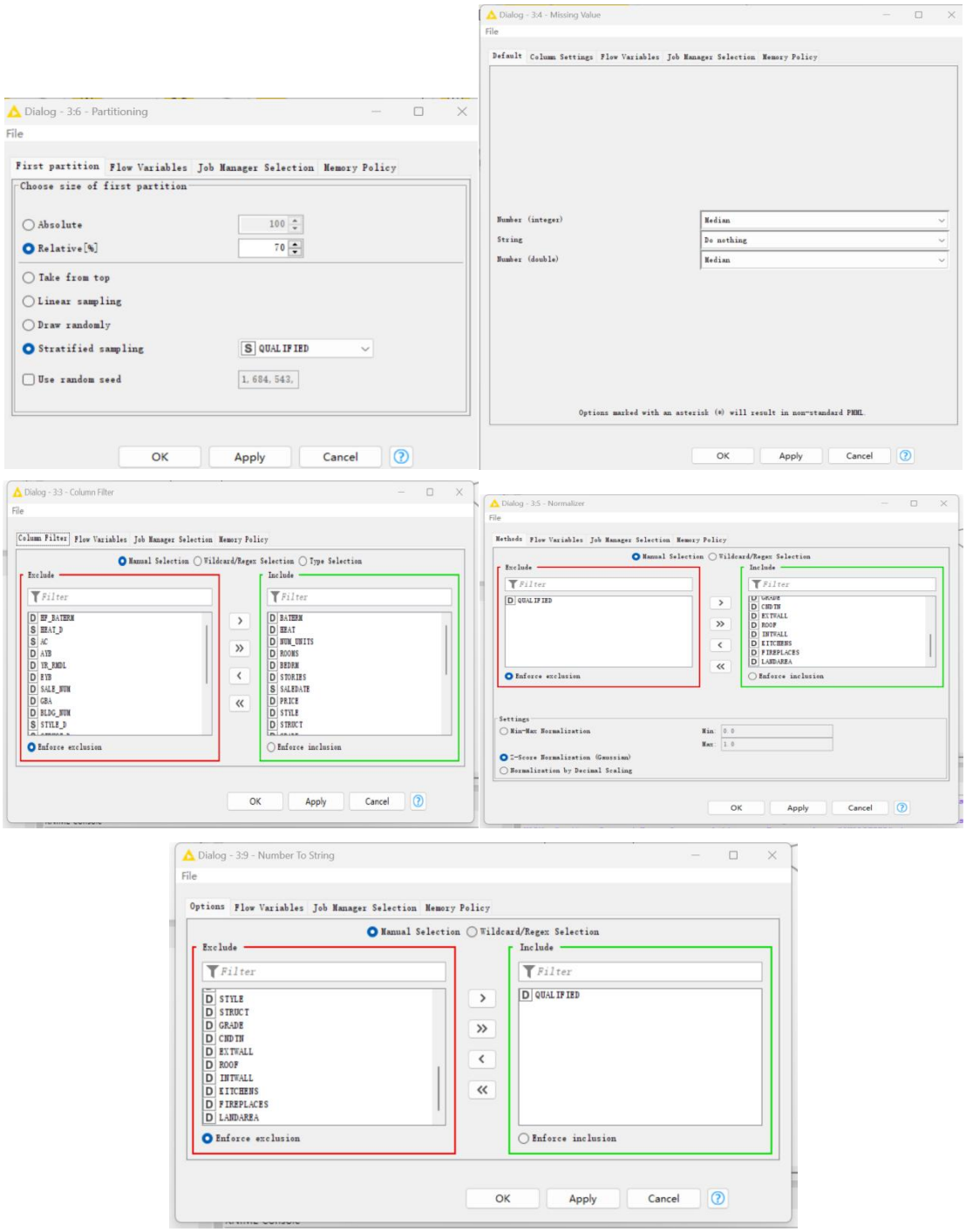
Accuracy	89.675%
Precision	95.18%
AUC	95.27%
Error	10.325%
Recall	85.51%
Cohen's kappa	0.792%
F1 score	90.34%

5.5 Gradient Boosted Trees

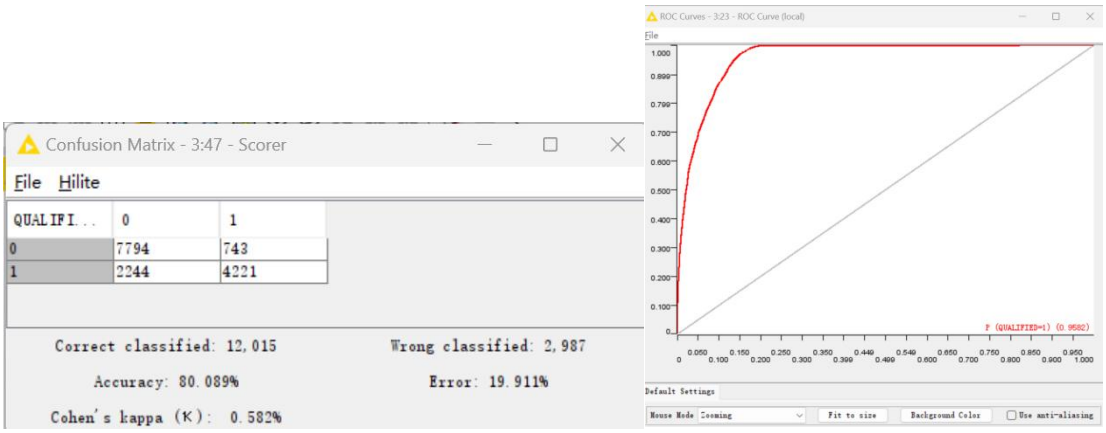
5.5.1 Using Gradient Boosted Trees node



Gradient boosting decision tree is an iterative decision tree algorithm, which consists of multiple decision trees, constructs the tree in a continuous way, and each tree is a graph to correct the error of the previous tree. Gradient boosting decision tree and SVM have always believed that it is an algorithm with strong generalization ability. The algorithm used to improve the tree is the average error loss function, each tree regression tree learns the conclusion residuals of all trees and fits the current residual regression tree.

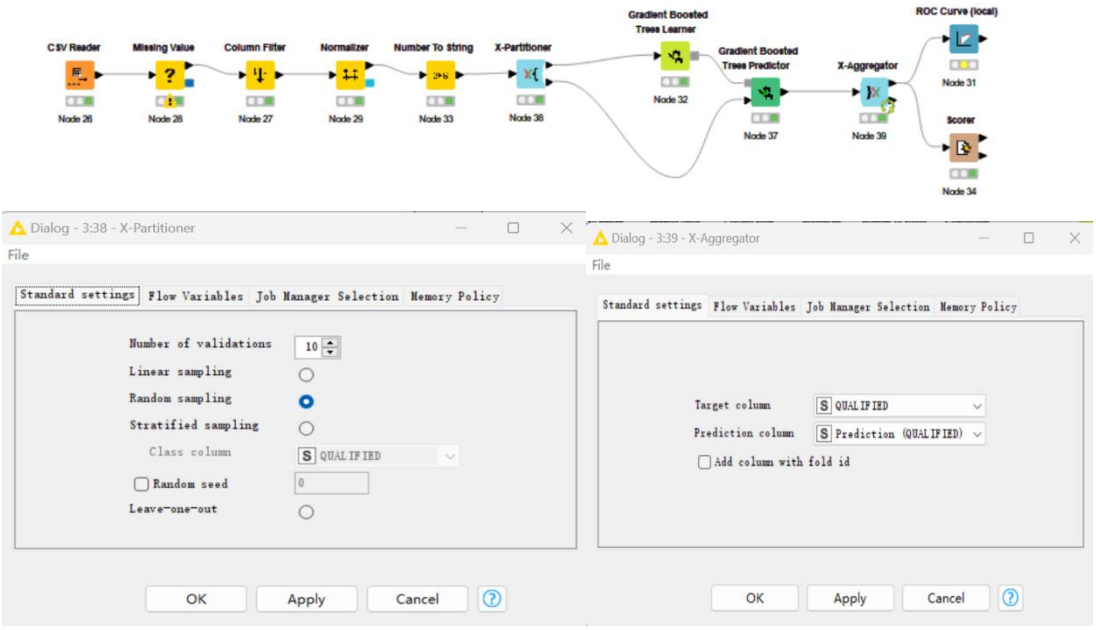


Here, we adjust the parameters of the model, including setting 70% of the training set and 30% of the test set, using the median instead of missing values, using Z-score normalization, filtering out some attributes that are of little use, and converting the type of quality attributes. Since the optimal AUC area and prediction accuracy of ROC have been obtained after many attempts with these parameters, these parameter settings are applicable to the preprocessing of all models.

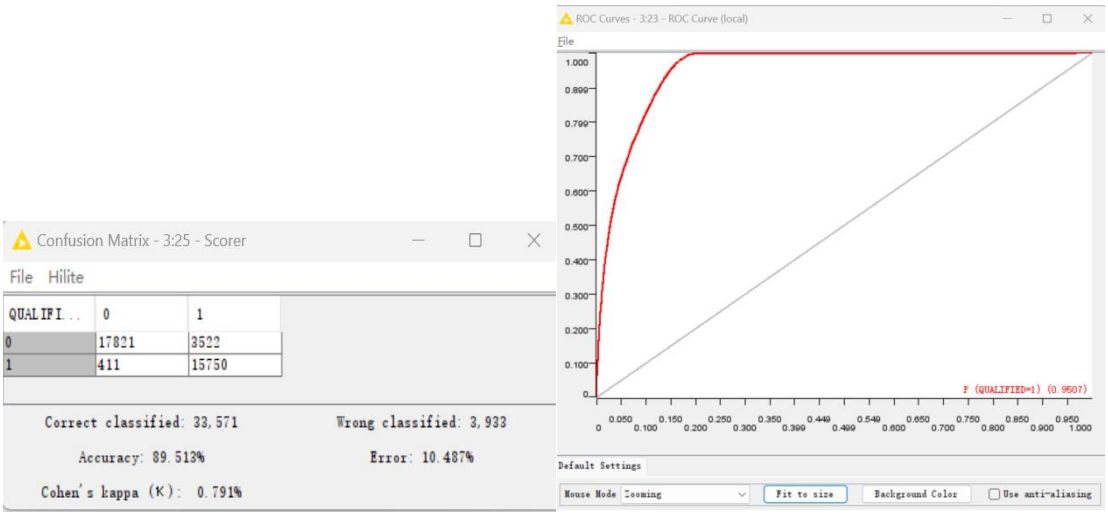


According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 80.089%, and the value of the ROC curve AUC is 95.82%.

5.5.2 Use x-partition and x-aggregator to preprocess the classifier



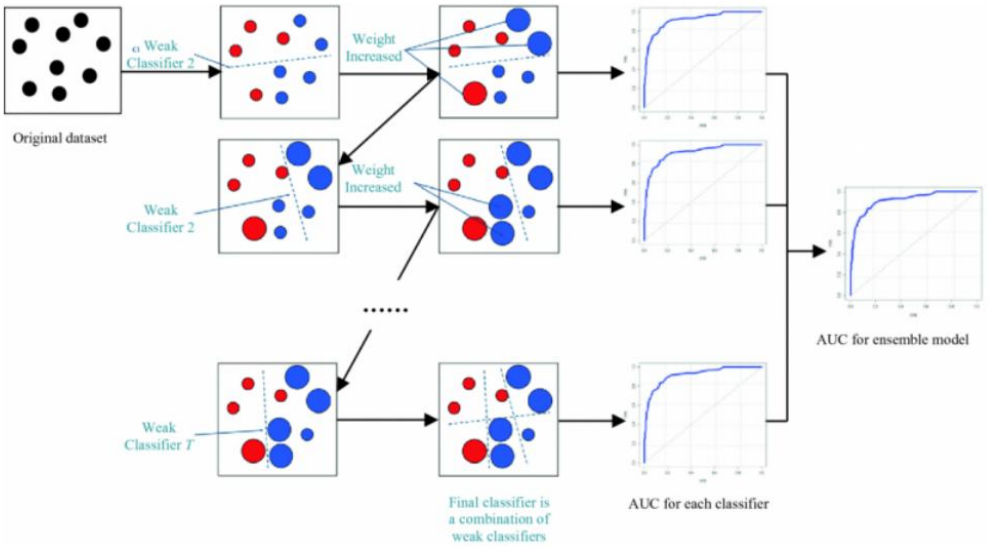
Use these x-partition and x-aggregator nodes to split and process data in parallel, which can facilitate the calculation of data, statistically dispersed data is called a global result, divide computing tasks into sub-tasks for merging and summarizing, and facilitate data storage and retrieval. The application of these x-partition and x-aggregator nodes often improves the computational efficiency of the model as well as large data processing. We set the number of validations to 10 and use random samples.



According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.513%, and the value of the ROC curve AUC is 95.07%. Compared with the model without this node, the accuracy rate and AUC of ROC are higher.

5.5.3 Conclusion

The gradient boosted trees classifier follows the greedy function approximation algorithm, which can be adopted when we want to reduce the error. This can be thought of as a method for transforming weak learners into strong learners and the last model prediction will be the result of the overall weighted prediction provided by the previous all-number model. Increasing the gradient solves the highly correlated problem between variables. Gradient boosted trees (Gradient Boosting – What You Need to Know — Machine Learning, 2020):



The gradient boosting tree has super high prediction accuracy, especially when dealing with large-scale datasets and complex problems, iteration can be used to improve the accuracy of the model, and the model does not require much preprocessing to automatically process missing values and obtain feature importance rankings. However, the model has complex calculations, so it needs to waste a lot of resources and time, and overfitting due to too many iterations of depth expansion, and the learning rate is set too high.

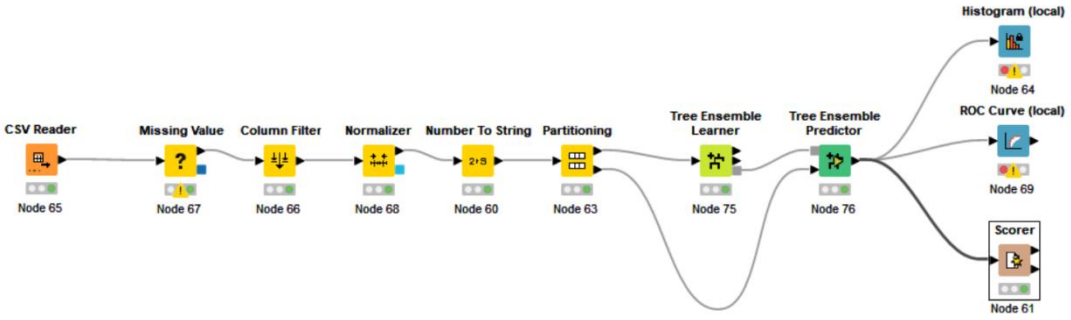
Both the gradient boosting tree and the random forest use the decision tree measurement as the basic classification algorithm, but through the AUC area in the R OC curve and prediction accuracy in the exercise, we can see that the two algorithms are more accurate in prediction and fitting. There are significant differences in stability,

Score for best Decision tree:

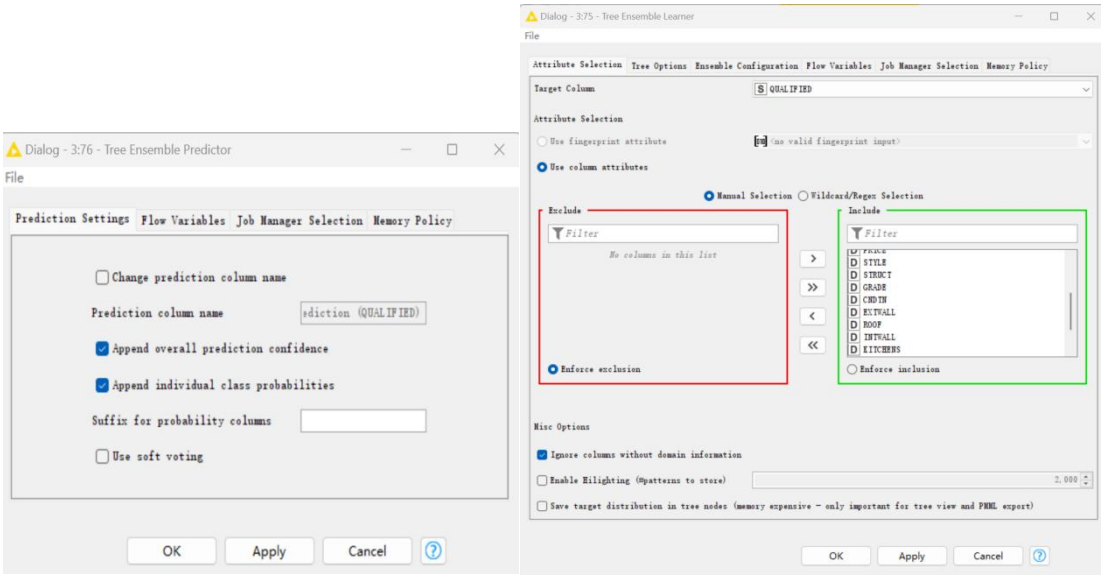
Accuracy	89.513%
Precision	97.72%
AUC	95.07%
Error	10.487%
Recall	83.53%
Cohen's kappa	0.791%
F1 score	90.01%

5.6 Tree Ensemble Classifier

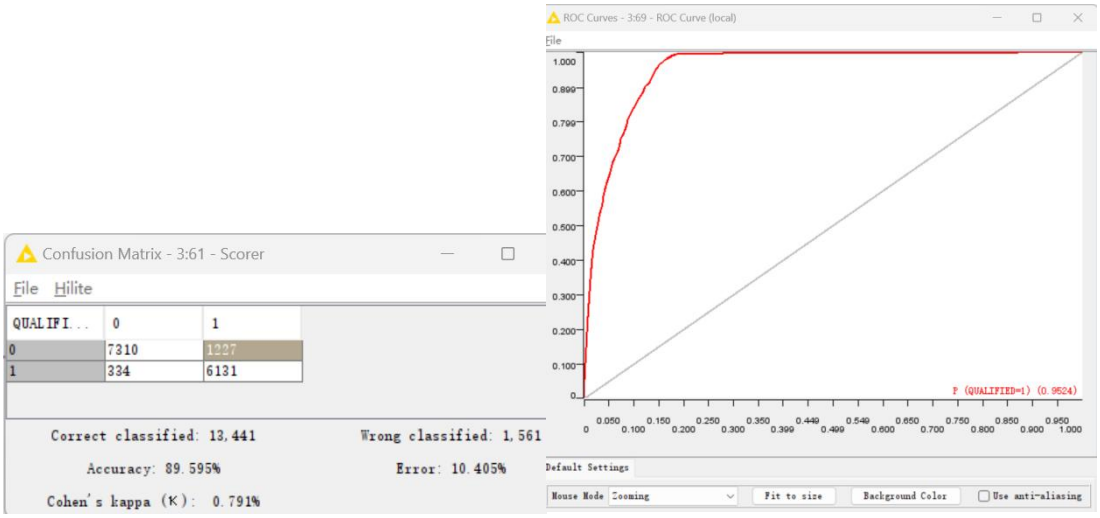
5.6.1 Use Tree Ensemble node



Tree Ensemble Classifier node is a general classifier that allows users to select different ensemble learning algorithms, including random forest and gradient boosting tree based on decision tree. Random forests are constructed using random sampling and random subsets of features, and the results are predicted by voting or averaging. Gradient boosting trees focus on iterative integration, and continuously improve the performance of the model by continuously constructing to reduce the errors and weights of the previous tree's prediction sample.



In the node parameters, we set" Ignore columns without domain information", we will take a screenshot of the related operation, and the result is as follows:



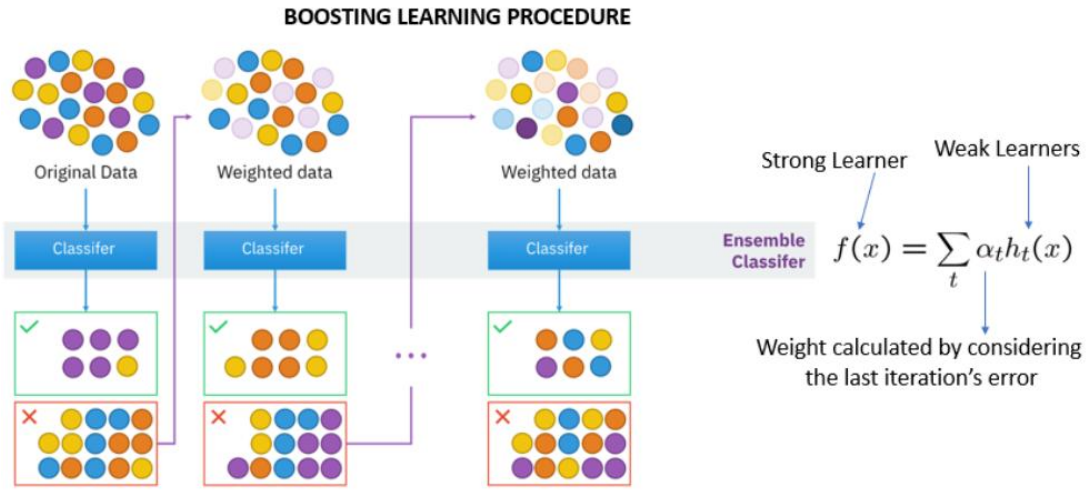
According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.595%, and the value of the ROC curve AUC is 95.524%.

5.6.2 Conclusion

Random forest and gradient boosting tree are the embodiment of Tree Ensemble node, in which conceptual bagging and boosting are used in random forest and gradient boosting tree. By adjusting the node parameters, we studied the prediction accuracy of the model and the ROC curve AUC area to reach the following conclusions.

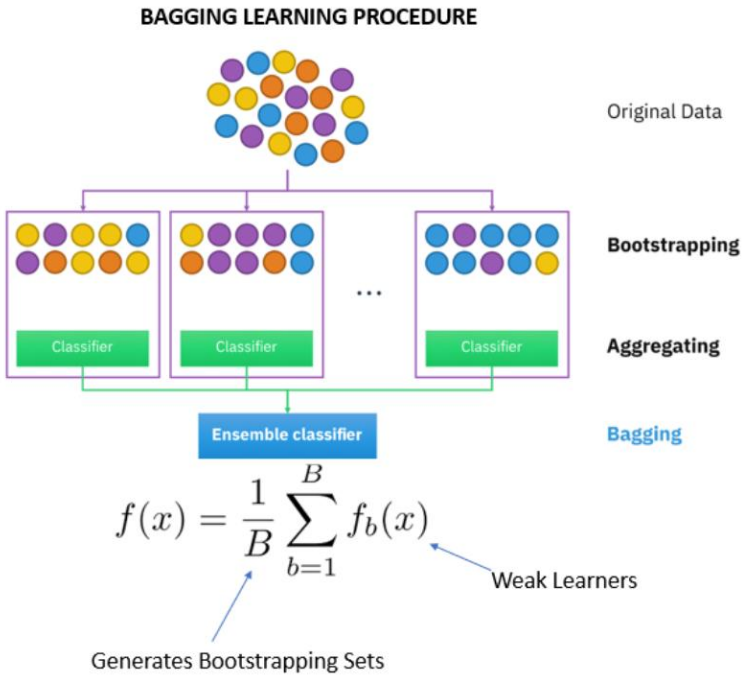
Boosting:

Boosting is usually used in gradient boosting trees, and the basic principle is iteration, giving weight to each weak learner, and modifying the errors of the previous learner by the latter weak learner. Picture here show Boosting (Wikipedia Contributors, 2019):



Bagging:

Bagging simply uses the auto-adoption method to train multiple subsets in the raw data, build a weak learner for each stimulus, and the construction process is carried out concurrently, thoroughly using each data of the overall sample. Picture here show Bagging (Wikipedia Contributors, 2019):



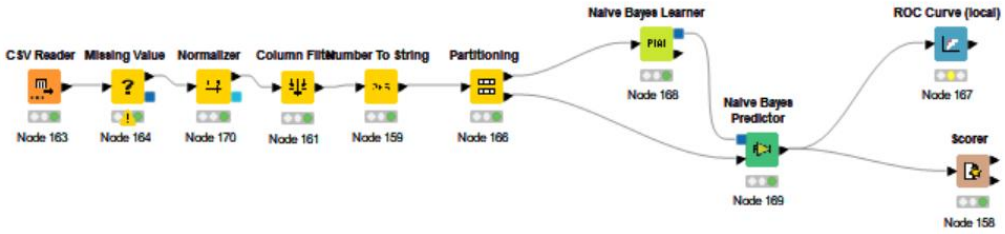
Score for best Decision tree:

Accuracy	89.595%
Precision	95.625
AUC	95.524%
Error	10.405%
Recall	85.65%

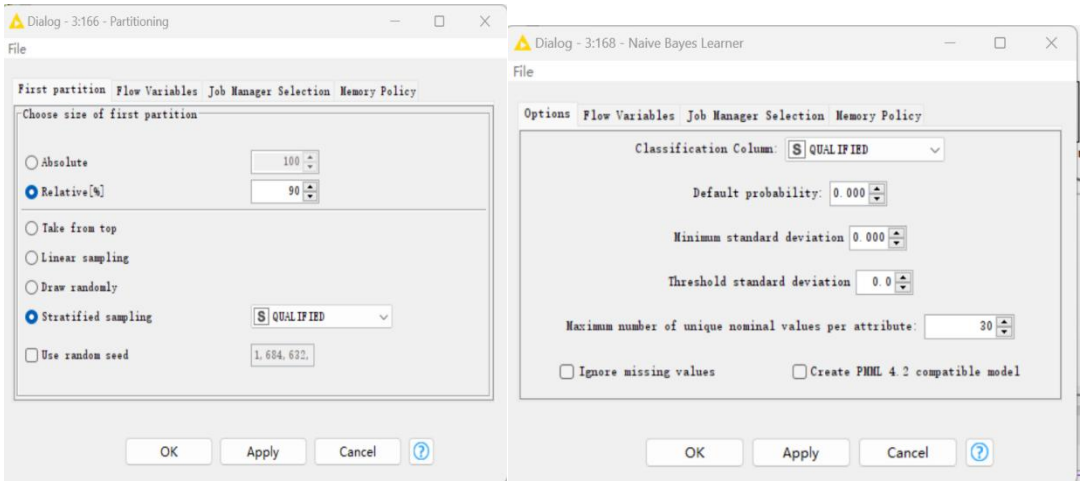
Cohen's kappa	0.791%
F1 score	90.37%

5.7 Naive Bayee

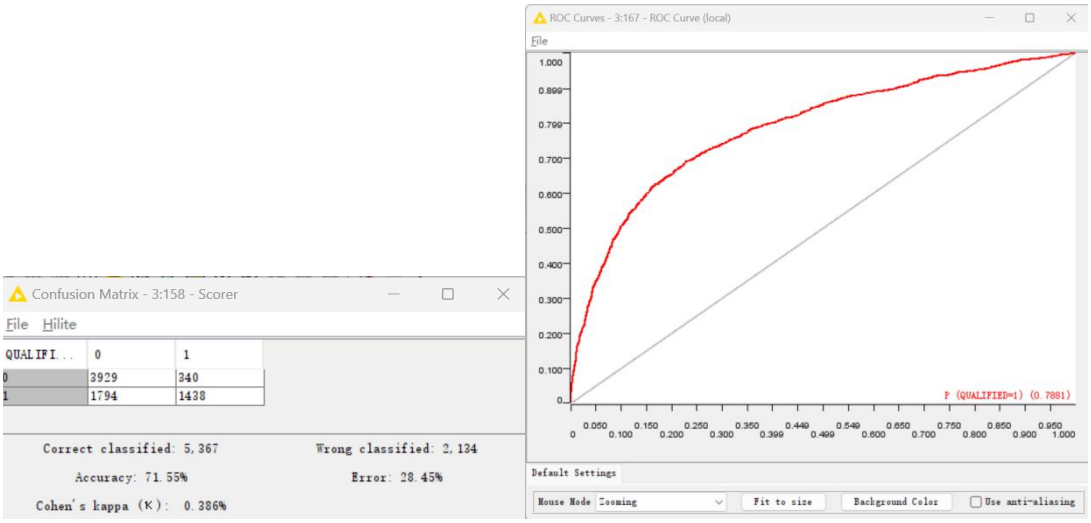
5.7.1 Using Naive Bayee node



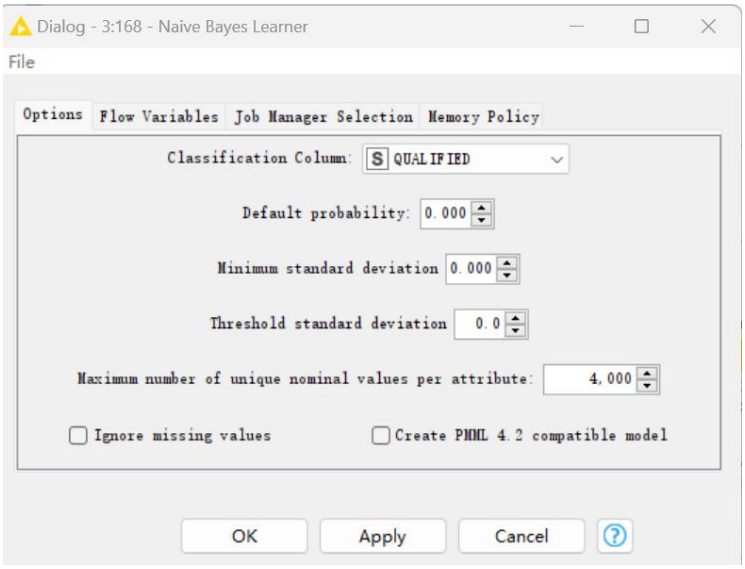
The Naive Bayes classification system is a classifier node based on Bayes' theorem algorithm that constructs a hypothesis that all features are independent. This node computes the Gaussian distribution of attributes for each number of rows in each class and attributes, similar to a simple probability classifier.



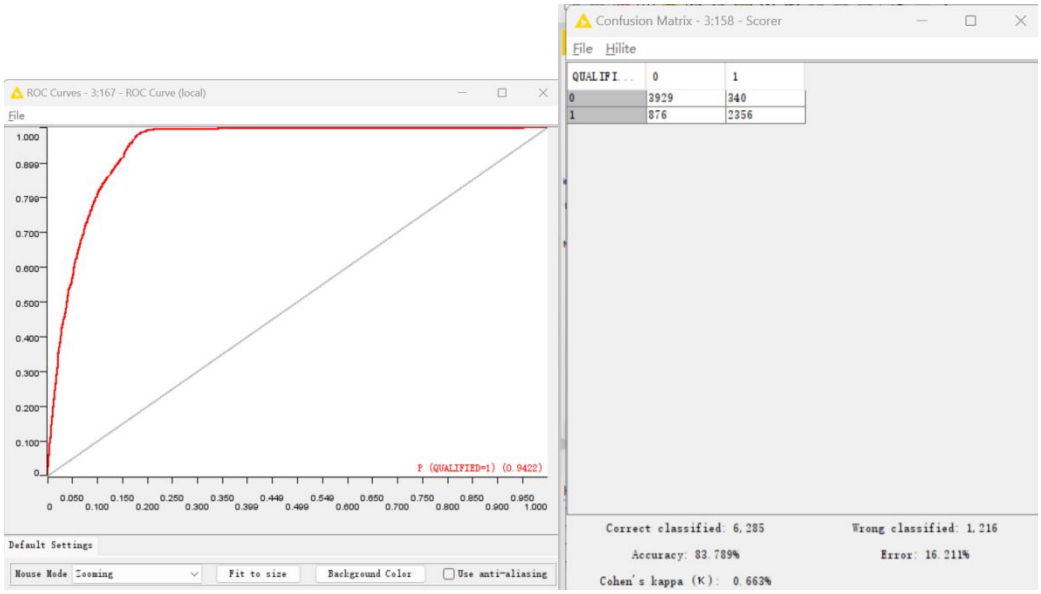
In adjusting the model, we set 90% of the training machine and 10% of the test machine and adjust unique nominal value the maximum number of per attribute is 30, the following are the parameter results obtained by the model:



According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 71.55% , and the value of the ROC curve AUC is 78.81%. By adjusting the parameters of Naive Bayee model several times, and looking at the final accuracy rate and ROC curve, it can be found that naive bayee node is not suitable for this data set, and the accuracy can not be obtained well by using naive bayee node, and even submitting to Kaggel is prone to overfitting.



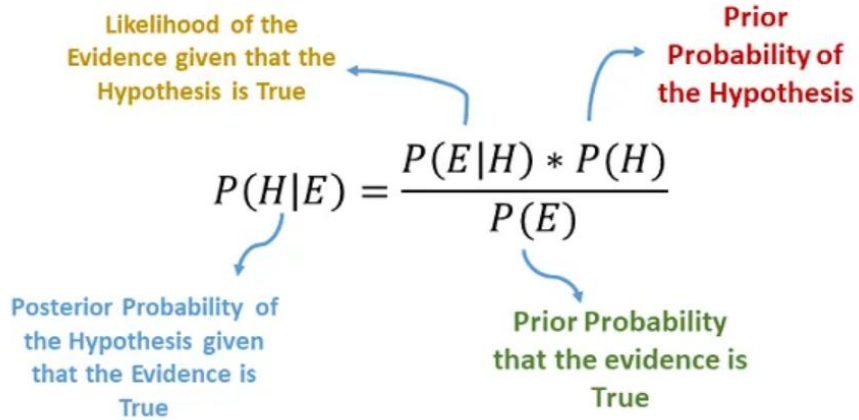
After many attempts and adjusting different nodes, we found that when adjusting the maximum number of unique values of each attribute to 4000, we can get a better accuracy rate and R O C curve.



According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 83.789%, and the value of the ROC curve AUC is 94.22%.

5.7.2 Conclusion

The advantage of Naive Bayes is that it requires a small amount of training data, so the training time is small, and it can be suitable for processing continuous and discrete data that is not sensitive to irrelevant features. Here show Naive Bayes algorithm (Gamal, 2021):



The model cannot obtain a good ROC curve AUC area and accuracy may be that the feature conditions are not independent, because in the Bayesian algorithm, we propose that the assumption is that the special liters are independent of each other, but in this data set, there may be strong correlation between features, and there may be dependencies between features or uneven probability of conditional distribution, which may also lead to the performance degradation of the classifier.

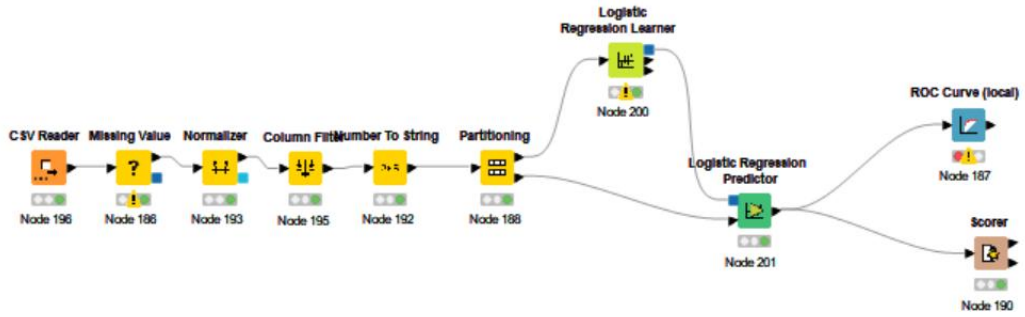
Score for best Decision tree:

Accuracy	83.789%

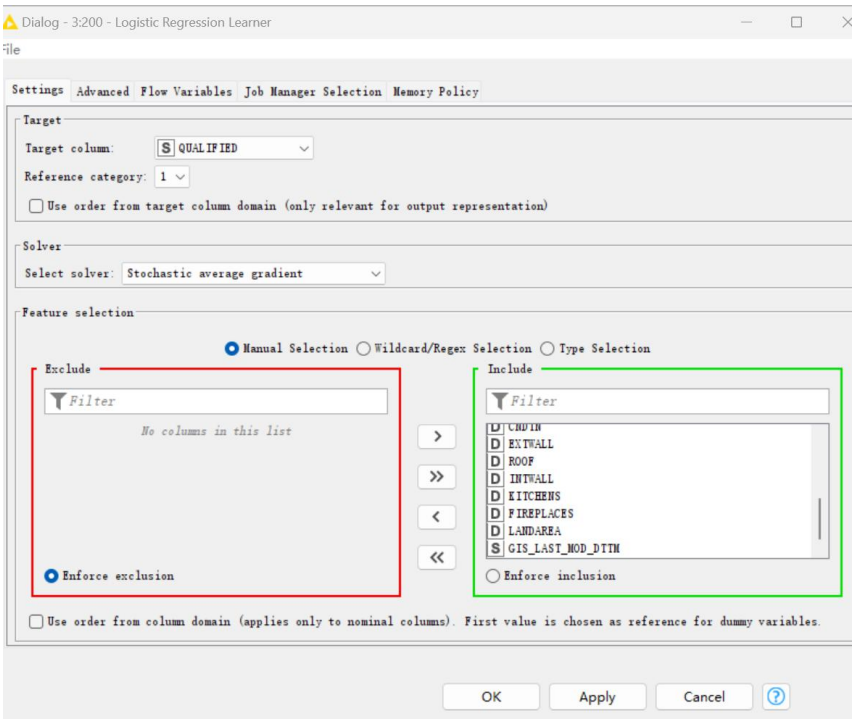
Precision	95.62%
AUC	94.22%
Error	16.211%
Recall	85.65%
Cohen's kappa	0.663%
F1 score	90.37%

5.8 Logistic Regression

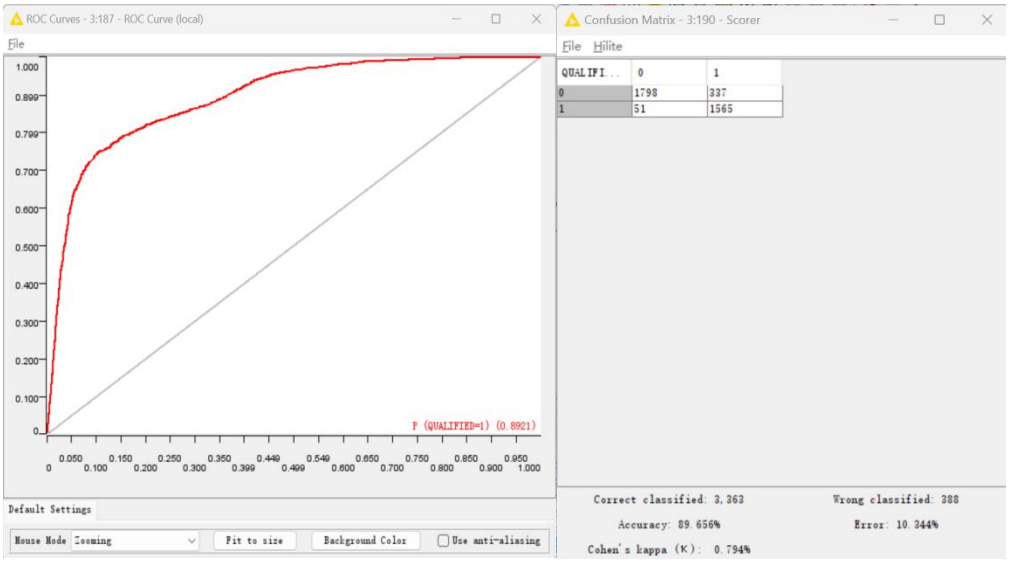
5.8.1 Using Logistic Regression node



Logistic regression is a classifier that uses logistic functions to map the output of a linear model to a probability between zero and one to classify only. It differs from linear regression in logistic regression, which is used for categorical and continuous variables, which is relatively easier to understand, but logistic regression can handle large amounts of data.

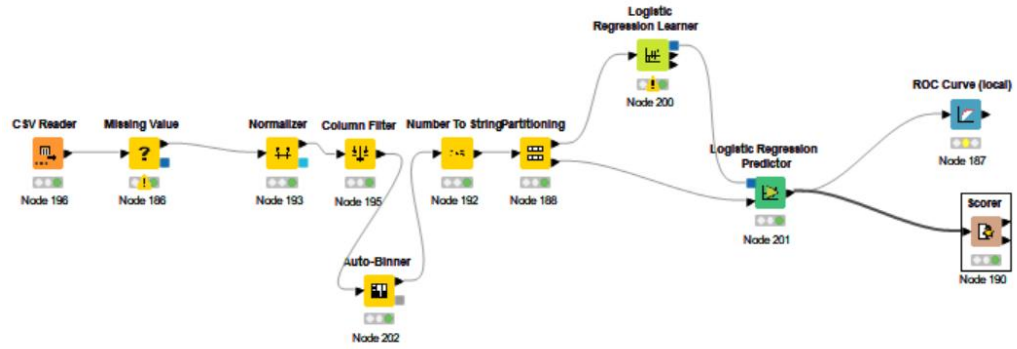


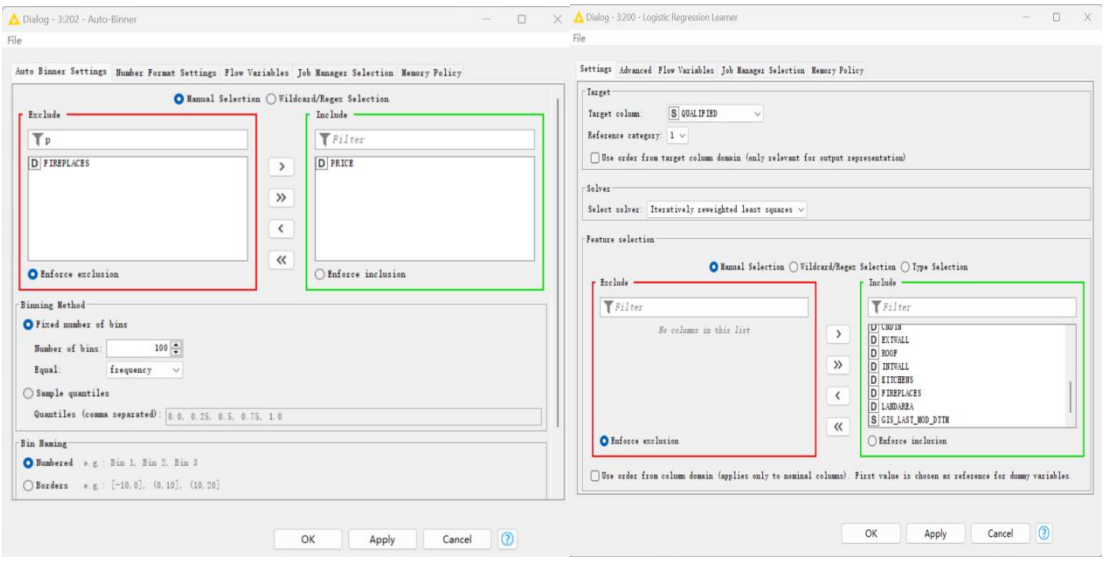
We choose that the solver is “Stochastic average gradient”, and here is the ROC curve and accuracy of the output of this model:



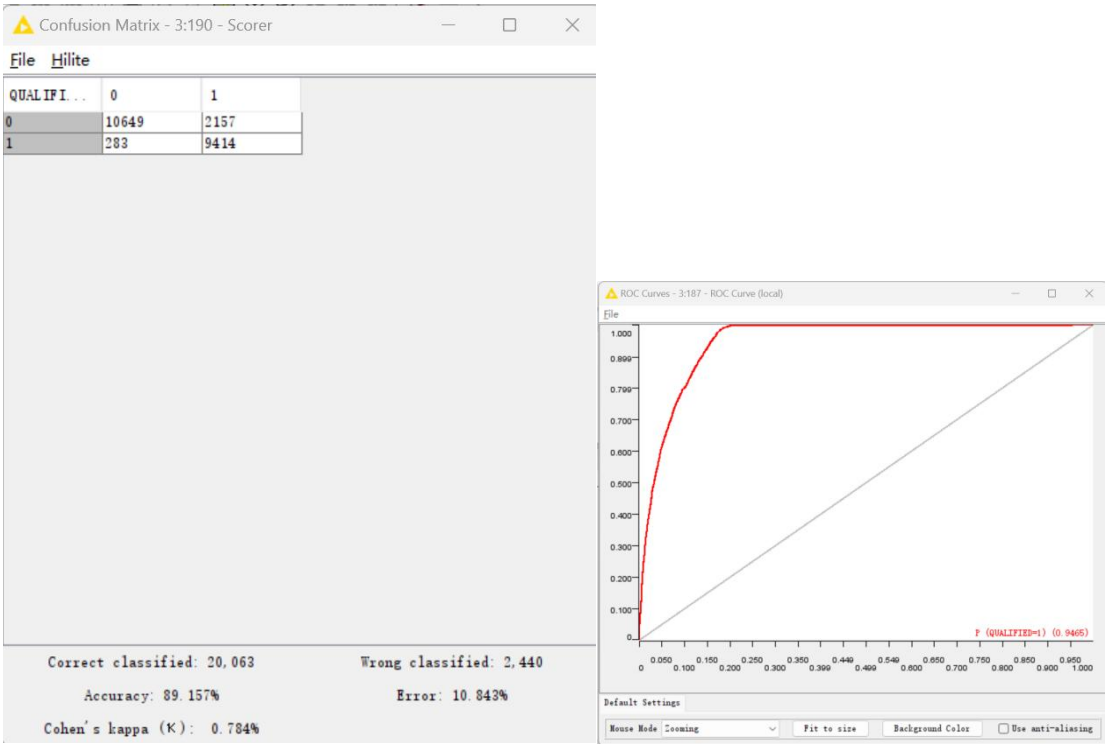
According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.656% , and the value of the ROC curve AUC is 89.21%.

5.8.2 Use automatic binning to preprocess the classifier





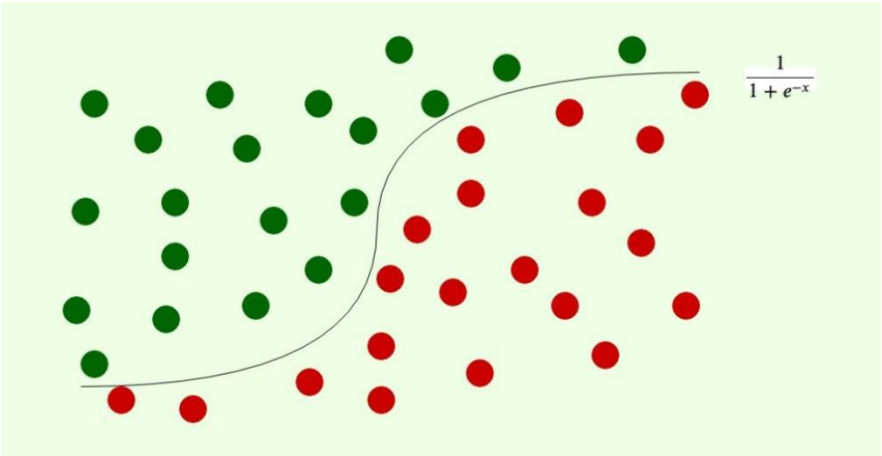
We tried to use automatic binning to improve the reliability of data input and reduce the impact of exceptions only on the model. We found that the automatic binning function has a great impact on this model. Improve data accuracy and model performance. We set up 100 boxes, divided into bins according to width.



According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.159%, and the value of the ROC curve AUC is 94.65%.

5.8.3 Conclusion

As shown in the following figure, logistic regression is like a line separating the green and red dots (Logistic Regression - Voxco, n.d.):



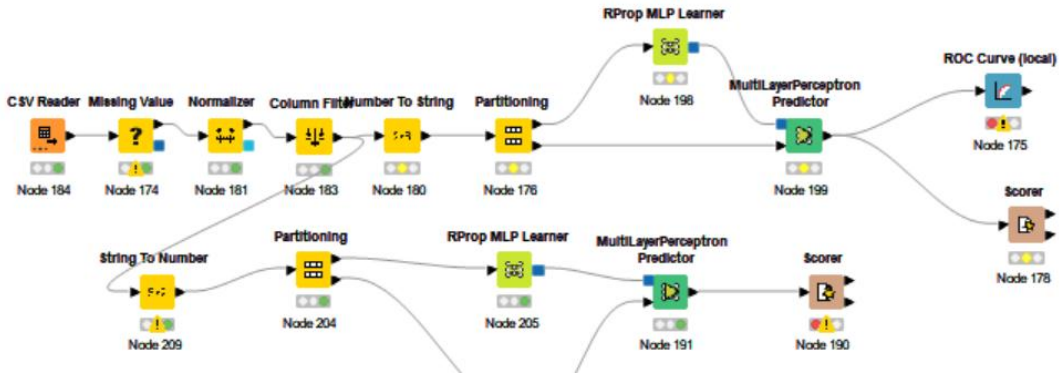
The classifier model, compared with other classification models such as decision trees, can not obtain high accuracy and the AUC area of the ROC curve may be due to logistic regression, assuming that the relationship between features and logarithmic probability is linear, but if there is a nonlinear relationship between features and features, logistic regression may not be captured, and the correct response cannot be corrected. At the same time, logistic regression is more sensitive to outliers, so it may have a greater impact on the results of fitting.

Score for best Decision tree:

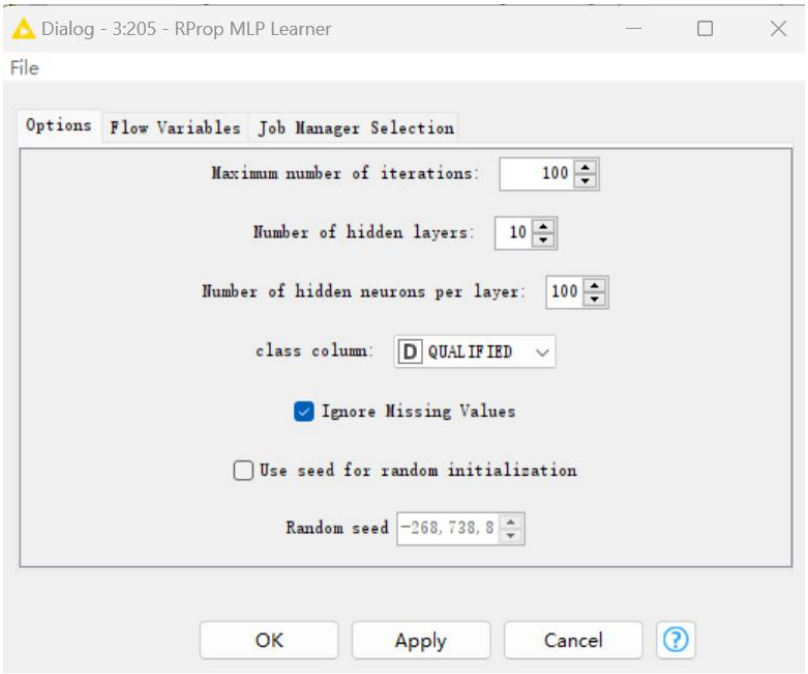
Accuracy	89.656%
Precision	97.29%
AUC	89.21%
Error	10.344%
Recall	84.22%
Cohen's kappa	0.794%
F1 score	90.24%

5.9 MLP

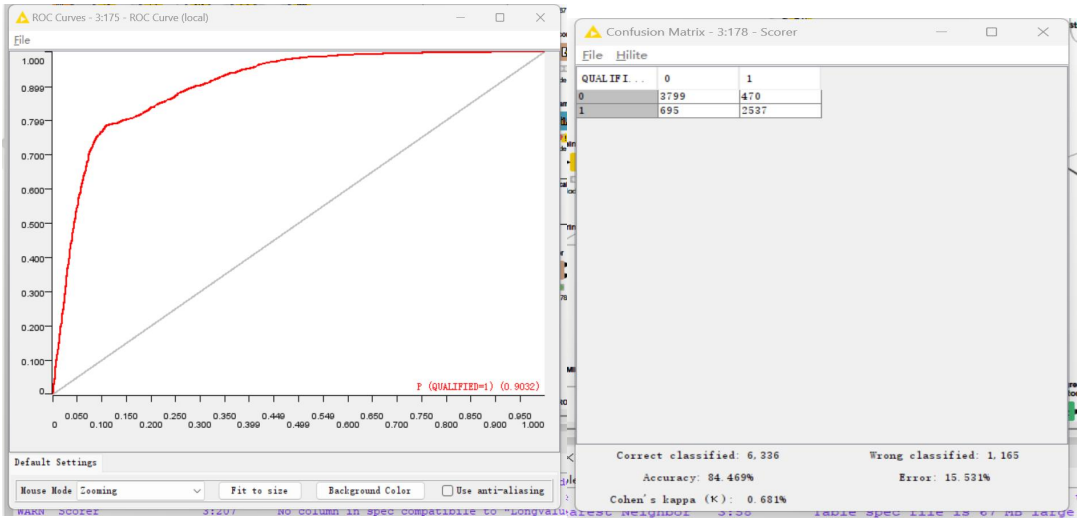
5.9.1 Using RProp MLP node



RProp is a gradient descent algorithm, a neural network algorithm used to train multiple layers of perception, very similar to the back-propagation algorithm, it can be used to adjust weights and biases in a neural network to improve the accuracy of the model

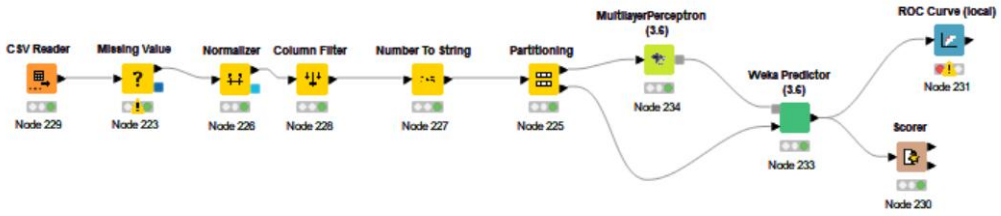


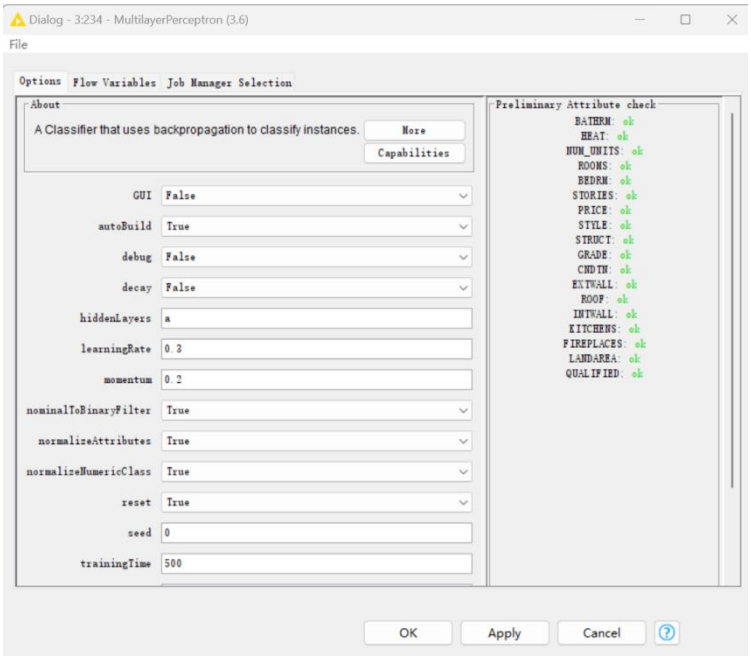
At this node, we set the hidden layout to 10, the maximum number of iterations is 100, the number of each hidden neural network is 100, and we set to ignore missing values, according to the following operation, we view the final result:



According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 84.469%, and the value of the ROC curve AUC is 90.32%.

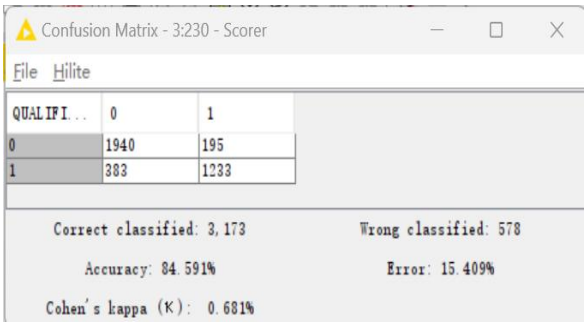
5.9.2 Using Multilayer Perceptron (MLP) Node

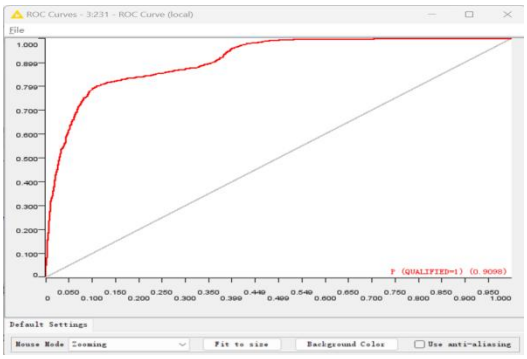


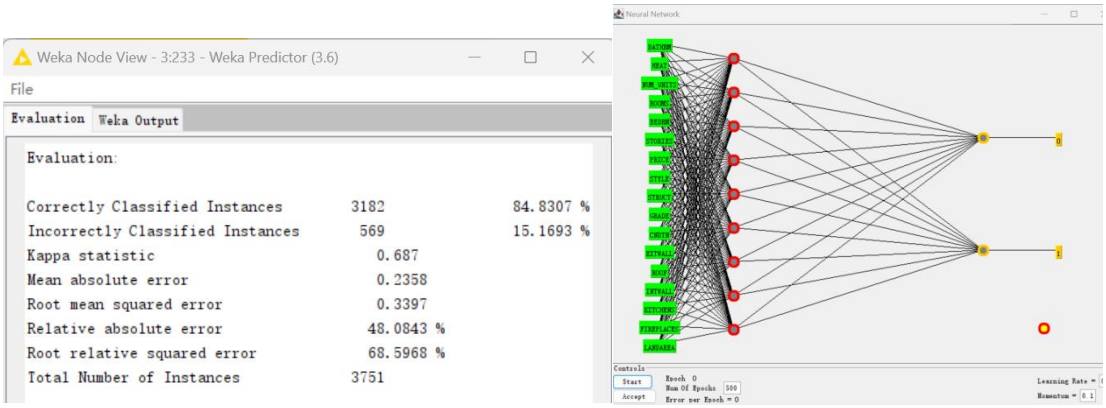


MLP node uses the backpropagation algorithm, which not only calculates the loss function, adds weights and biases to the neural network, and adjusts the parameters according to the error function behavior, so that the network can be better optimized.

MLP is a common prefeedback neural network, in which neurons propagate from the input layer to the output layer like brain neurons. Here we set the learningrate to 0.3, set the momentum to 0.2, and then observe the result:





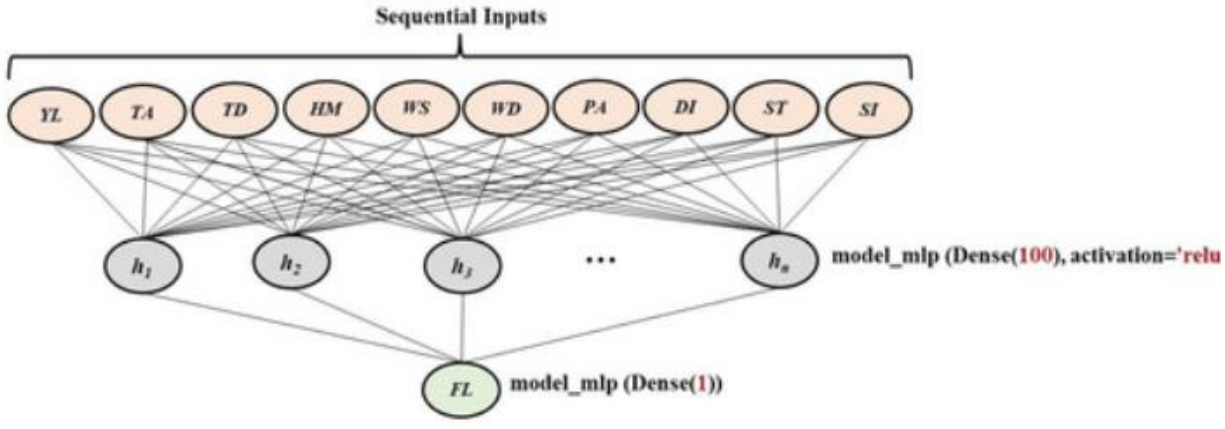


According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 84.591%, and the value of the ROC curve AUC is 90.96%.

5.9.3 Conclusion

MLP and Rrop MLP use different types of algorithms, one focuses on backpropagation and updates the model with gradient descent, and the other focuses on adaptive learning rate, adjusting the learning rate according to parameter gradient changes. In the process of training and testing common sense, we found that the training speed of RPROP will be faster, and there is no need to specify parameters to reduce the human factor to affect the improved accuracy.

MLP has at least three node layers, each node chamber uses a nonlinear activation function, the model has a significant advantage is that it has strong robustness and fault tolerance, suitable for applying to complex data models and tasks, searching for nonlinear relationships. Below is a schematic diagram of the MLP logic (BetaE, n.d.):



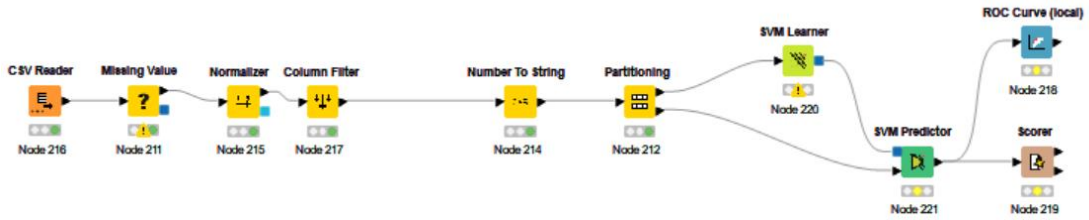
Score for best Decision tree:

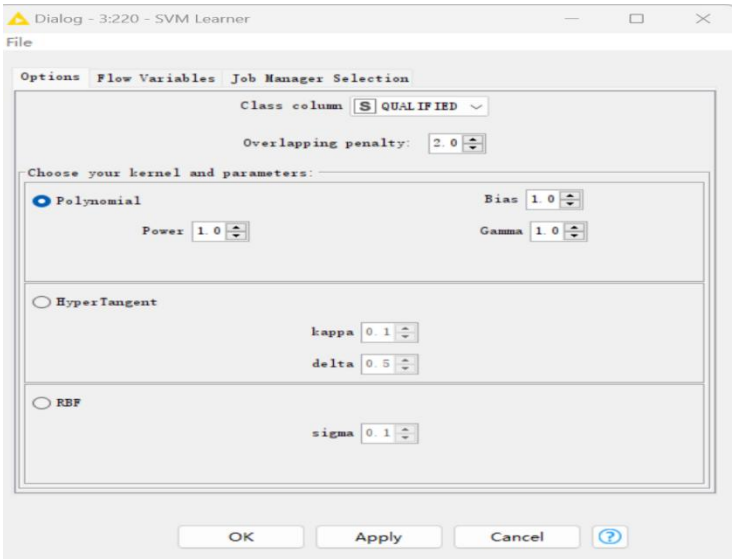
Accuracy	84.591%
Precision	83.53%
AUC	90.98%
Error	15.409%

Recall	90.88%
Cohen's kappa	0.681%
F1 score	87.04%

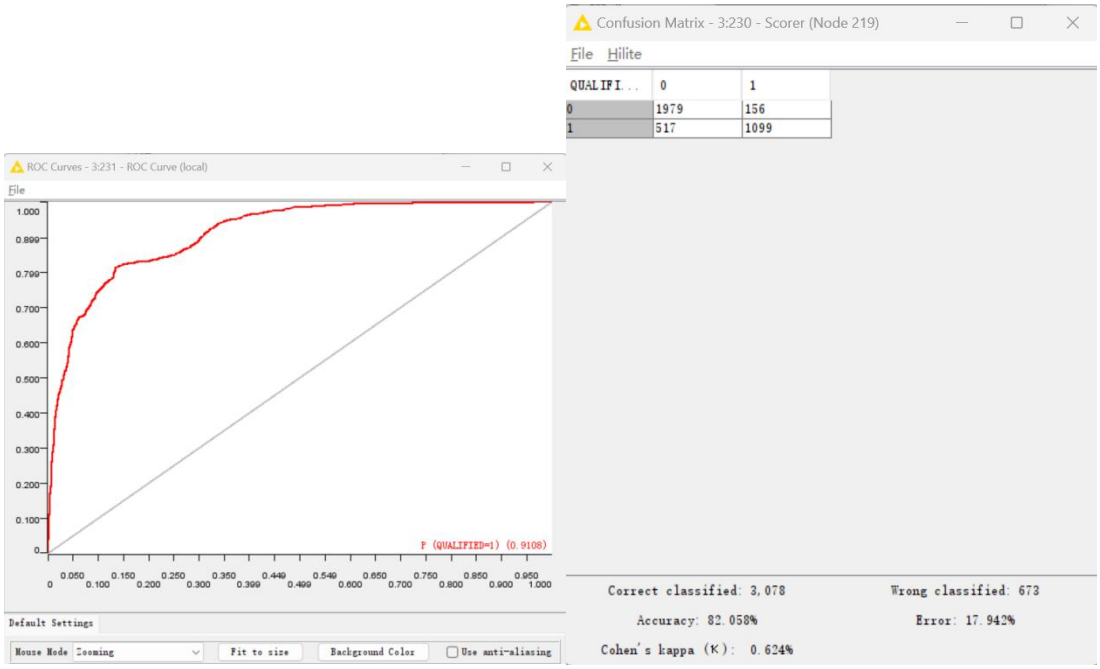
5.10 SVM

5.10.1 Using SVM node





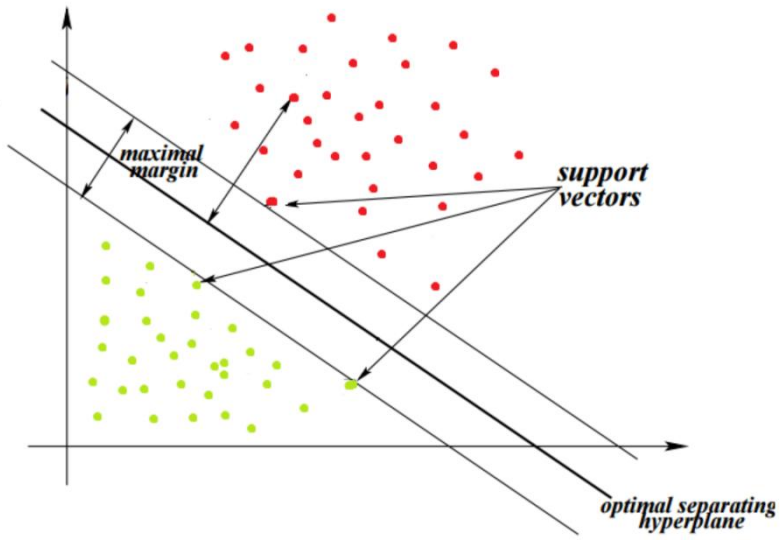
The full name of SVM is a support vector machine, which is a non-probability binary linear classifier, and the point closest to the separated hyperplane in the training dataset sample is called the support vector, and his purpose is to find an optimal hyperplane that can separate data samples of different classes.



According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 82.055%, and the value of the ROC curve AUC is 91.08%.

5.10.2 Conclusion

In training, we found that SVM takes a very long time to train a large amount of data, and does not reflect good accuracy and performance, the advantage of S B M is that it has good classification performance in high-dimensional space, and has good generalization ability in small sample data, but the model is easily affected by data noise and overlap, resulting in poor performance of the model in this dataset. The calculation method and visual representation of S V M are attached here:



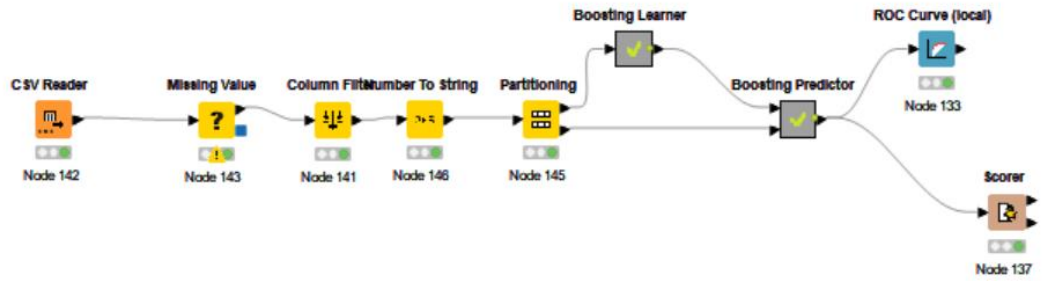
Score for best Decision tree:

Accuracy	82.058%

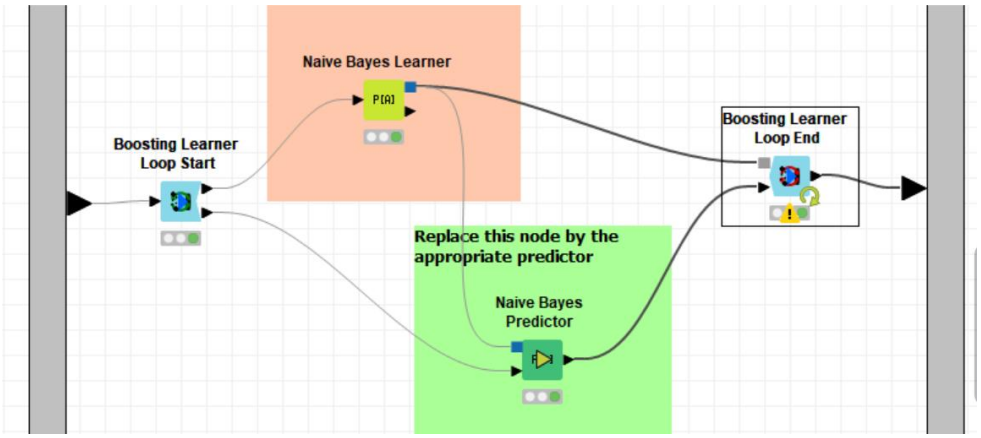
Precision	79.21%
AUC	91.08%
Error	17.942%
Recall	92.62%
Cohen's kappa	0.624%
F1 score	85.39%

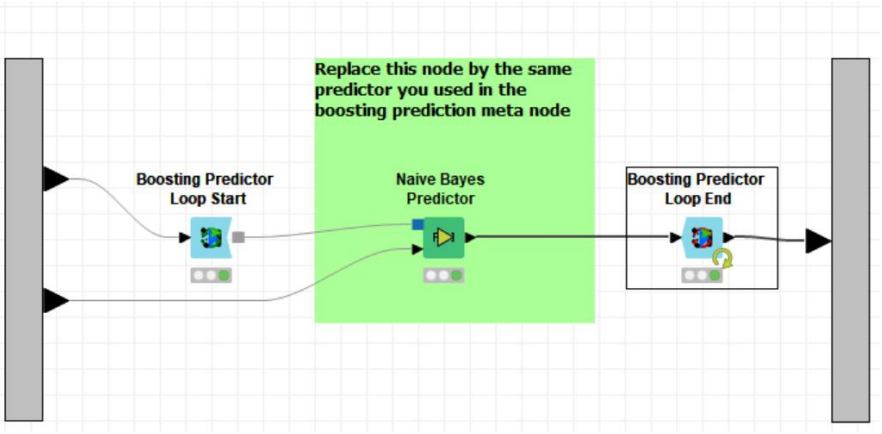
5.11 Boosting

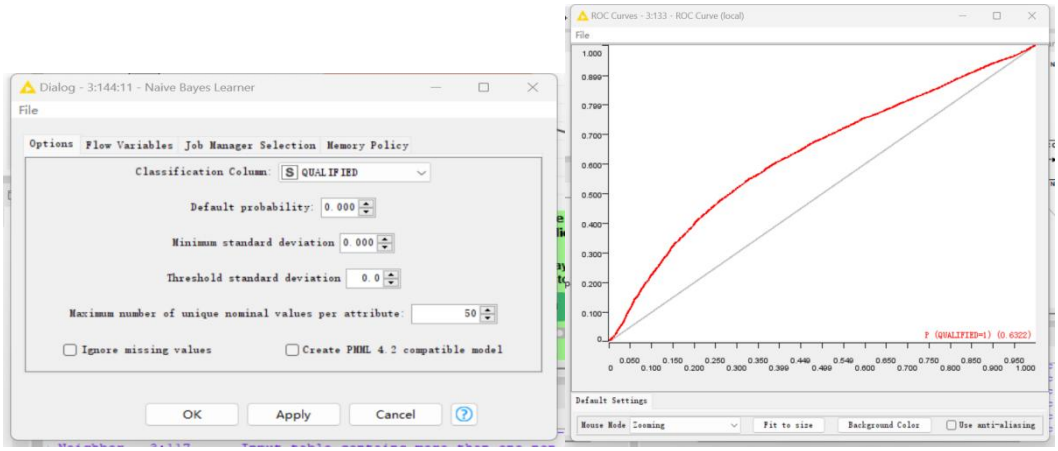
5.11.1 Using boosting learner and predictor

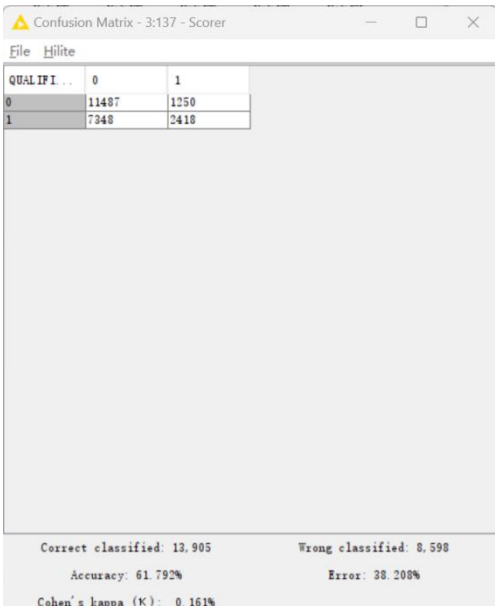


Boosting learner combines multiple weak learners to improve the performance of the overall model, and when we look at the details of this node, the model used by this node is a naïve Bayes model that uses boosting learner at the same time before using the classifier loop.boosting has anti-fitting ability, and iteratively eliminates the influence of noise data, and naïve Bayes model uses Bayesian algorithm to verify probability classification, the combination of the two can cancel each other out the shortcomings and improve the accuracy.









According to the confusion matrix and ROC curve, the prediction accuracy of this classifier is 89.595%, and the value of the ROC curve AUC is 63.22%. However, through experiments, we found that we could not get good accuracy and performance, which may be a serious imbalance of categories in the training dataset or weak learning ability. We can try to adjust the parameters or increase the credibility of the data to get a higher accuracy.

5.11.2 Conclusion

Through our research, we find that the gradient boosting tree and boosting learner both use iterative training weak learners, but according to the accuracy of the research model and the AUC area of ROC, we can get that the accuracy of gradient boosting trees will be higher. The advantage of gradient boosting trees is that the method of optimizing the loss function, the traditional method is by weighting the error rate and weighted residual value, but the gradient boosting number uses gradient descent to optimize, so it has better classification function.

Score for best Decision tree:

Accuracy	61.792%
Precision	61.05%
AUC	63.22%
Error	38.208%
Recall	90.14%
Cohen's kappa	0.161%
F1 score	72.94%

6. Best classifier

For this number set, we use a variety of classifiers and a variety of data preprocessing nodes to try to obtain the highest prediction accuracy and the best AU C area of the RO C curve. In the process of collecting data, we also used other metrics to evaluate the performance of the model:

- ✓ Accuracy

Accuracy refers to the ratio of the number of correct classifications to the total number of datasets, and the calculation formula is (TP + TN) / (TP + TN + FP + FN)

- ✓ Precision

It represents the proportion of the qualified sample predicted by the model that is truly qualified, and the higher the proportion, the higher the probability that the predicted house will be correct.

- ✓ AUC

Refers to the area in the ROC curve that is particularly suitable for evaluating dichotomous models, and if the higher the value of AUC, the closer to one, the better the performance.

- ✓ Error

Refers to the proportion of errors in the prediction, the formula is Error = (FP + FN) / (TP + TN + FP + FN).

- ✓ Recall

Recall refers to the ability of the model to correctly detect positive cases, which is different from the correct rate in terms of different angles of attention, recall is more concerned about the ability to recognize positive cases, and the accuracy rate comprehensively considers the classification accuracy of all positive and negative cases.

- ✓ Cohen's kappa

It is used to measure the uniformity between classifiers and reference standards.

- ✓ F1 score

Is the harmonic average of precision and recall and is used for model comparison when recall and accuracy conflict

Determine the best classifier criteria

Initially, my criterion for selecting the optimal model was accuracy, for the following reasons:

1. In this dataset, predicting whether a house is qualified is a binary problem, and the final classification result tends to be sample average, and we are more concerned with whether the model can predict the ratio between the correct sample number result and the total sample size.
2. The accurate performance intuitively observes the predictive ability of the model, and at the same time comprehensively considers the classification accuracy of positive and negative samples, and the data can be output directly from the confusion matrix without more calculation, which has the convenience of operation and the convenience of operation.

In addition to accuracy, I also value the metrics are precision and recall, because I used a **chart** to see the distribution of the results of the predictions, I found that more houses were predicted as unqualified, accuracy allows me to know that the model misjudged the unqualified rooms as a qualified rate, and when the accuracy rate, the lower the chance of mispositive. At the same time, I tried, uploaded different models to Kaggle, and found that the correct rate of training the model does not represent the accuracy of the predictive model.

Below are the names of all the classifiers and the indicators, the same algorithm may use several different preprocessing methods, we select the best performing results in the model for statistics：

Classifier name	**DT**	**KNN**	**RT**	**GBT**	**NB**	**LR**	**Boosting**	**SVM**	**MLP**
Accuracy	89.41%	88.654%	89.675%	89.513%	83.789%	89.656%	61.792%	82.058%	84.591%
Precision	96.47%	96.16%	95.18%	97.72%	95.62%	97.29%	61.05%	79.21%	83.53%
AUC	94.21%	93.78%	95.27%	95.07%	94.22%	89.21%	63.22%	91.08%	90.98%
Error	10.59%	11.346%	10.325%	10.487%	16.211%	10.344%	38.208%	17.942%	15.409%
Recall	84.27%	83.36%	85.51%	83.53%	85.65%	84.22%	90.14%	92.62%	90.88%
Cohen's kappa	0.788%	0.774%	0.792%	0.791%	0.663%	0.794%	0.161%	0.624%	0.681%
F1 score	90.03%	89.22%	90.34%	90.01%	90.37%	90.24%	72.94%	85.39%	87.04%

The name of the final selection classifier is GBT (Gradient Boosting Tree) for the following reasons:

1. The model performs well in several indicators, especially with high accuracy and

precision, and also receives high scores when uploading the model's prediction data to Kaggle, this means that it can make accurate predictions on the sample.

2. This model does not require too much pre-operation, runs fast and has strong stability
3. The test results obtained by the model trained using GB T use the chart to view and find that the data distribution is the most uniform, so it proves that the prediction ability is better.

7. Kaggle submission

The model uploaded to Kaggle uses a gradient boosted tree, in the process of training the model, we first use the median instead of missing values to process data of type int and double, filter the data with little impact on the data with a filter node, use Z-Score to standardize the data to reduce the impact in the data weight, and then adjust the parameters to obtain the best accuracy, accuracy and recall.

8. Conclusion

In this assignment, we use the model and algorithm learned in class to train the model to predict whether the house is qualified or not. In the exercise, I deeply realized that if you need to obtain good prediction ability, you need to preprocess the data, including cleaning irrelevant data, and dealing with outliers and missing values. In the process of model parameter tuning, I also realized that the correct rate of running on the machine does not represent the correct rate of the predictive model, and there may be situations such as overfitting, which require continuous trial and in-depth research.

Through practice, I also found that if you want to obtain a model with high predictive ability, you need to have a clear understanding of the data set, including the distribution of data, outliers and the output of predicted samples, and in the process of data analysis, we should also be good at applying various visualization tools to use visualization to discover and study data.

In the future, I will continue to explore more data analysis tools in the process of continuous learning, so as to better and more comprehensively train better and better models.

Reference

Arain, F. N. (2021, April 4). Decision Tree Classification Algorithm» DevOps. DevOps. https://www.devops.ae/decision-tree-classification-algorithm/

BetaE. (n.d.). Snap.stanford.edu. Retrieved May 23, 2023, from http://snap.stanford.edu/betae/

Gamal, B. (2021, April 11). Naïve Bayes Algorithm. Analytics Vidhya. https://medium.com/analytics-vidhya/na%C3%AFve-bayes-algorithm-5bf31e9032a2

Gradient Boosting – What You Need to Know — Machine Learning. (2020, August 5). DATA SCIENCE. https://datascience.eu/machine-learning/gradient-boosting-what-you-need-to-know/

k-Nearest Neighbor (kNN) Classifier - Wolfram Demonstrations Project. (n.d.). Demonstrations.wolfram.com. Retrieved May 23, 2023, from https://demonstrations.wolfram.com/KNearestNeighborKNNClassifier/

Kumar, N. (2021, March 26). Introduction to Support Vector Machines (SVMs). MarkTechPost. https://www.marktechpost.com/2021/03/25/introduction-to-support-vector-machines-svms/

Logistic Regression - Voxco. (n.d.). Retrieved May 23, 2023, from https://www.voxco.com/blog/logistic-regression/

Sharma, H., & Kumar, S. (2016). A Survey on Decision Tree Algorithms of Classification in Data Mining. Www.semanticscholar.org. https://www.semanticscholar.org/paper/A-Survey-on-Decision-Tree-Algorithms-of-in-Data-Sharma-Kumar/93071221663df46568d5e1edf3e0476d1d2422cc

What is a Random Forest? (n.d.). TIBCO Software. https://www.tibco.com/reference-center/what-is-a-random-forest

Wikipedia Contributors. (2019, September 14). Boosting (machine learning). Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Boosting_%28machine_learning%29

Wikipedia Contributors. (2019, October 22). Confusion matrix. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Confusion_matrix