Module Code : CSMDM21

Assignment Report Title : Data Analytics and Mining

Data Analytics and Mining

Department of Computer Science
University of Reading

Assignment Evaluations:

1) Importance of Data pre-processing and understanding in choosing a validation method and a model.
2) Understanding of problem statement while evaluating the performance and accuracy of the model.
3) Importance of choice of parameter or hyperparameter for the model under study.

Task 1: Data Understanding and Preprocessing

Construct a KNIME workflow to understand the data and discuss it in the report, along with your findings. Next, identify the required data preprocessing steps, perform them and report the KNIME workflow constructed to perform these tasks.

Solution:

We load *Breast-Cancer-Wisconsin* dataset into KNIME. To do this we use "File Reader" node. In my device the dataset is saved at location "/Users/ameyadamle/knime-workspace/CSMDM21_30827018_Assignment/_30827018_assignment/data/breast-cancer-wisconsin.data".

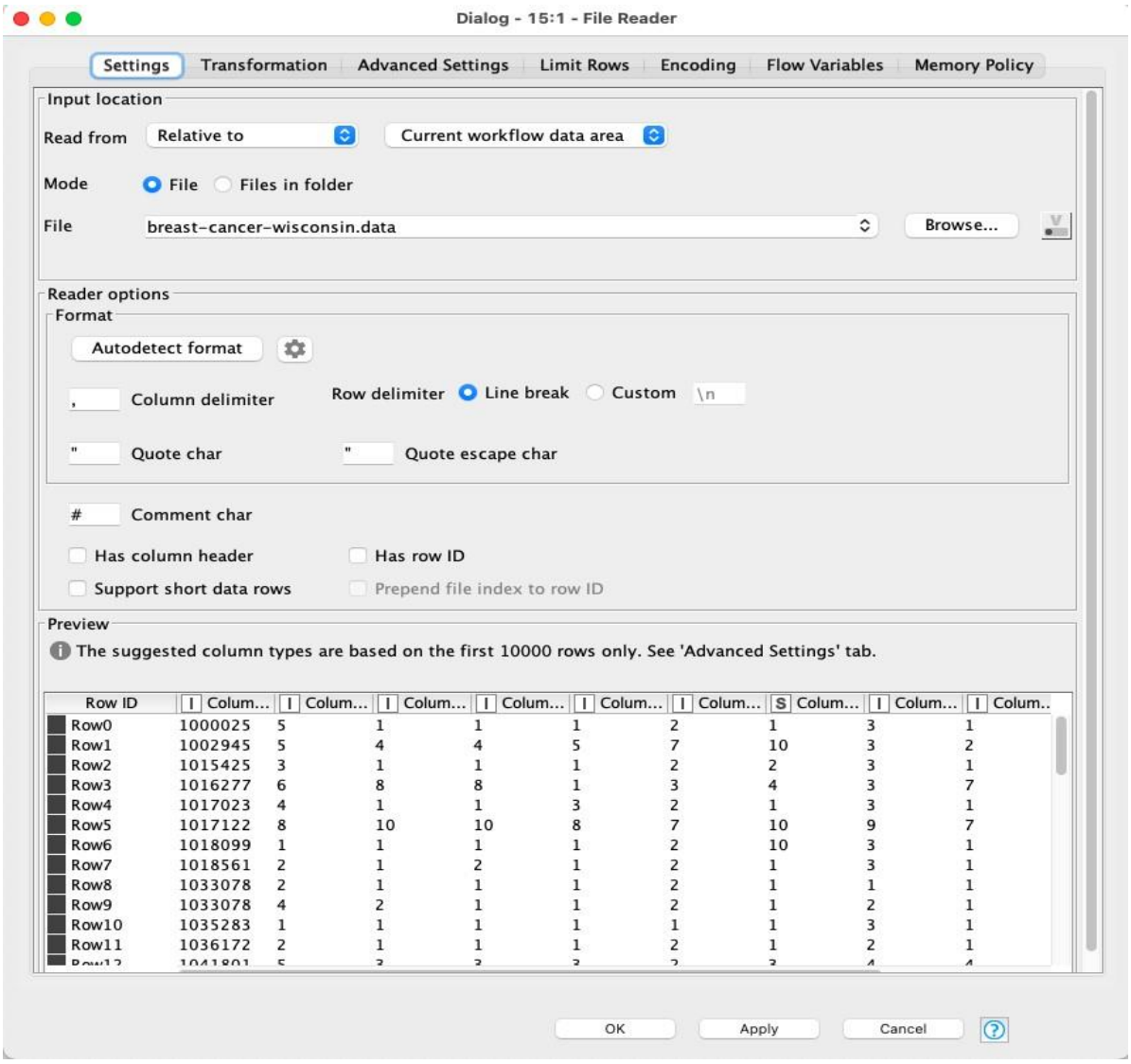


Figure 1 Config of "File Reader"

This dataset does not include column names so we need to read another dataset containing column names using file reader. Column names are saved in a text document at the same location named as "breast-cancer-wisconsin.TXT".

1) We use "Extract Column Header", "Transpose", "Row ID" nodes to create a key to link the column names in the text document to the dataset in data file.
2) Next, we use "Joiner" node to join the column names in the loaded dataset and the column names in text document at correct location using the key created in the previous step.

3) Finally, we use “Insert Column Header” node to insert the column names in the text file to our dataset.

Here we encounter that this dataset consists of 699 records. We can conclude that this is a comparatively small dataset and while choosing a classification model, a model that describes high bias and low variance should. A high bias and low variance model will include more assumptions about the target function. This is necessary as a low bias and high variance model will try to accurately fit the model and due small amount of data it would lead to overfitting. Hence models like Naïve Bayes classifier, Logistic Regressor, Random Forest with less number of trees should be chosen instead of choosing Neural Network, SVM, KNN.

This workflow is converted into a Metanode named as “Column Name Inserter”. Output from node “Insert Column Header” goes further for preprocessing.

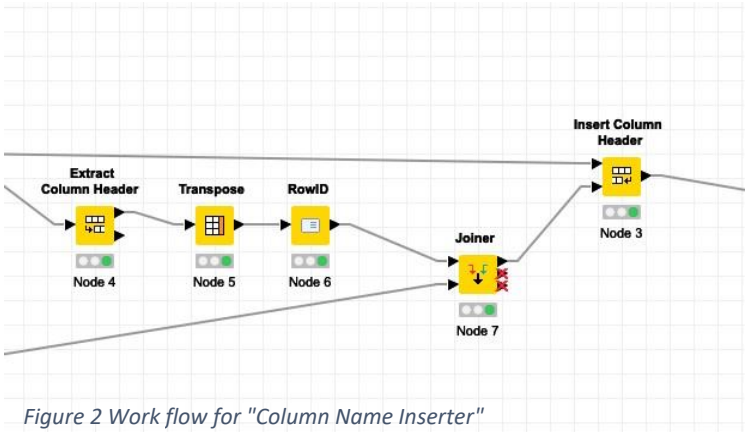


Figure 2 Work flow for "Column Name Inserter"

We use “Duplicate Row Filter” node to remove duplicate row from our dataset. In this case we encounter 8 duplicate rows. In the advance option of this we have an option to choose first, last, min, max duplicate row to remove. We can also retain order.

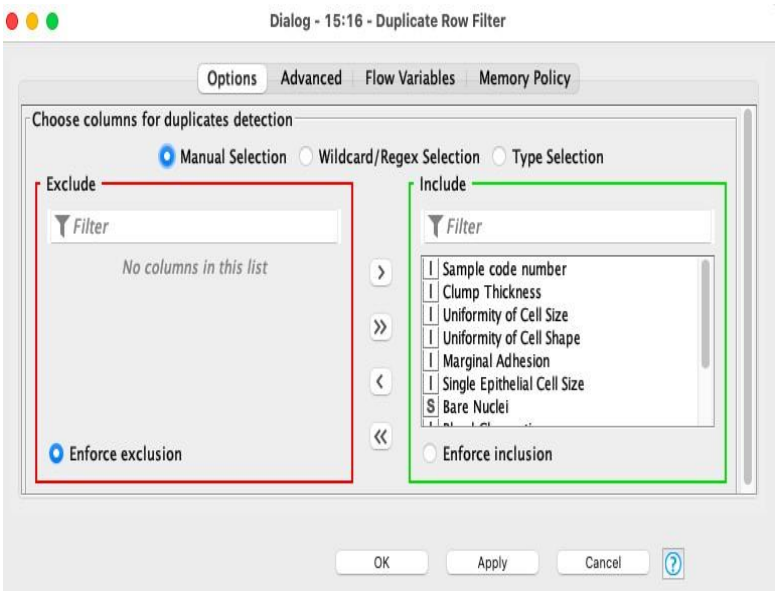


Figure 3 Config of node "Duplicate Row Filter"

In the description file, we can see that there are 16 missing values in our dataset. We have an option to remove these values, or we can impute these values by a value that would be good representation of the missing attribute. To find out the location of these missing values we use “Statistics” node.

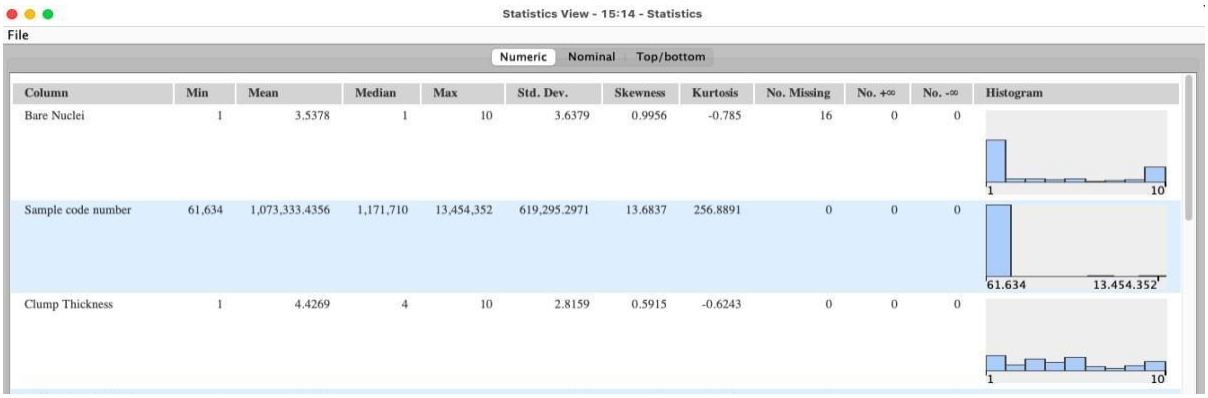


Figure 4 Statistics View

We can see in statistics view of "Statistics" node that there 16 missing values in column "Bare Nuclei".

We use node "Missing Value" to impute rounded mean values in place of the missing values. We can see in statistics view that mean for "Bare Nuclei" column is 3.5378, so rounded mean of 4 is imputed in place of missing values in the "Bare Nuclei" column.

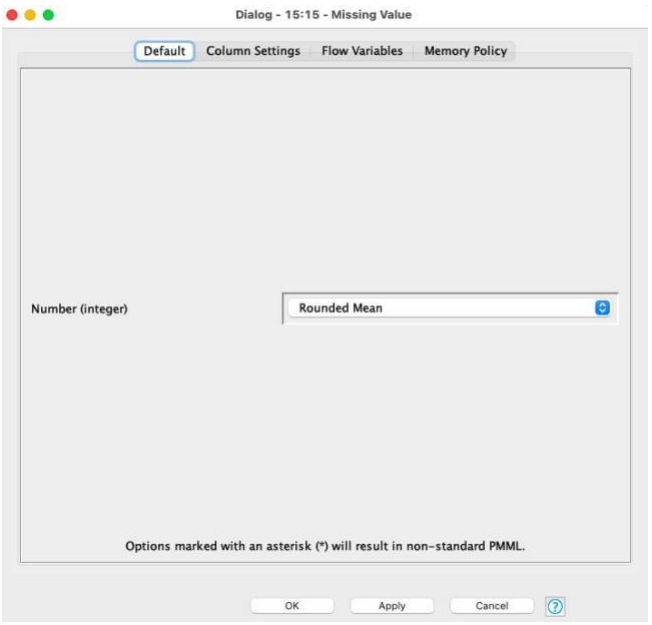


Figure 5 Config of node "Missing values"

We use "Color Manager" node to assign "Class" column value 2 to color "Blue" and value 4 to color "Red". We use a file named Breast_cancer_class_names.csv at our data location as dictionary file to create a "Class Name" column. We use "Cell Replacer" node to add a column named "Class Name" in which we denote class value 2 as "Benign" and class value 4 as "Malignant".

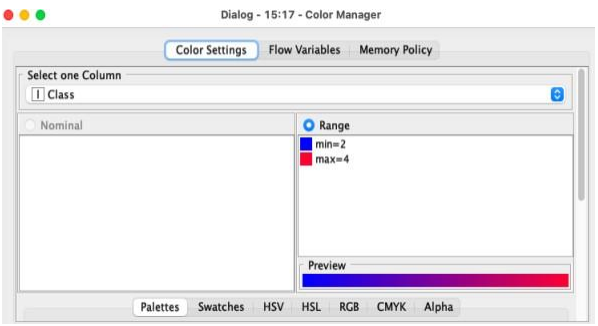


Figure 6 Config of node "Color Manager"

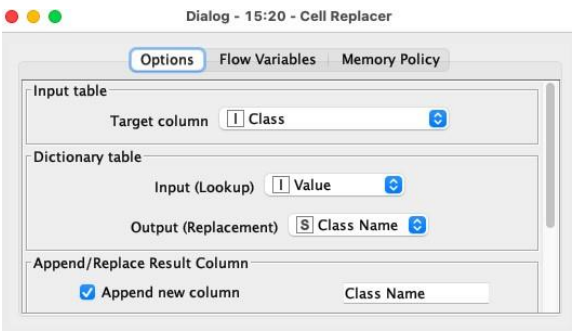


Figure 7 Config of node "Cell Replacer"

Row ID	Sampl...	Clump...	Unifor...	Unifor...	Margi...	Single ...	Bare ...	Bland ...	Norm...	Mitoses	Class	Class ...
Row2	1015425	3	1	1	1	2	2	3	1	1	2	Benign
Row3	1016277	6	8	8	1	3	4	3	7	1	2	Benign
Row4	1017023	4	1	1	3	2	1	3	1	1	2	Benign
Row5	1017122	8	10	10	8	7	10	9	7	1	4	Malignant
Row6	1018099	1	1	1	1	2	10	3	1	1	2	Benign
Row7	1018561	2	1	2	1	2	1	3	1	1	2	Benign
Row8	1033078	2	1	1	1	2	1	1	1	5	2	Benign
Row9	1033078	4	2	1	1	2	1	2	1	1	2	Benign
Row10	1035283	1	1	1	1	1	1	3	1	1	2	Benign
Row11	1036172	2	1	1	1	2	1	2	1	1	2	Benign
Row12	1041801	5	3	3	3	2	3	4	4	1	4	Malignant
Row13	1043999	1	1	1	1	2	3	3	1	1	2	Benign
Row14	1044572	8	7	5	10	7	9	5	5	4	4	Malignant
Row15	1047630	7	4	6	4	6	1	4	3	1	4	Malignant

Figure 8 Table with "Class Name" column and assigned colour

We convert “Class” column from integer to string using “Number To Sting” node for further utilization. For data understanding we use four nodes “Statistics”, “Pie/Donut Chart”, “Box Plot(local)” and “Linear Correlation”.

1) We use “Pie/Donut Chart” node to see division of “Benign” and “Malignant” class. We also observe count and percentage of respective class using pie chart.

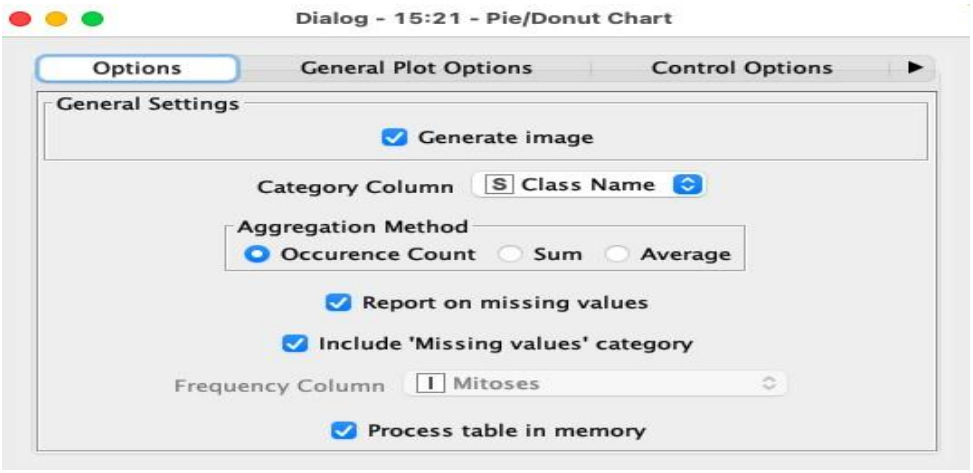


Figure 9 Config of Pie chart

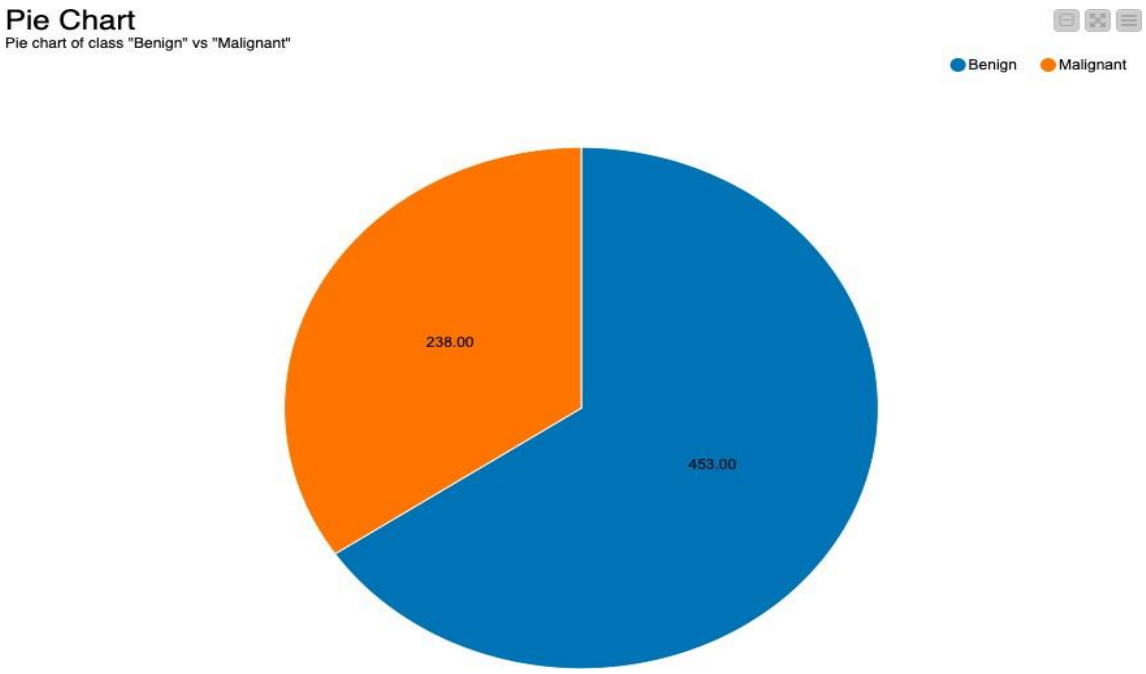


Figure 10 Pie Chart of "Benign" vs "Malignant"

2) We use Box plot to understand behavior of attributes using spread over the range.

Row ID	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses
Minimum	1	1	1	1	1	1	1	1	1
Smallest	1	1	1	1	1	1	1	1	1
Lower Quartile	2	1	1	1	2	1	2	1	1
Median	4	1	1	1	2	1	3	1	1
Upper Quartile	6	5	5	4	4	5	5	4	1
Largest	10	10	10	8	7	10	9	8	1
Maximum	10	10	10	10	10	10	10	10	10

Figure 11 Robust Statistics

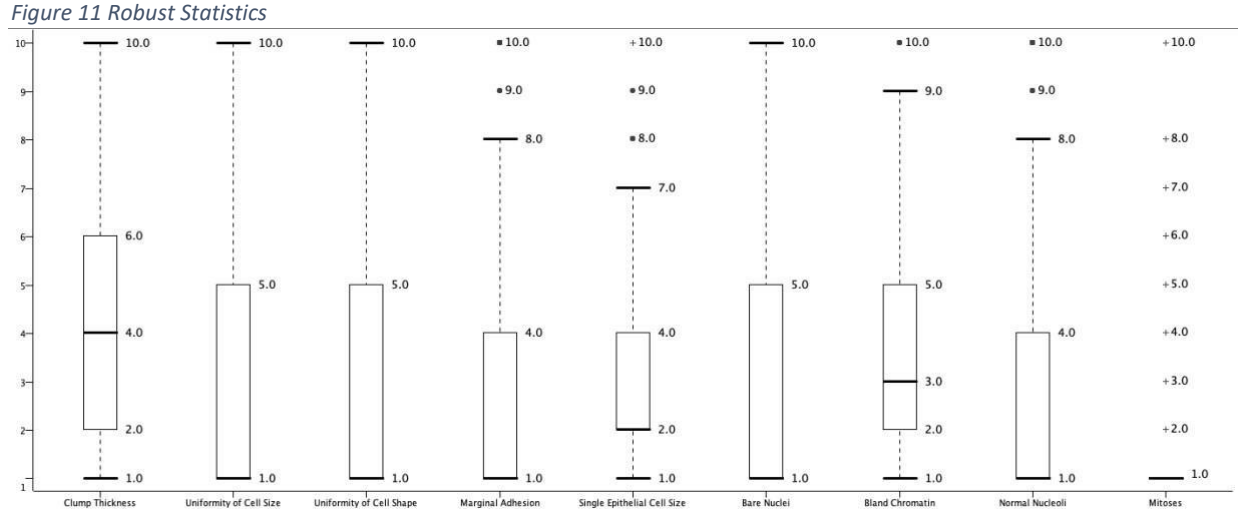


Figure 12 Box plot for dataset

We can observe that some columns have outliers with the column “Mitoses” being most critical.

3) We use “Statistics” node to get more information about the data.

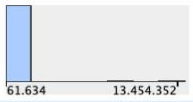
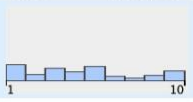
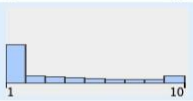
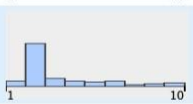
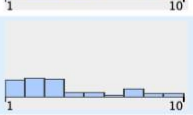
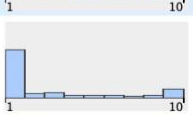
Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
Sample code number	61.634	1,073,333.4356	1,171,710	13,454,352	619,295.2971	13.6837	256.8891	0	0	0	
Clump Thickness	1	4.4269	4	10	2.8159	0.5915	-0.6243	0	0	0	
Uniformity of Cell Size	1	3.1302	1	10	3.0413	1.2332	0.1064	0	0	0	
Uniformity of Cell Shape	1	3.2012	1	10	2.9599	1.1637	0.0185	0	0	0	
Marginal Adhesion	1	2.8249	1	10	2.8666	1.5076	0.9321	0	0	0	
Single Epithelial Cell Size	1	3.2113	2	10	2.1999	1.719	2.2151	0	0	0	
Bare Nuclei	1	3.5485	1	10	3.5961	0.9978	-0.7459	0	0	0	
Bland Chromatin	1	3.4356	3	10	2.4423	1.1028	0.1899	0	0	0	
Normal Nucleoli	1	2.8828	1	10	3.0663	1.4075	0.4264	0	0	0	
Mitoses	1	1.5933	1	10	1.7231	3.5446	12.5134	0	0	0	

Figure 13 Overall Statistics

Here is a tabular representation of statistics like minimum, maximum, mean, median, standard deviation, range etc. for all attributes. In "Statistics" node we get two more result tables.

Nominal histogram table gives histogram plots for each attribute in a tabular format. Occurrence table is table representing value of each attribute, frequency of the values and relative frequency of the values or simply relativities which can be helpful in comparison.

4) Linear Correlation is used to get correlation matrix and correlation analysis of the dataset.

Row ID	D Clump Thickness	D Uniformity of Cell Size	D Uniformity of Cell Shape	D Marginal Adhesion	D Single Epithelial C...	D B
Clump Thickness	1.0	0.6433395836390875	0.6537521438469436	0.48794919871338...	0.51744778444287...	0.58
Uniformity of Cell Size	0.643339583639...	1.0	0.9054195250361641	0.71311700661814...	0.74711119592466...	0.68
Uniformity of Cell Shape	0.653752143846...	0.9054195250361641	1.0	0.69098903788355...	0.71439343394139...	0.70
Marginal Adhesion	0.487949198713...	0.7131170066181426	0.6909890378835559	1.0	0.60847726277002...	0.66
Single Epithelial Cell Size	0.517447784442...	0.7471111959246608	0.7143934339413938	0.60847726277002...	1.0	0.57
Bare Nuclei	0.586827739034...	0.6841088441398816	0.7060808755993517	0.66954527054699...	0.57706293130611...	1.0
Bland Chromatin	0.561076365258...	0.7595252400173924	0.7384546454904078	0.66981270199529...	0.62051763644653...	0.67
Normal Nucleoli	0.535711797895...	0.7272393608711997	0.7246932137323925	0.60245348191129...	0.63405814845689...	0.58
Mitoses	0.350353618115...	0.46006429937155013	0.44059240911094893	0.41716725173032...	0.48264402966214...	0.33

Figure 14 Correlation matrix numerical representation

- We can see that "Uniformity in cell size" and "Uniformity in cell shape" have a very high correlation of 0.90541. While the next maximum correlation measure is about 0.759 between "Bland Chromatin" and "Uniformity in cell size".
- From the summary of all correlation figures we can conclude that 75% of the correlations are less that 0.70.
- Cell size and cell shape seem to be strongly correlated to each other.

- The effect of cell size on class is hard to distinguish from the effect of cell shape on class because increase in cell size is highly associated with increase in cell shape.
- Thus, we get a hint to discard one of the attributes columns for building models as it can be negatively affected by inclusion of highly correlated attributes.

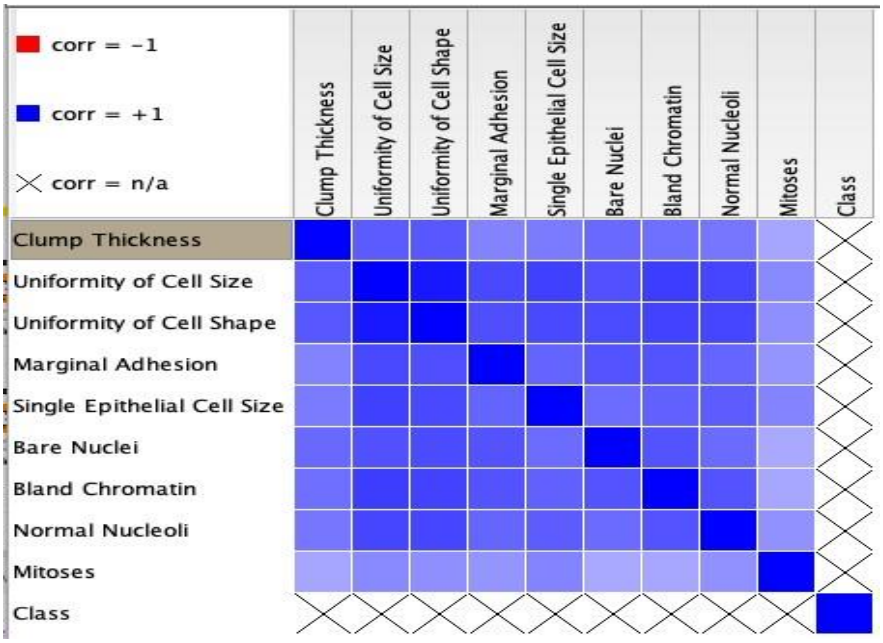


Figure 15 Graphical Representation of correlation matrix

Handling Outliers:

Outliers are data points that fall outside the expected range of values and can have a significant impact on the statistical analysis. Here "Numeric Outlier" node is used to identify and handle outliers.

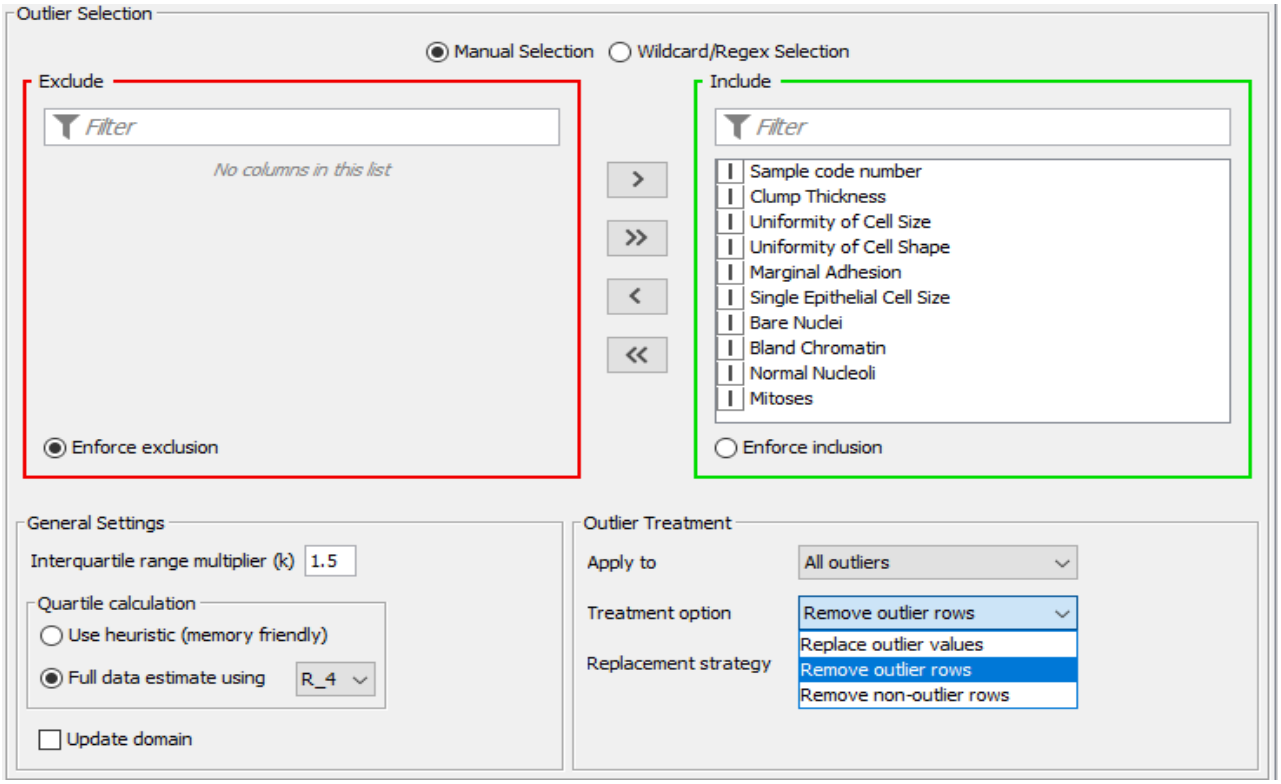


Figure 16 Configuration of Numeric Outlier node

Here Interquartile range is used to calculate upper and lower bounds for handling outliers.

Further you encounter with three options to treat outliers, “Remove Outliers”, “Remove non-outliers” and “Replace Outliers”.

Figure below represents the annotation used for workflow for handling outliers. “Interactive Histogram” node can be used to view count of features after treating outliers.

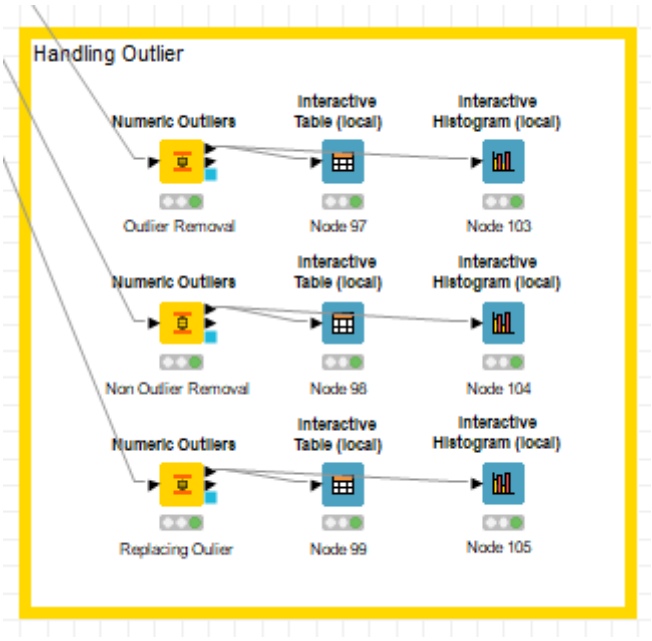


Figure 17 Complete Workflow for handling outliers

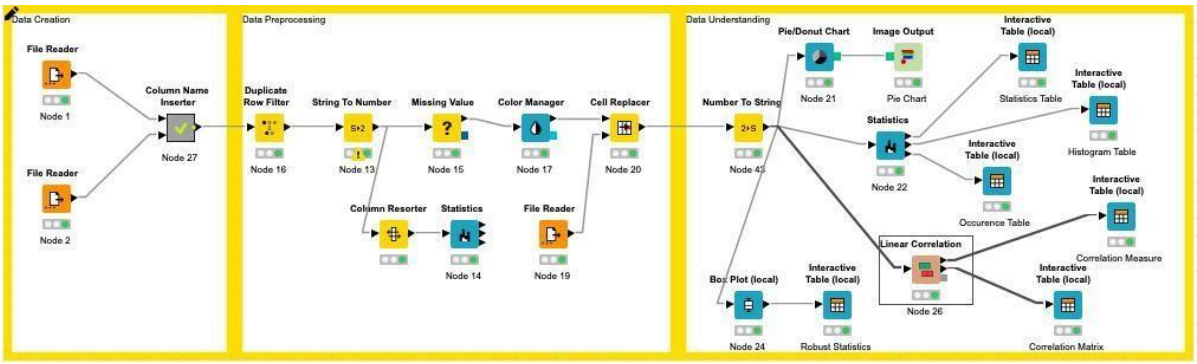


Figure 18 Complete workflow for Task1

Task 2: Classification

After that, construct a KNIME workflow to build at least two classification models using the dataset by experimenting with at least two different algorithms and/or their hyperparameters. You can use any classification algorithms. In the report, describe the adopted algorithms with justifications/discussions. In addition, report the KNIME workflow constructed to perform these tasks, including and explaining relevant node configurations/parameters.

Solution:

1) Random Forest Classification:

Random Forest Classifier is an ensemble learning method that combines multiple decision trees to improve performance and reduce overfitting. It works by creating a large number of decision trees on random subsets of the data and combining their predictions through a voting mechanism RF classifier is generally more accurate than single decision tree and can easily handle high dimensional data. It consumes more time to train and predict than other algorithms. First, we use “X-Partitioner” node to use 10-Fold Cross validation for our model. We set number of validations to 10, choose sampling technique as random and set random seed to 10.

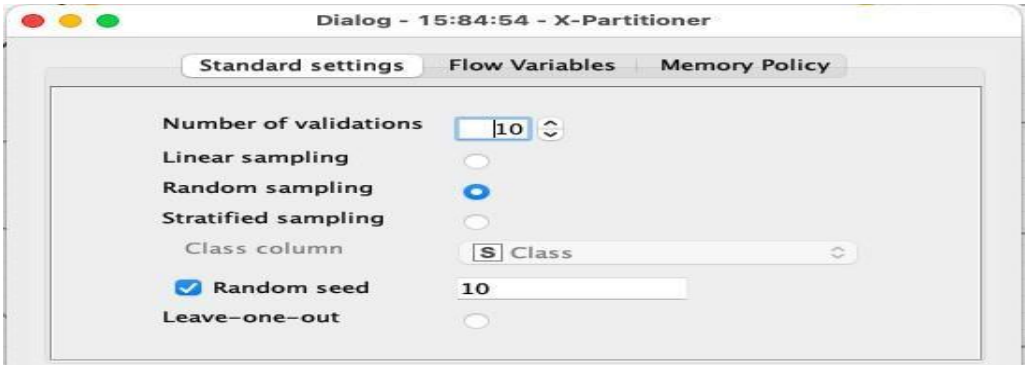


Figure 19 Config for X-Partitioner

Training set goes to node "Random Forest Learner" and Testing set goes to "Random Forest Predictor".

For Random Forest Learner we choose target column as "class". We exclude "cell size" column to avoid multicollinearity. We use split criterion as "Information Gain Ratio" which means attributes with fewer categories are favored.

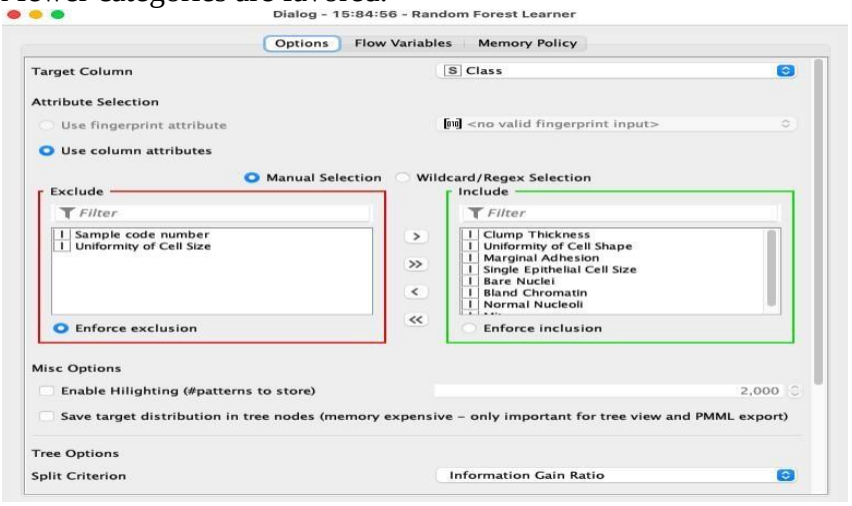


Figure 19 Config of Random Forest Classifier

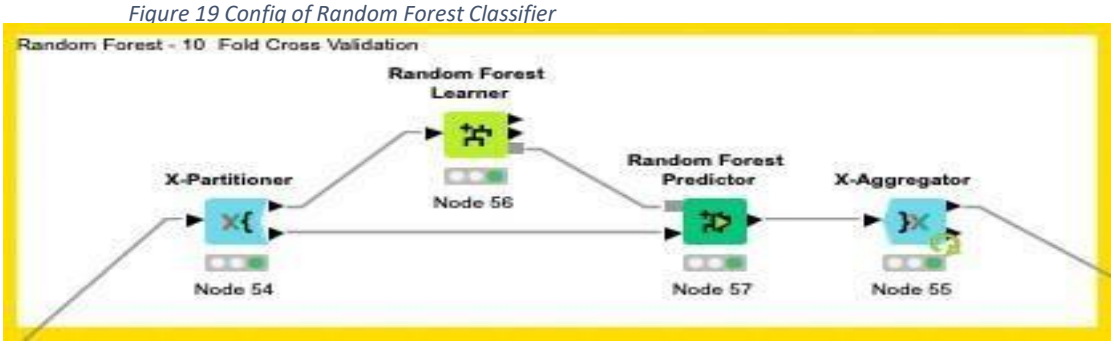


Figure 20 Workflow for Random Forest Classification

2) Decision Tree Classification:

Decision tree can handle both categorical and continuous data and they are easy to understand and explain. However, they are prone to overfitting and can produce unstable results when the tree is too deep.

Same as earlier we use the X-Partitioner to use 10-Fold classification for our model.

We use 10-Fold classification throughout all models as our dataset is not very small so we could afford to carry out multiple random validation set. Leave one out cross validation is generally better way of choosing validation set when the dataset is small, but here we have a considerably large dataset so it would be computationally expensive to use LOOVC.

Class column is "Class", Quality measure is Gain Ratio as there are less number of values among all attributes, Pruning method is no pruning as we do not want large number of levels in trees and minimum number of records per node is 10.

As the spread of the data is small, we prefer Gain Ratio over Gini Index.

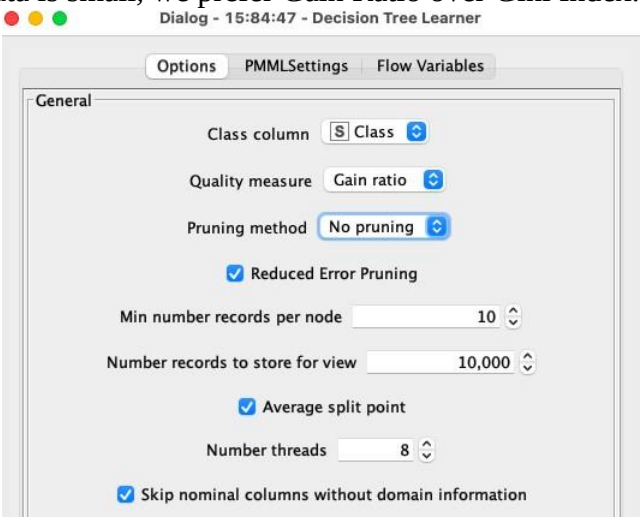


Figure 21 Config for Decision tree

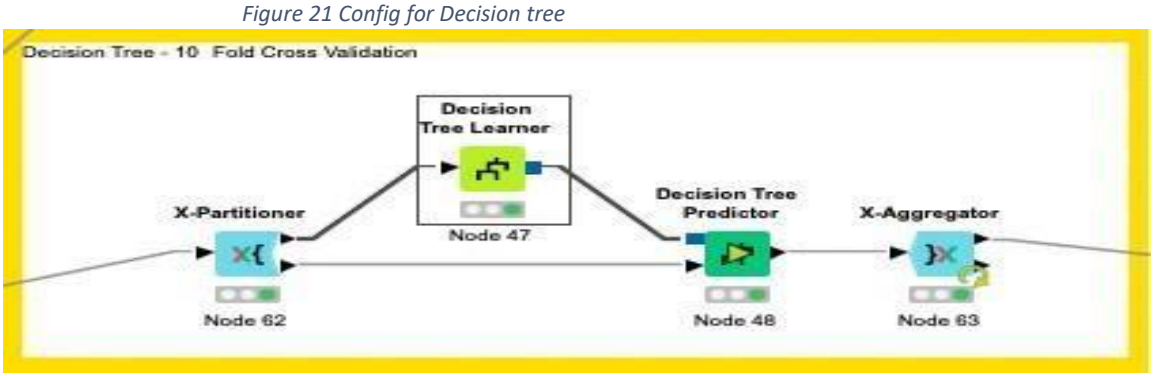


Figure 22 Workflow for Decision Tree

3) Logistic Regression

Logistic regressor is a linear classifier that models the probability of an event occurring based on the input features. It is a simple and efficient algorithm that can be used for both binary and multiclass classification problems. As it assumes a linear relationship between the input feature and the output variable, which may not be true in some cases.

Same as Random Forest, Cell size column is discarded in the model to avoid multicollinearity. Target column is set to class, Reference category is 4 (Malignant) and solver is set to Iteratively Reweighted Least Squares as it finds maximum likelihood estimates.

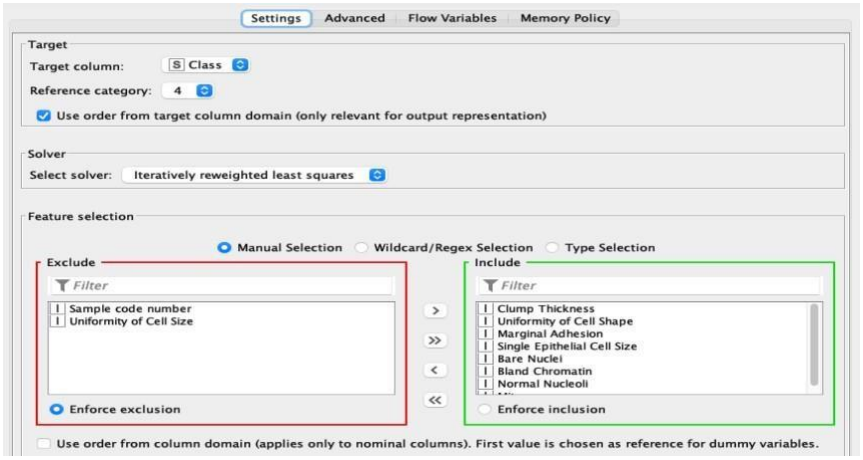


Figure 23 Config for Logistic Regression

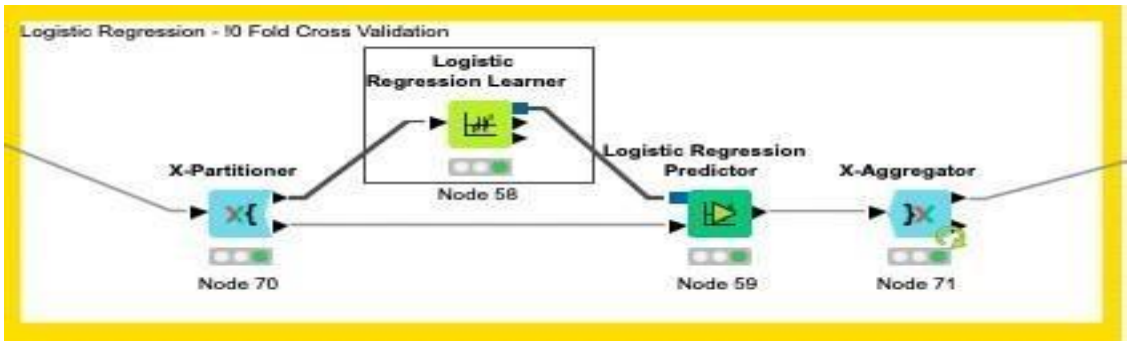


Figure 24 Workflow for Logistic Regression

4) Naïve Bayes Classifier

Naïve Bayes classifier is a probabilistic algorithm that uses Bayes' theorem to predict the class of a new data point. It assumes that input features are conditionally independent given the output variable, which is not always true. Naïve Bayes classifiers are simple, fast and can handle high dimensional data. However, they may not be as accurate as other complex algorithms.

Naïve Bayes does not have any hyper parameter to tune. We keep most of the value in configurations to default.

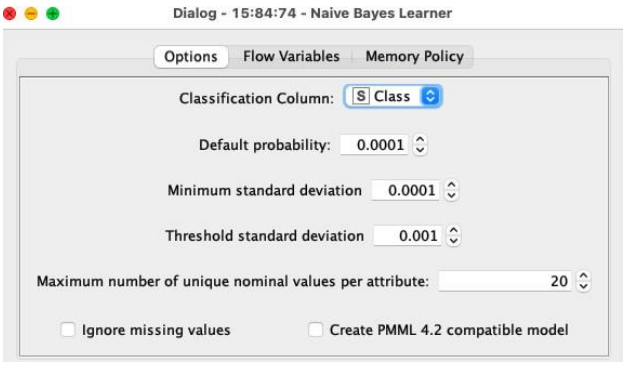


Figure 25 Config for Naive Bayes

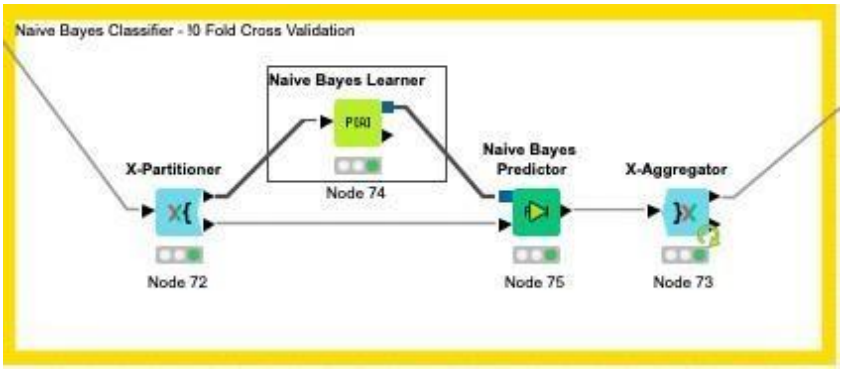


Figure 26 Workflow for Naive Bayes

All four models are combined under a Metanode named "Classification Models" This Metanode is connected to a final Metanode named "Model Evaluation" which contains evaluation statistics.

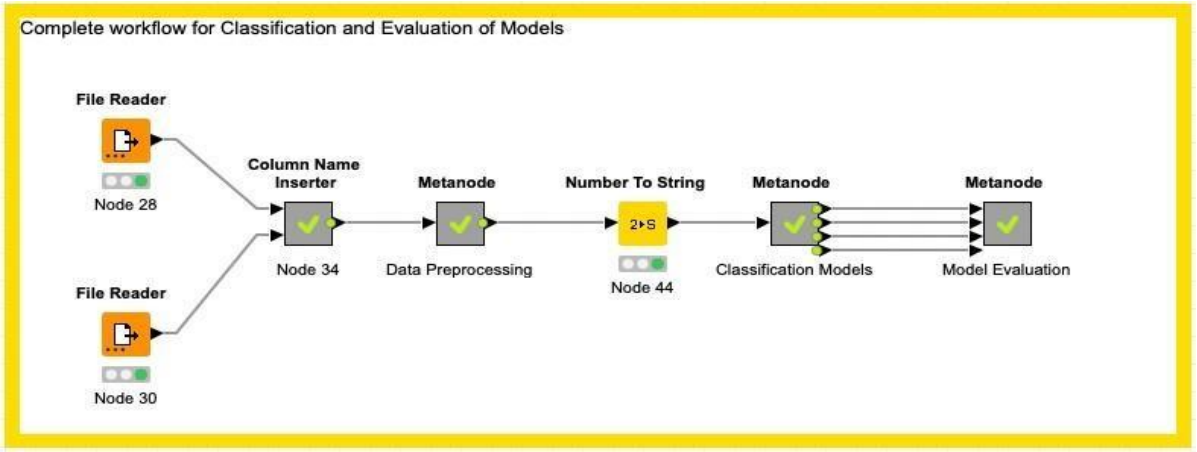


Figure 27 Workflow for classification models and evaluation

Task 3: Model Evaluation

Finally, evaluate the trained models using appropriate performance measures and evaluation methods. In the report, describe and justify the adopted performance measures and evaluation methods, present and discuss the results and their reliability. In addition, report the KNIME workflow constructed to perform these tasks.

Solution:

In this problem we must predict Class as Benign or Malignant, i.e., the cancer tumor cell is harmless or severe. So, there could be four possible cases of accuracy while prediction malignant cell.

True Positive: Cell is Malignant and predicted as Malignant
False Negative: Cell is Malignant and predicted as Benign
False Positive: Cell is Benign and predicted as Malignant
True Negative: Cell is Benign and predicted as Benign

In this case if False Negatives are high then person who has cancer would not be treated correctly and would eventually die due to lack of treatment. Thus, we need to make sure that False Negatives are relatively minimum while evaluating models.

Hence in this case Recall (TP/(TP+FN)) should be higher than Precision (TP/(TP+FP)) as cost of False Negative is higher here.

"X-Aggregator" node calculates combined error and is connected to node "Scorer". Configuration for scorer is same for all models.

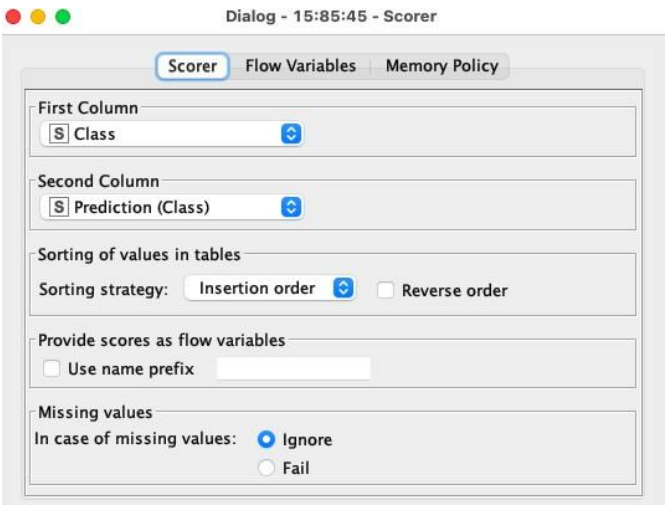


Figure 28 Config for Scorer

Confusion matrix and Accuracy statistics

1) Random Forest.

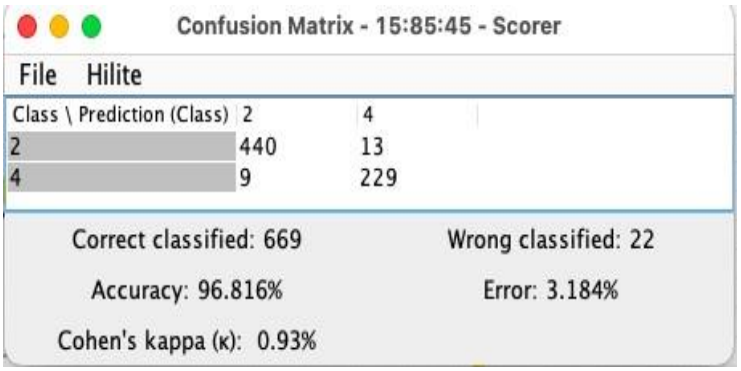


Figure 29 Confusion matrix 1)

Row ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kappa
2	440	9	229	13	0.971	0.98	0.971	0.962	0.976	?	?
4	229	13	440	9	0.962	0.946	0.962	0.971	0.954	?	?
Overall	?	?	?	?	?	?	?	?	?	0.968	0.93

Figure 30 Accuracy Statistics 1)

Total wrong predictions are 22 and false negative in malignant are 9 which is less than the false negatives for benign that are 13. Also recall is higher than precision in malignant and less than that for benign.

Thus, we can conclude that with accuracy 96.816% Random Forest is the best classification technique in this case.

2) Decision Tree

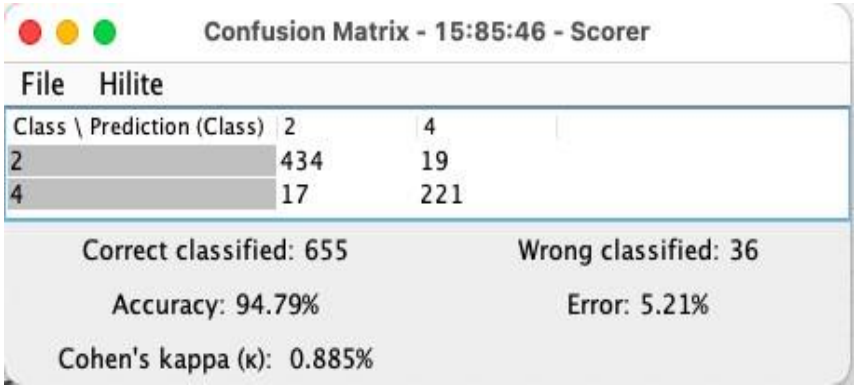


Figure 31 Confusion matrix 2)

Row ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kappa
2	434	17	221	19	0.958	0.962	0.958	0.929	0.96	?	?
4	221	19	434	17	0.929	0.921	0.929	0.958	0.925	?	?
Overall	?	?	?	?	?	?	?	?	?	0.948	0.885

Figure 32 Accuracy Statistics 2)

In this case, wrongly predicted instances are 36, of which 17 are false negatives for malignant and 19 are false negatives for benign. Recall is higher than precision in malignant and less than recall in benign. So model is good but in comparison to Random Forest is has predicted many wrong instances.

3) Logistic Regression

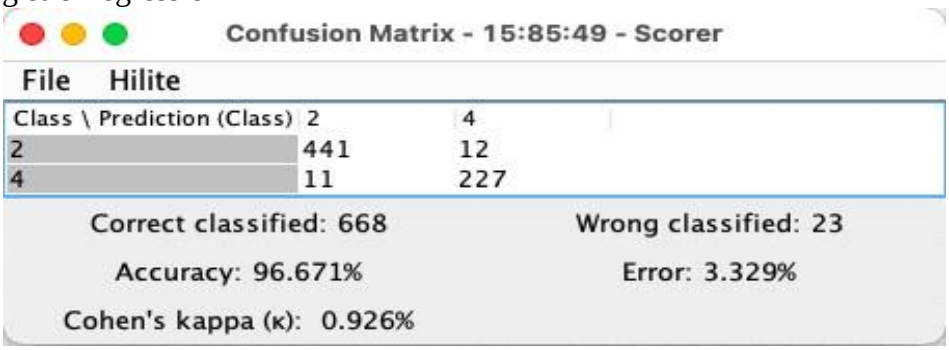


Figure 33 Confusion matrix 3)

Row ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kappa
2	441	11	227	12	0.974	0.976	0.974	0.954	0.975	?	?
4	227	12	441	11	0.954	0.95	0.954	0.974	0.952	?	?
Overall	?	?	?	?	?	?	?	?	?	0.967	0.926

Figure 34 Accuracy Statistics 3)

In this case, wrongly predicted instances are 23, of which 11 are false negatives for malignant and 12 are false negatives for benign. Recall is same as precision in malignant and less than recall in benign. So model is good but in comparison to Random Forest is has predicted many instances of false negative malignant. However, this model is still better than Decision tree.

4) Naïve Bayes Classifier

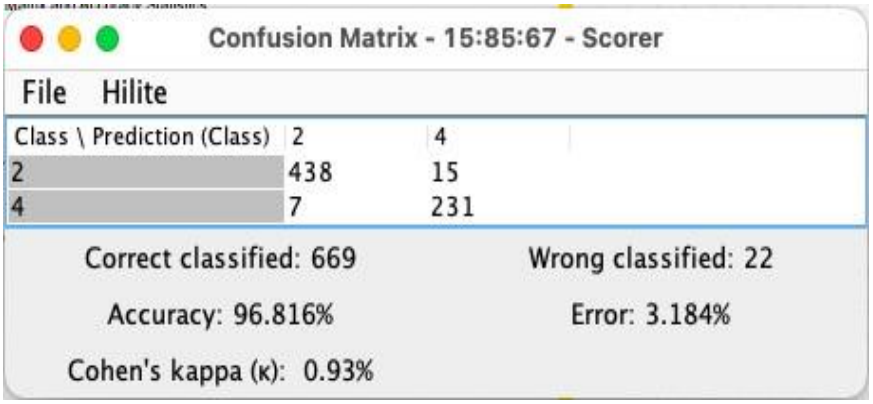


Figure 35 Confusion matrix 4)

Row ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kappa
2	438	7	231	15	0.967	0.984	0.967	0.971	0.976	?	?
4	231	15	438	7	0.971	0.939	0.971	0.967	0.955	?	?
Overall	?	?	?	?	?	?	?	?	?	0.968	0.93

Figure 36 Accuracy Statistics 4)

In this case, wrongly predicted instances are 22, of which 7 are false negatives for malignant and 15 are false negatives for benign. Recall is greater than precision in malignant and higher than recall in benign. So model is not good fit, it might be wrongly or overfit as well.

All the scorer nodes and their interactive tables are combined into a Metanode named Model Evaluation.
For better judgement F-score can also be compared between models to better understand balance between Precision and Recall.

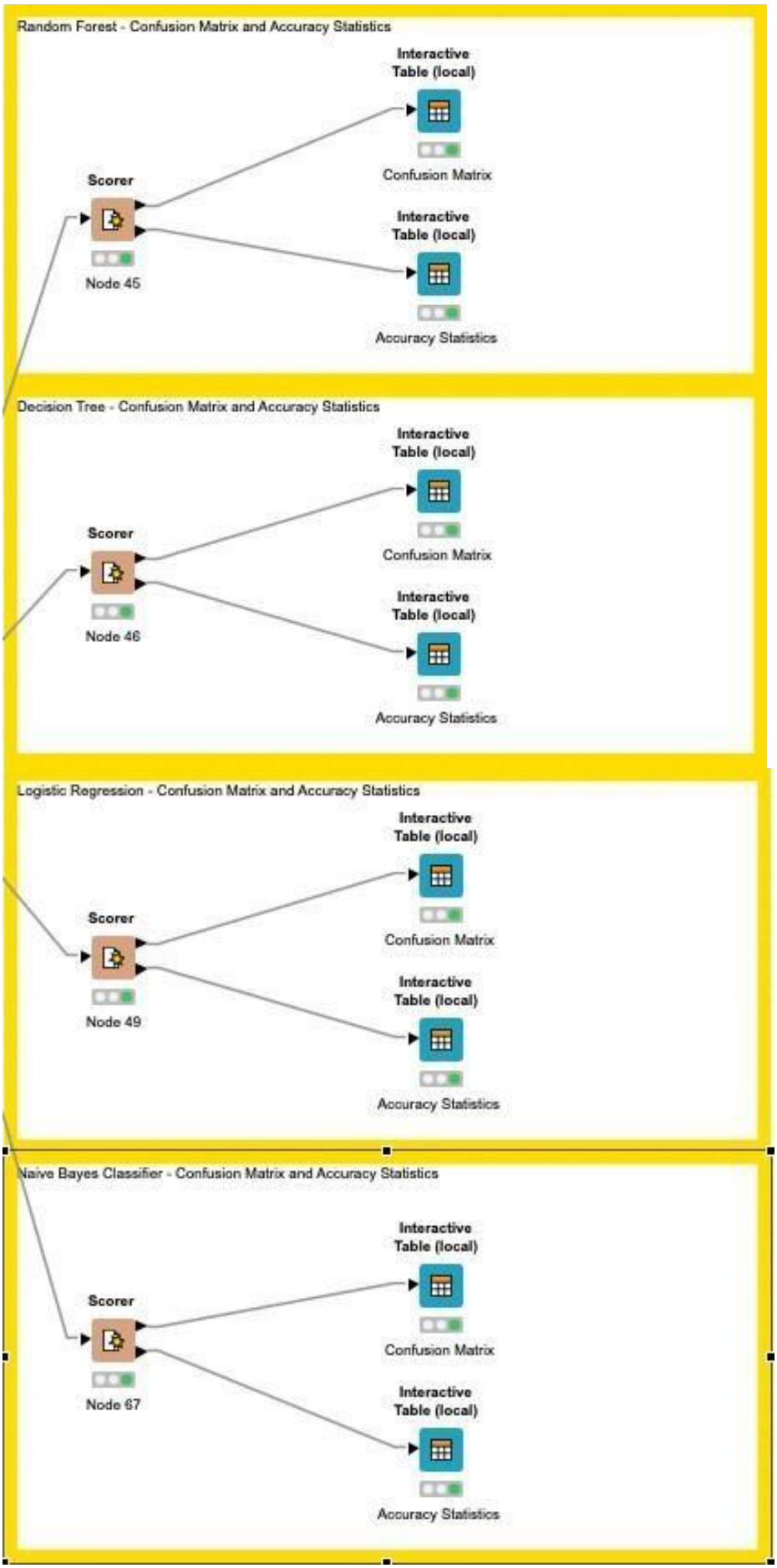


Figure 37 Workflow for Model Evaluation

Conclusion:

- All four models give a good understanding of nodes and interactive elements of KNIME software.
- Using multiple models for solving the same problem give an edge of experience and learning that one can always solve a problem using different models, however the model we choose define the curve of validation and evaluation.
- Decision trees show that, we may be selecting all the parameters correctly, but choice of division of train, test and validation set using different methods such as hold out, leave one out cross validation, re-substitution have a huge impact on accuracy of the model.
- Testing accuracy for different models serves as a chance to evaluate performance of different models apart from choosing models and sampling techniques.
- Survey of evaluation and performance techniques is helpful to calculate accuracy and cost of outcomes on a particular case basis.
- This problem gives a good understanding of data preprocessing, data understanding, a solid approach to working on classification models and comparing accuracies of different models and sampling techniques on same data.

References:

[1] J. Han, M. Kamber, and J. Pei, Data mining: concepts and techniques, 3rd ed. Amsterdam: Elsevier/Morgan Kaufmann, 2012.

[2] C. M. Bishop, Pattern recognition and machine learning. New York: Springer, 2006.

[3] R. Carvalho, "Analysis and Modeling of Breast Cancer Data." rstudio-pubs-static.s3.amazonaws.com https://rstudio-pubs-static.s3.amazonaws.com/205358_1549528d922b4c8793b526b434e9a5e6.html (accessed Nov. 4, 2022)