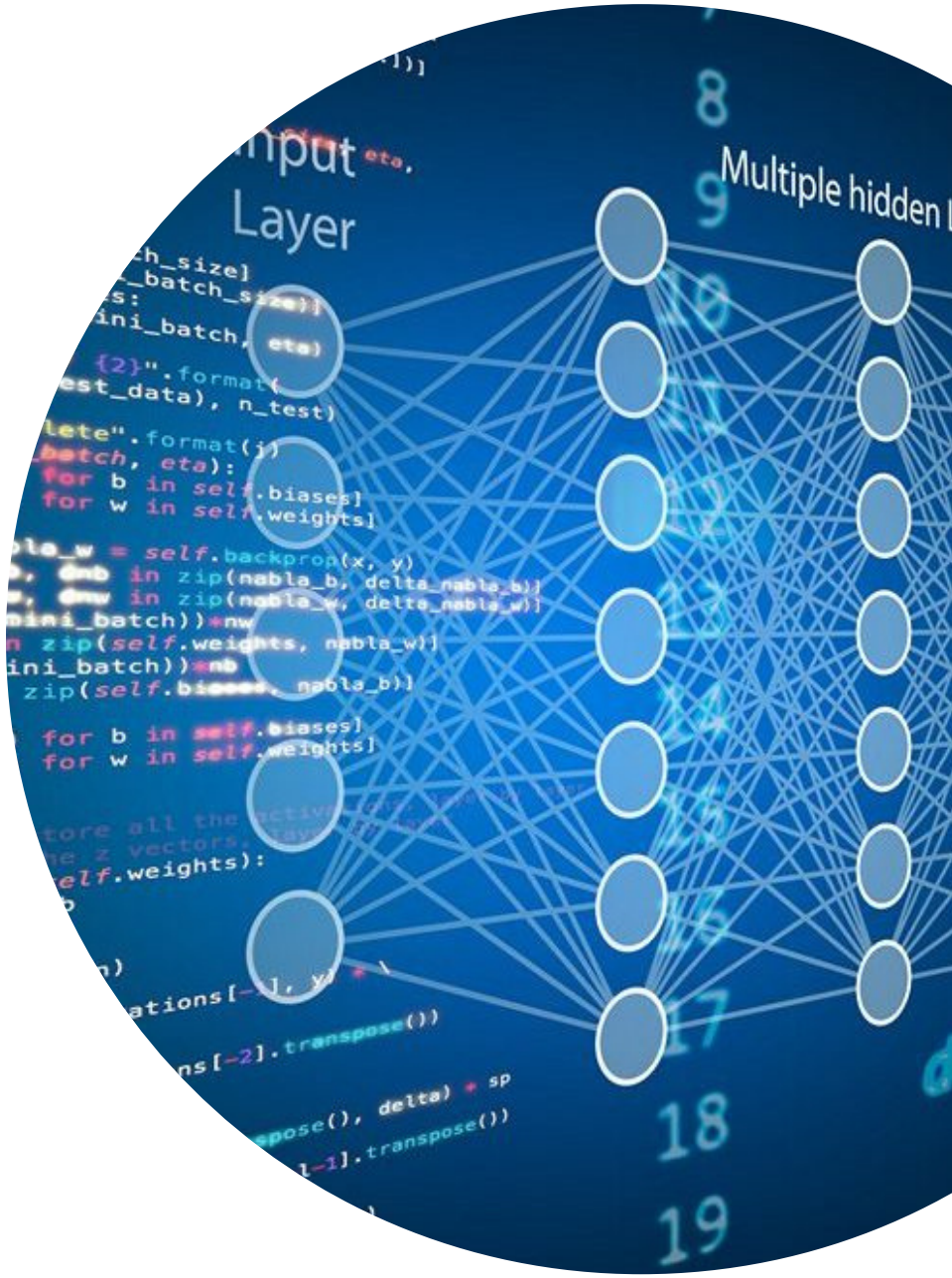
31250 - Intro To
Data Analytics

Assignment 3: Data Mining In Action



ZHITAN WU
(24747735)

Table of Contents

01 - Introduction

The dataset comprises medical details in a hospital, with 47180 records of individual patients and 27 attributes. It likely represents comprehensive patient records for an entire suburban or town population. The attributes range from the patients' gender, age, and the length of days that they have stayed in the hospital. Other traits such as the admitted location, medical department alongside the procedure, and how many chart events they have recorded are also crucial in terms of patient data.

As with data being recorded in extensive databases, data quality issues such as missing values, duplicates, or outliers need to be addressed before any analysis and classification can take place. The primary attribute of focus is 'ExpiredHospital', whereby the objective is to predict the 'future values' as binary values of either a '0' or '1'. The goal of this assignment is to derive and select efficient classifiers by using KNIME's nodes with parameters that does the job thoroughly and accurately. It is to be achieved by training the data from the file 'Assignment3-Healthcare-Dataset.csv' which will form the data in 'Assignment3-Unknown-Dataset.csv' as input and submitting the predictions as output that is akin to 'Assignment3-Kaggle-Submission-Random-Sample.csv'. The nodes that are used as part of the classifiers are scattered throughout this report.

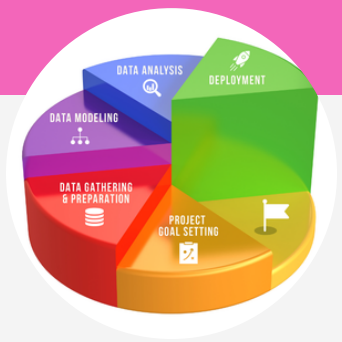
Predicting 'ExpiredHospital' in clinical contexts can allow hospitals to more efficiently optimise their resource allocation and care. Healthcare facilities can conduct their individual research and contribute to medical knowledge while being up-to-date with government protocols and regulations for accurate patient data records. This may also include patient or family communication about end-of-life care. The patients who have the highest mortality should be prioritised in terms of the hospital treatment. Overtime, such analytics may also allow the hospital to see and respect the causes and risk factors involved in patient deaths. They can aid in risk mitigation, infection control by early intervention, or specialised monitoring to improve patient outcomes.

02 - Methodology Overview

A detailed approach to solve the data mining problem starts with data cleansing. This has been conducted with KNIME, and includes aspects such as missing values, binning, column filter and 'number to string'. For each, care was taken according to the nature of the data to choose specific methods that avoided bias while enhancing the interpretability of end results.

After data preprocessing has been conducted to make the values as similar as possible for a fair analysis, it is time to implement classification models. The main classification models that will be used include Decision Trees (DT), K-Nearest-Neighbours (KNN), Random Forest (RF), Support Vector Machines (SVM) and Probablistic Neural Networks (PNN). Other classification models that are included include the Fuzzy Rule (FR), Gradient Boosted Trees (GBT), Tree Ensemble (TE) and Naive Bayes (NB). Other techniques that are included in making an appropriate model include the ROC curve, confusion matrix and using the SMOTE to handle unbalanced data issues. The dataset has been split into the train and test datasets as normal partitions to not introduce complexity that can increase running times.

After conducting these techniques, it is crucial to evaluate which one has the highest accuracy. The model chosen for classification that is the most effective will also ultimately have the highest Kaggle submission score. This will solve the data analytics problem.



'The classification models of Random Forest (RF), Decision Trees (DT), K-Nearest-Neighbours (KNN), Support Vector Machines (SVM) and Neural Networks (NN) are some of the most fundamental models to implement. It is crucial to opt for the model that has the best accuracy among them, and select that as the optimal one for predicting values.'

03 - Data Preprocessing/Transformations (1)

To effectively derive a classifier for the attribute at hand which is 'ExpiredHospital', it is necessary to consider the values of other attributes in the dataset. This is because they may have subtle influence on the predicted outcomes. The attributes to adequately consider for the analysis are 'Age', 'AdmitDiagnosis', 'NumCallouts', 'NumDiagnosis', 'NumProcs', 'AdmitProcedure', 'NumCPTevents', 'NumInput', 'NumLabs', 'NumMicroLabs', 'NumNotes', 'NumOutput', 'NumRx', 'NumProcEvents', 'NumChartEvents', and 'TotalNumInteract'. As part of this, 'LOSdays', 'insurance' and 'religion' may be considered although it may not directly relate to mortality since each patient's condition is distinct. Also, attempts to normalise the related numerical attributes for the tree-based models like Random Forest (RF) has seen to decrease the Kaggle submission score in testing and normalising 'ExpiredHospital' made no sense as the values are already either 0s or 1s. Label encoding worked better than one-hot encoding, and the choice was then made to do that where possible. Some form of normalisation is present for the models such as Neural Network (NN) and Support Vector Machines (SVMs). Parameter optimisation methods that implements the in-built KNIME nodes has also been experimented with, but not in too much detail. Considering these factors, here's an overview of the preprocessing and transformations that has been performed on the data inserted in the csv writer before making the classification models:

A. Missing Values (KNIME)

Missing values should be considered when predicting 'ExpiredHospital' in data analysis and classification. This is because they can jeopardise the outcome of data classification and processing by introducing bias and reducing data integrity. They can also alter statistical properties such as the mean, median, range, variance and standard deviation. The attributes of 'LOSgroupNum', 'age', 'NumCallouts', 'marital_status', 'religion' and 'AdmitDiagnosis' has missing values that will be filled in with data preprocessing. Missing values for the identified attributes have mostly been processed in Excel prior to processing in KNIME with the in-built nodes and has been done on the known and unknown datasets.

B. Auto-Binner (KNIME)

The 'Auto-Binner' node has been used to transform the values of the target attribute which is 'ExpiredHospital'. This may eliminate the possibility of noisy data and overfitting, while possibly enhancing model performance when it discretises and categorizes features and handling missing values in a robust manner. Implementing this into the models has allowed for greater model accuracy alongside higher Kaggle submission scores. It has been done to the known and unknown datasets for consistency in analysis and classification.

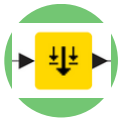
03 - Data Preprocessing/Transformations (2)

C. Normaliser (KNIME)

In terms of the attributes, normalisation has been applied to the numerical ones for models such as Support Vector Machines (SVMs) and the Neural Network models (PNN and RProp MLP) to enhance their training stability. It did not work too well with the K Nearest Neighbour model and hence, the decision was made to not include it there. The attributes that will be normalised include 'admit_type', 'NumCallouts', 'age', 'LOSdays' and 'NumLabs'. Generally, normalising values makes them easier to interpret, and less prone to outliers as the relationships of data points are preserved.

D. Domain Calculator (KNIME)

The domain calculator has been used as the commencement node for performing label encoding to the categorical attributes, depending on the type of model. It allows for a range of values to be specified for taking into consideration during the encoding as 'possible values' and the numerical attributes as the possible min-max values to establish a range. This is typically used in conjunction with the 'Category-To-Number' node, a key component that elaborates the label encoding process.



E. Column Filter (KNIME)

The 'Column Filter' in KNIME has been used on both of the known and unknown dataset, after predictions are made by most of the model's respective 'Predictor' node. It filters out the attributes that are not related to solve the data mining problem and also the attribute results after earlier preprocessing has been conducted such as binning. Overall, it plays an essential role to ensure that only 'Predicted-ExpiredHospital' remains in the output to be converted into a csv file.

F. Number-To-String (KNIME)

Altering a number data type to a string for attributes is useful for KNIME to identify them as class attributes. This is an integral component of preprocessing that aims to alter 'ExpiredHospital' into a data type that is easily recognisable for analysis by other nodes in the classification models. Overall, it generally allows for better efficient manipulation and analysis of data appropriately.

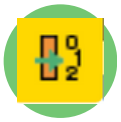
03 - Data Preprocessing/Transformations (3)

G. Partitioning (KNIME)

Partitioning in KNIME is crucial to divide the prescribed data into samples for training and testing purposes. The default configuration is to set aside 70% of the data for training and the remaining 30% for testing. However, it has been decided that to maximise the way that the model recognises patterns for predicting future values, 90% of the data is used for training and the remaining 10% is for testing. It has also been configured to use the random seed for reproducibility, consistency, and for the ease of comparison as each model is tested on the same partitions.

H. String - To - Number (KNIME)

The 'String-To-Number' node in KNIME converts the 'Predicted-ExpiredHospital' values from ones that are recognised as strings to a suitable numerical format that is typically a double. This will ensure that the csv writer at the end will recognise them as numbers when constructing the csv file while eliminating any potential errors. The finalised csv file will have the predictions done by the model in a neat and ordered presentation.



I. Category - To - Number (KNIME)

The 'Category-To-Number' node acts as the second node for label encoding to be performed on the known dataset with the train and test. It allows the categorical or nominal data to be converted to numbers based on their characteristics and features. For example, newborns may be converted to a '0' and 'Pneumonia' may be converted to a '3'. This will ensure that all of the data that is present for analysis will be of a similar format and that precious data would not be lost.

03 - Data Preprocessing/Transformations (3)

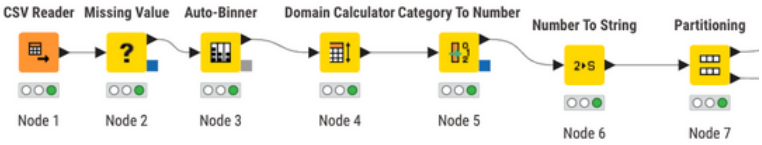
J. Microsoft Excel Functions

Some of the functions are easier to do in Microsoft Excel, particularly since it has been found that some newborns are classified as married. The patients who have an 'AdmitDiagnosis' as 'newborn' should all be aged 0 years old and the people who are less or equal to 18 years old should all be single and not married at all. A function was also used to calculate an appropriate 'LOSgroupNum' for each patient, which is:
=IF(AND(C2>=0, C2<=3), 1, IF(AND(C2>=4, C2<=7), 2, IF(AND(C2>=8, C2<=12), 3, IF(C2>12, 4, ""))))).

All of these are also conducted with the prescribed unknown dataset after some similar preprocessing has been done to the data.

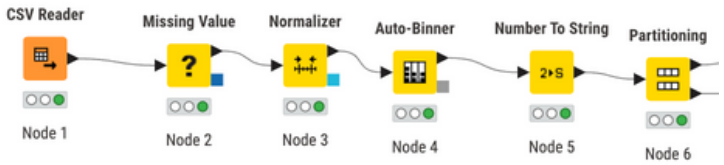
03 - Data Preprocessing/Transformations (3)

Preprocessing Nodes - Known Dataset (For RF, DT, Tree Ensemble, Naive Bayes, Gradient Boosted Trees (GBT) and Fuzzy Rule)



These are the main preprocessing nodes that has been implemented for the models of Random Forest, Decision Tree, Tree Ensemble, Naive Bayes, Gradient Boosted Trees and Fuzzy Rule. They include filling in the missing values that are present in the numerical and string attributes that cannot be done appropriately by Microsoft Excel and binning into suitable numbers. Normalising is generally not needed for the tree-based algorithms such as Random Forest and Decision Trees. Naive Bayes typically uses text classification and Fuzzy Rule uses linguistic variables, meaning that normalisation is not required for them to function. If normalisation was to be performed, loss of information may occur and the accuracy of data may decrease. The model also attempts to do label encoding after the domain has been calculated, and alters the target of 'ExpiredHospital' into a string for it to be recognised as a class attribute. Then, it partitions the data into the training and test sets for suitable analysis with the random seed checked for reproducibility.

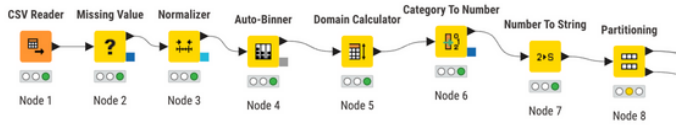
Preprocessing Nodes - Known Dataset (For SVM)



These are the main preprocessing nodes for Support Vector Machines, or SVMs. This does not have any encoding to make the computation time shorter to produce results. However, most of the other preprocessing techniques are present, for example dealing with the leftover missing values that Microsoft Excel cannot fix and performing min-max normalisation to the data for a more efficient analysis and classification. Then, it partitions the data into the training and test sets for suitable analysis with the random seed checked for reproducibility.

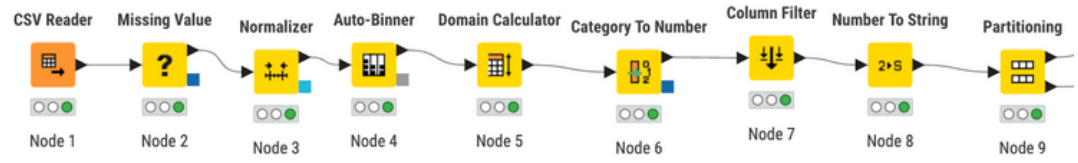
03 - Data Preprocessing/Transformations (3)

Preprocessing Nodes - Known Dataset (For KNN and PNN)



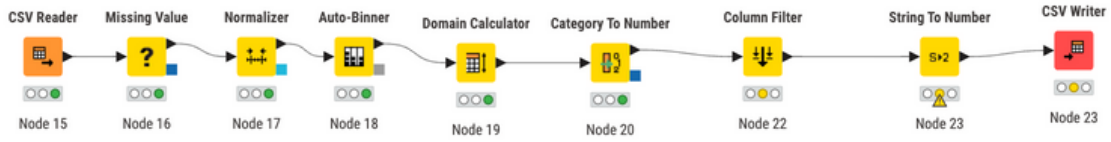
These are the main preprocessing nodes for the models of K-Nearest Neighbour, Probablistic Neural Network (PNN) and MLP. In this case, the values of the unknown dataset are first dealt with by filling in the missing values that Microsoft Excel did not complete. It has label encoding, which may assist with the distance metrics of KNN and the input features that allow the networks to function as intended for models such as PNN and MLP. Binning helps to smooth the values to reduce the impact of noise or outliers, while normalising ensures that all metrics are treated equally with training being sped up and convergence being enhanced. The target attribute then gets converted to a class attribute with the 'Number-To-String' node. It partitions the data into the training and test sets for suitable analysis with the random seed checked for reproducibility.

Preprocessing Nodes - Known Dataset (For MLP)



For MLP, it is very similar to the topology for KNN and PNN. However, a 'Column Filter' is needed due to the fact that the MLP learner only accepts 'Number (double)' values and not any other integer data type. After that, the 'Number-To-String' node is used to make the target a class attribute for the model to perform classification and analysis while partitioning the data into the training and test sets with the random seed for reproducibility.

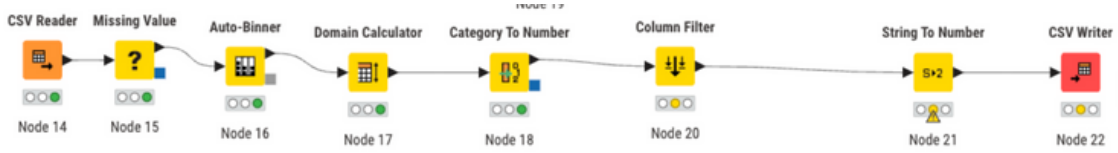
Preprocessing Nodes - Unknown Dataset (For MLP, PNN and KNN)



These preprocessing nodes are made to match the techniques used in the train and test. In this case, the values of the unknown dataset are first dealt with by filling in the missing values that Microsoft Excel did not complete. They are later normalised and binned before the categorical attributes are converted to numbers. The column filter's purpose is to get rid of all of the columns apart from the target attribute column, which may be named as 'Prediction (ExpiredHospital)' or equivalent. These predictions can be exported and saved as a csv document for later reference.

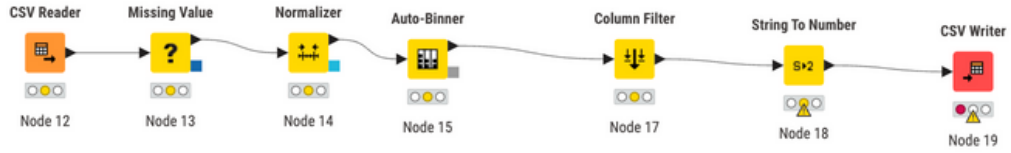
03 - Data Preprocessing/Transformations (3)

Preprocessing Nodes - Unknown Dataset (For Naive Bayes, DT, RF, Tree Ensemble and GBT)



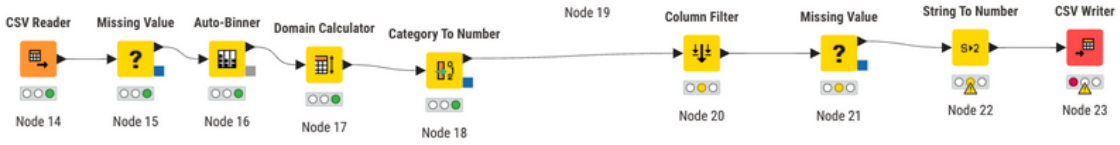
These preprocessing nodes are made to match the techniques used in the train and test. In this case, the values of the unknown dataset are first dealt with by filling in the missing values that Microsoft Excel did not complete. They are binned before the categorical attributes are converted to numbers. The column filter's purpose is to get rid of all of the columns apart from the target attribute column, which may be named as 'Prediction (ExpiredHospital)' or equivalent. However since this is only for tree-based models, normalisation is not very crucial as a component of it. These predictions can be exported and saved as a csv document for later reference.

Preprocessing Nodes - Unknown Dataset (For SVM)



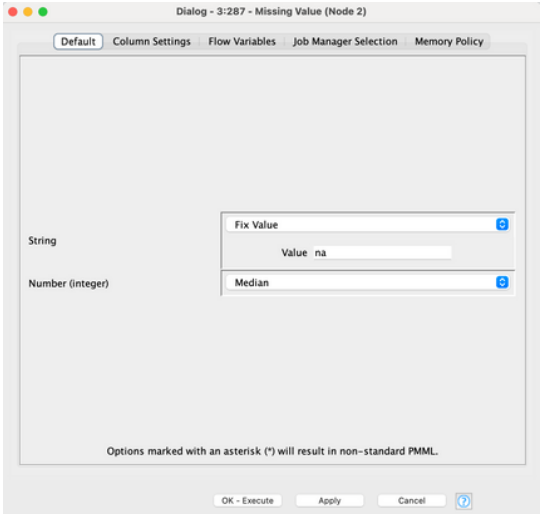
These preprocessing nodes are made to match the techniques used in the train and test. In this case, the values of the unknown dataset are first dealt with by filling in the missing values that Microsoft Excel did not complete. They are normalised before the categorical attributes are converted to numbers. The values are binned to smooth it out, eliminating the potential severe impacts of outliers or noise. The column filter's purpose is to get rid of all of the columns apart from the target attribute column, which may be named as 'Prediction (ExpiredHospital)' or equivalent. These predictions can be exported and saved as a csv document for later reference.

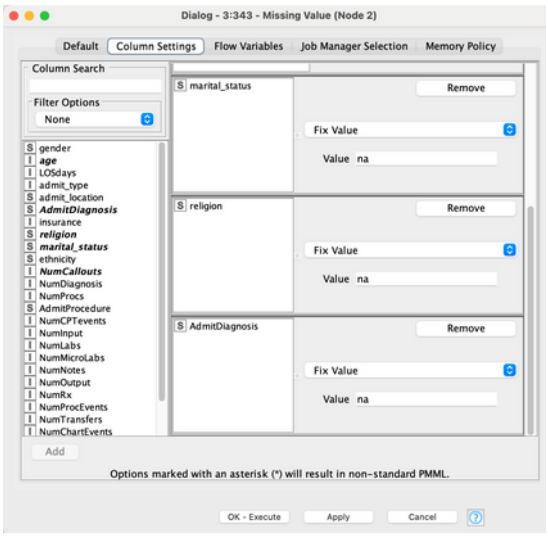
Preprocessing Nodes - Unknown Dataset (For Fuzzy Rule)

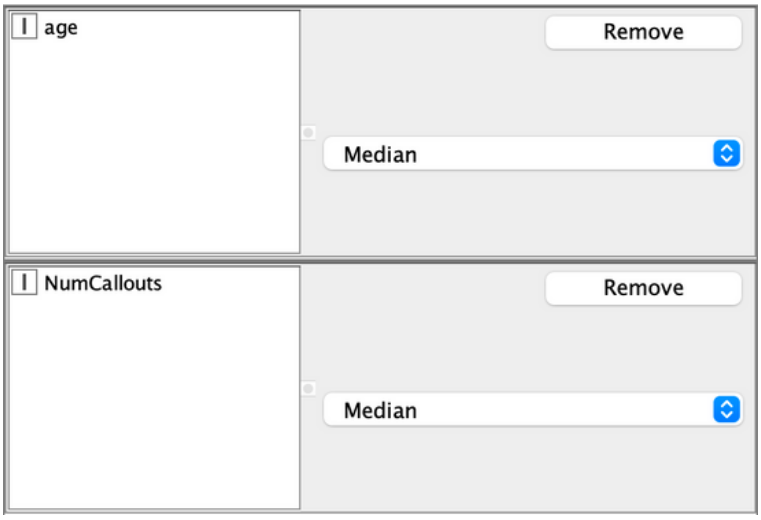


These preprocessing nodes are made to match the techniques used in the train and test. In this case, thevalues of the unknown dataset are binned before the categorical attributes are converted to numbers. The column filter's purpose is to get rid of all of the columns apart from the target attribute column, which may be named as 'Prediction (ExpiredHospital)' or equivalent. However, another 'missing value' node is needed as after the 'Predictor' node has been executed, missing values became present upon filtering out the relevant column. These predictions can be exported and saved as a csv document for later reference.

A. Missing Values (KNIME)







Missing Value Statistics

A3-Healthcare-Dataset

LOSgroupNum - 42447 (89.968207%)
marital_status - 8179 (17.335735%)
NumCallouts - 2370 (5.023315%)
age - 2339 (4.957609%)
religion - 372 (0.788470%)
AdmitDiagnosis - 20 (0.042391%)

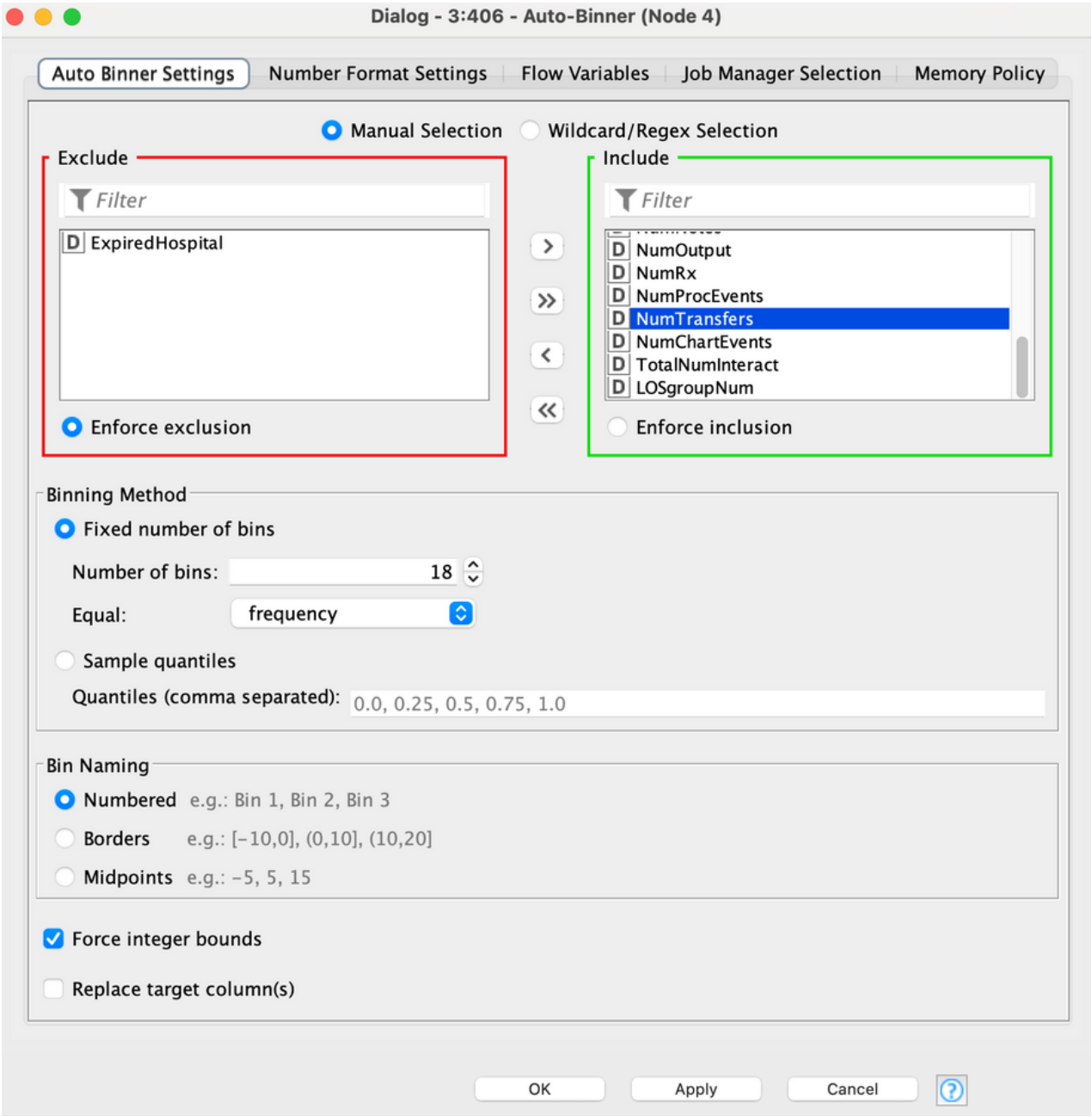
A3-Unknown-Dataset

LOSgroupNum - 10631 (90.123771%)
marital_status - 1949 (16.522550%)
NumCallouts - 578 (4.899966%)
age - 609 (5.162767%)
religion - 86 (0.729061%)
AdmitDiagnosis - 5 (0.042387%)

Missing values are prevalent in a number of attributes, with an overview of the number of missing values also been presented. It is essential to address them so that data analysis and classification can be more effective and streamlined. The top left screenshot illustrates the basic default settings for what strings and number (integer) should be filled in with, particularly if any attributes has been missed. Since string values are not numerical, filling it in with a default statistical measure does not make sense so the decision was made to fix the value by inputting 'na'. The attributes of 'Number (Integer)' have been filled in with the median that is not affected by any outliers or skewed by any extreme values.

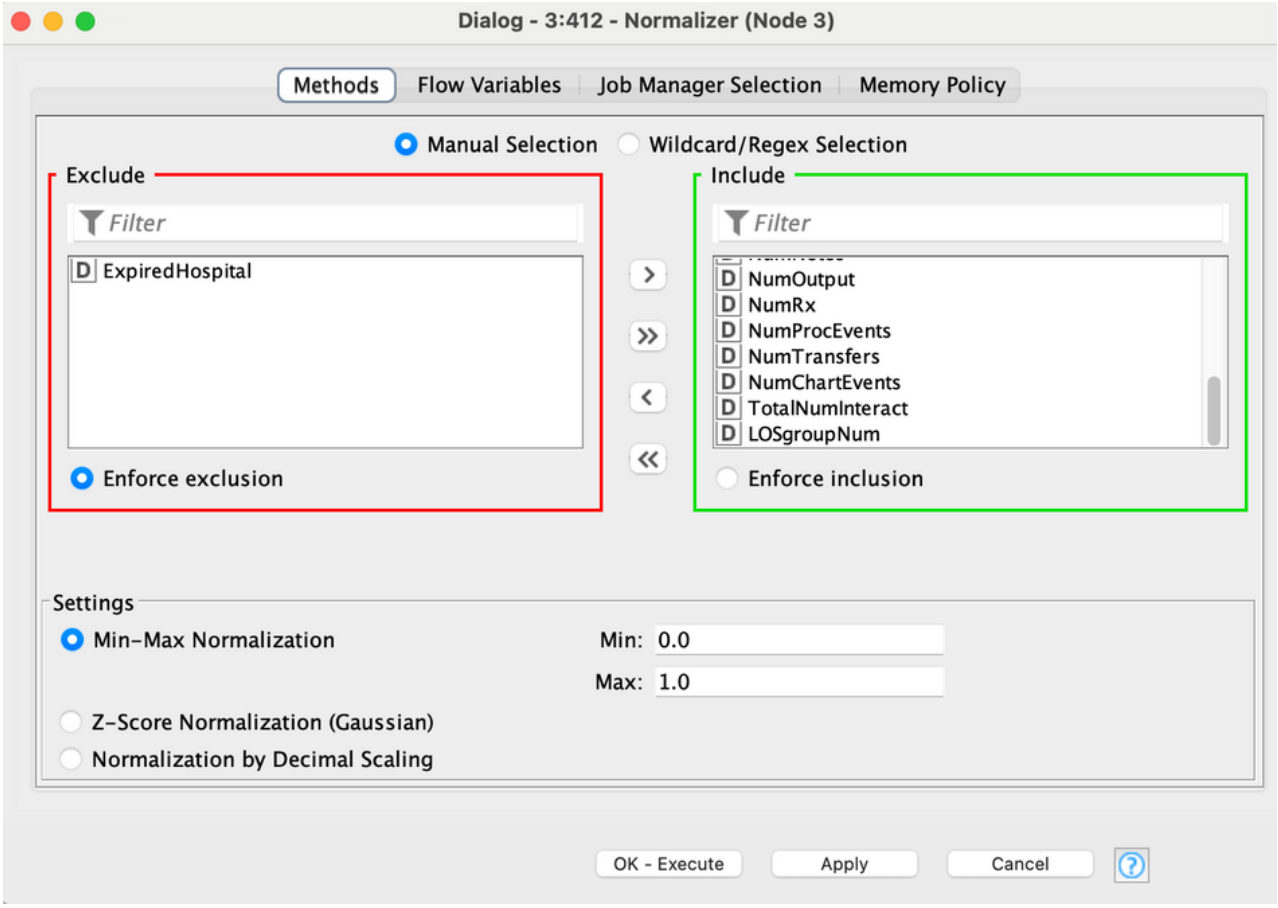
Following on, the values to fill for specific attributes in the dataset has also been defined. The numerical attributes of 'age', 'NumCallouts' and 'admit_type' has been filled with the median that does not get influenced as much by outliers or anomalies in data. The 'LOSgroupNum' has already been previously dealt with an Excel function, so it does not have to be amended here. There does not seem to be any missing values in the 'ExpiredHospital' attribute itself. The decision has also been made to fix the remaining missing values in the categorical (nominal) attributes that include religion and marital status with 'na' to preserve data structure, integrity and clarity.

B. Auto-Binner (KNIME)



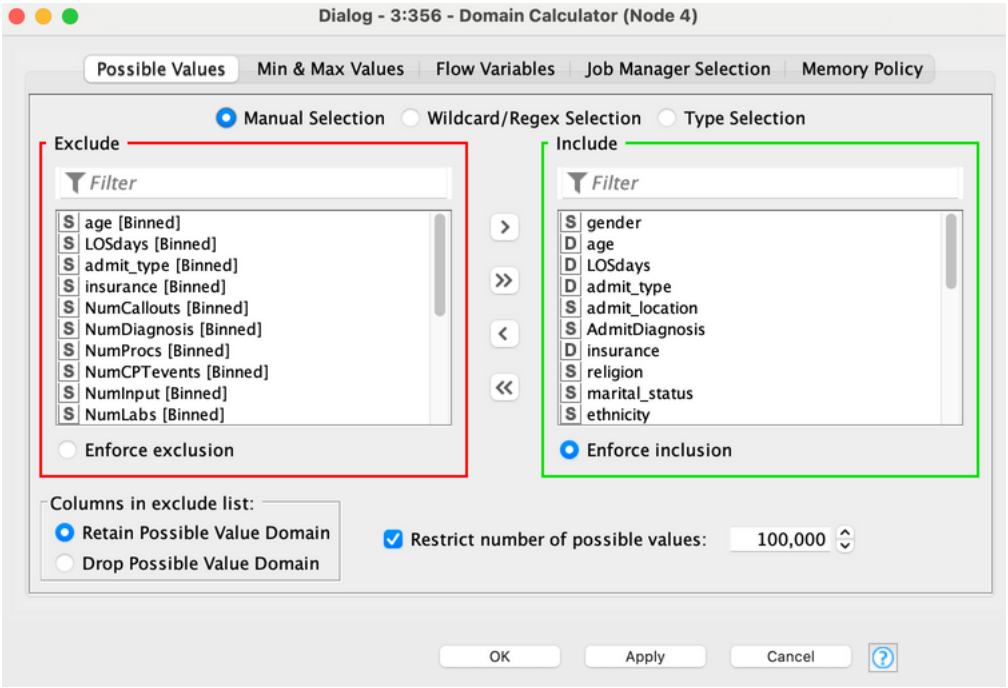
To construct an optimal model for the prescribed dataset, all attributes need to be considered. In this case, they are all of the type 'double', which is a numerical format. From here, it has been decided to use 18 bins with equal frequencies as the intention is to bin a total of 19 attributes that have the 'number (double)' data type. However, 'ExpiredHospital' is not binned as it makes no sense. This is repeated for the unknown dataset as well. The naming of the bins has been set here to 'Numbered' that means that it may be labelled as 'Bin 1', 'Bin 2' and 'Bin 3', but that does not matter too much. The final option to 'force integer bounds' has also been selected to ensure that the binned frequencies are strict and precise.

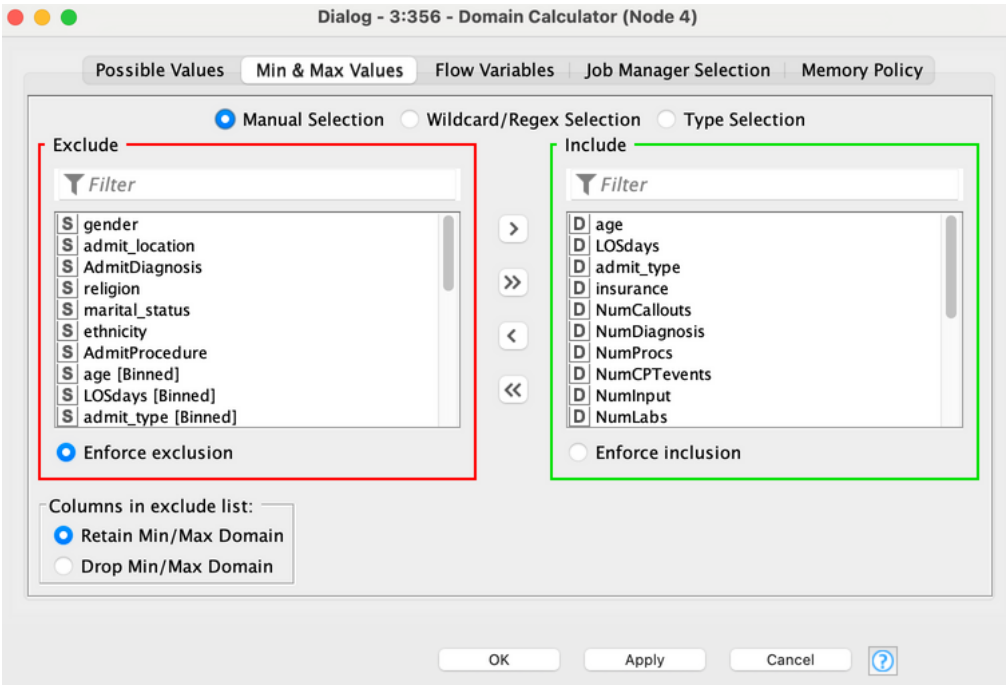
C. Normaliser (KNIME)



For the models of PNN, KNN, MLP, and SVM, normalisation was carried out on the numerical attributes that are either doubles or integers so that any apparent relationships can be made more obvious. This includes the attributes of age, LOSdays, admit_type, insurance, NumCallouts and NumProcs. The target attribute of 'ExpiredHospital' does not need to be normalised as its values are already between 0 and 1. It has also been selected to use 'min-max' normalisation between 0 and 1 for consistency that can lead to more accurate data and quicker output results when running the analysis model.

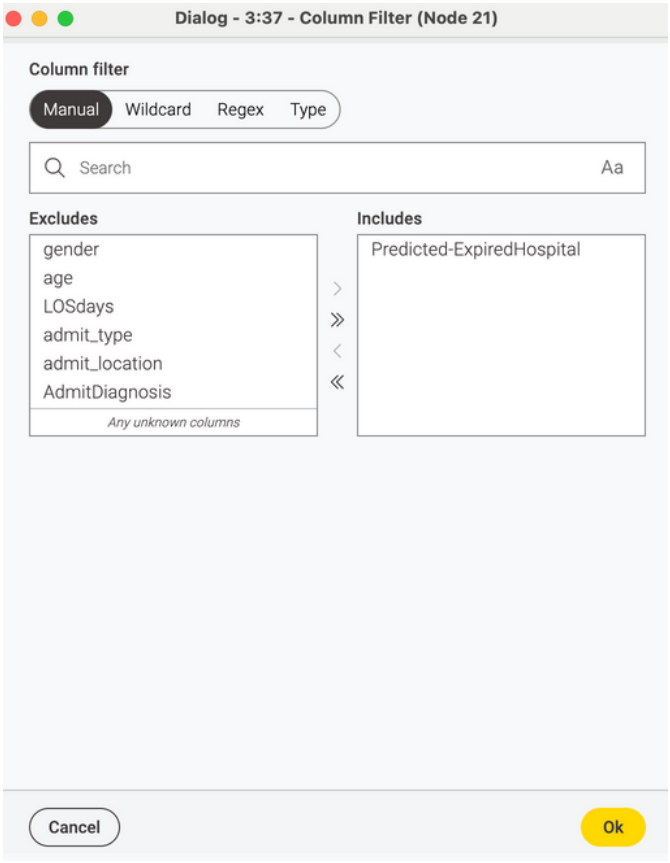
D. Domain Calculator (KNIME)





As part of the process for label encoding using KNIME, the domain calculator has been implemented for preprocessing. All of the attribute columns that has being binned are excluded so that it does not disturb the other data which can be used for analysis. This means that it includes the remaining ones that are numerical and string. For the min-max values, only the numerical attributes are included so that proper maximums and minimums can be sufficiently determined for the encoding process to occur.

E. Column Filter (KNIME)

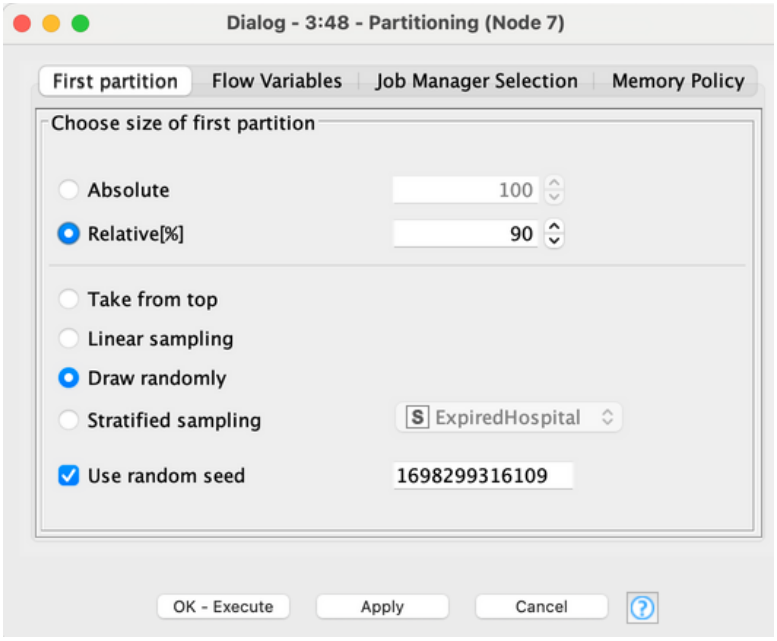


The column filter removes all of the other attribute columns that are a result of preliminary preprocessing for the unknown dataset. It can remove the values that are not of a 'Number (double)' data type, for example 'Number (Integer)', that is not compatible with the learner of models such as MLP. This may include the appended columns as a result of binning or some of the categorical attribute types getting filtered as they are not the primary focus of the data mining problem. When the target attribute column of 'Predicted-ExpiredHospital' has been filtered, it can proceed into the next step of the data preprocessing where it gets converted from a string to a number.

F. Number-To-String (KNIME)

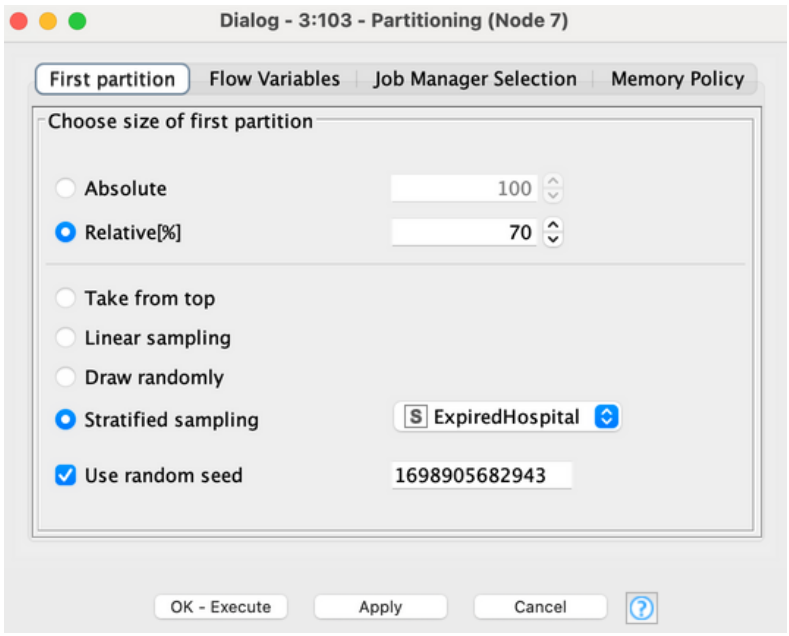
To do analysis with the target attribute of 'ExpiredHospital', it is paramount to alter the data type of it from a number to a string. This method unlocks various data processing options, with it allowing KNIME to recognise its prevalence as a class. Upon converting it to a string, it can be recognised for functions such as partitioning configurations, the 'SMOTE' function, and the respective learner and predictor nodes of most of the classification models.

G. Partitioning (KNIME)



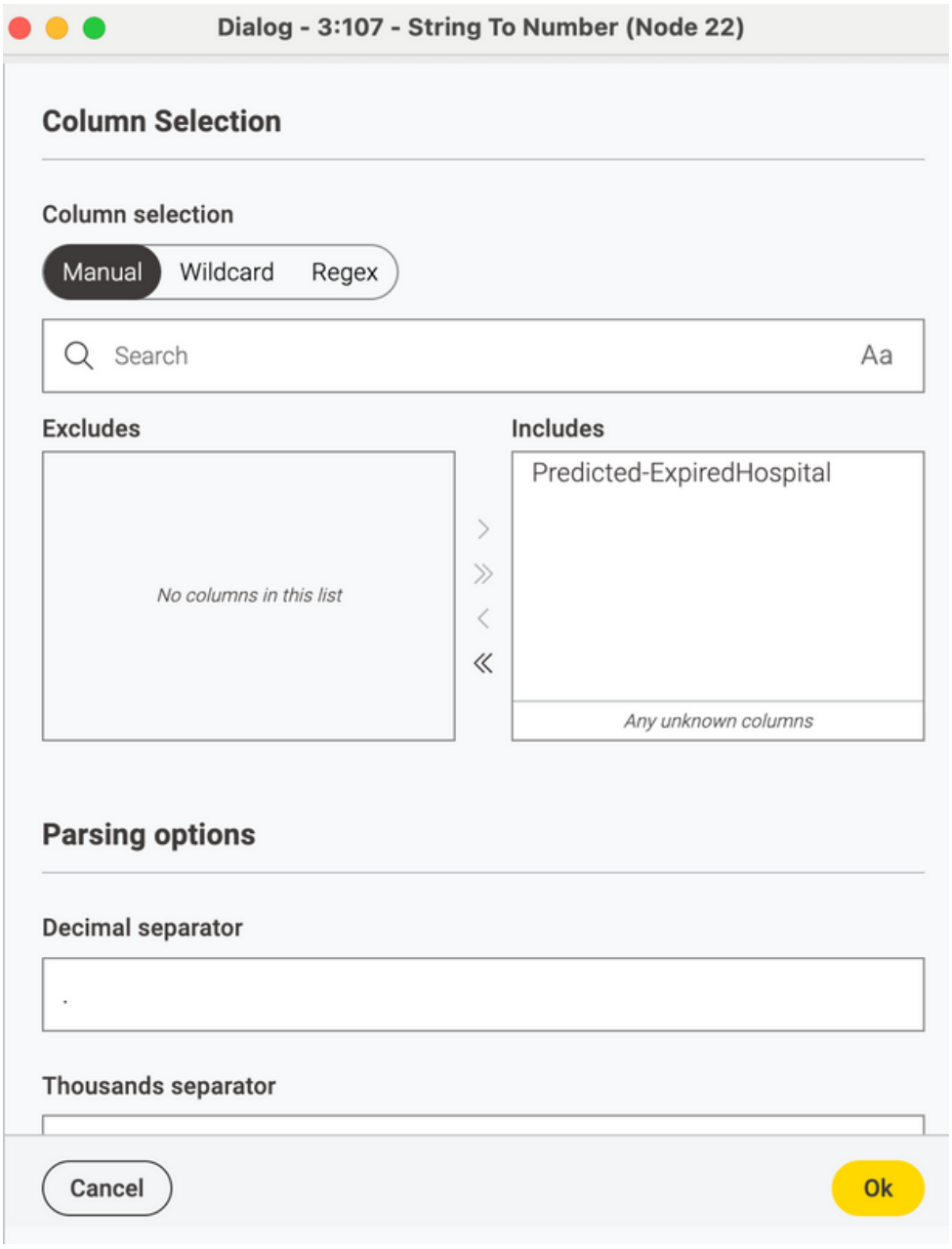
Default

SVM



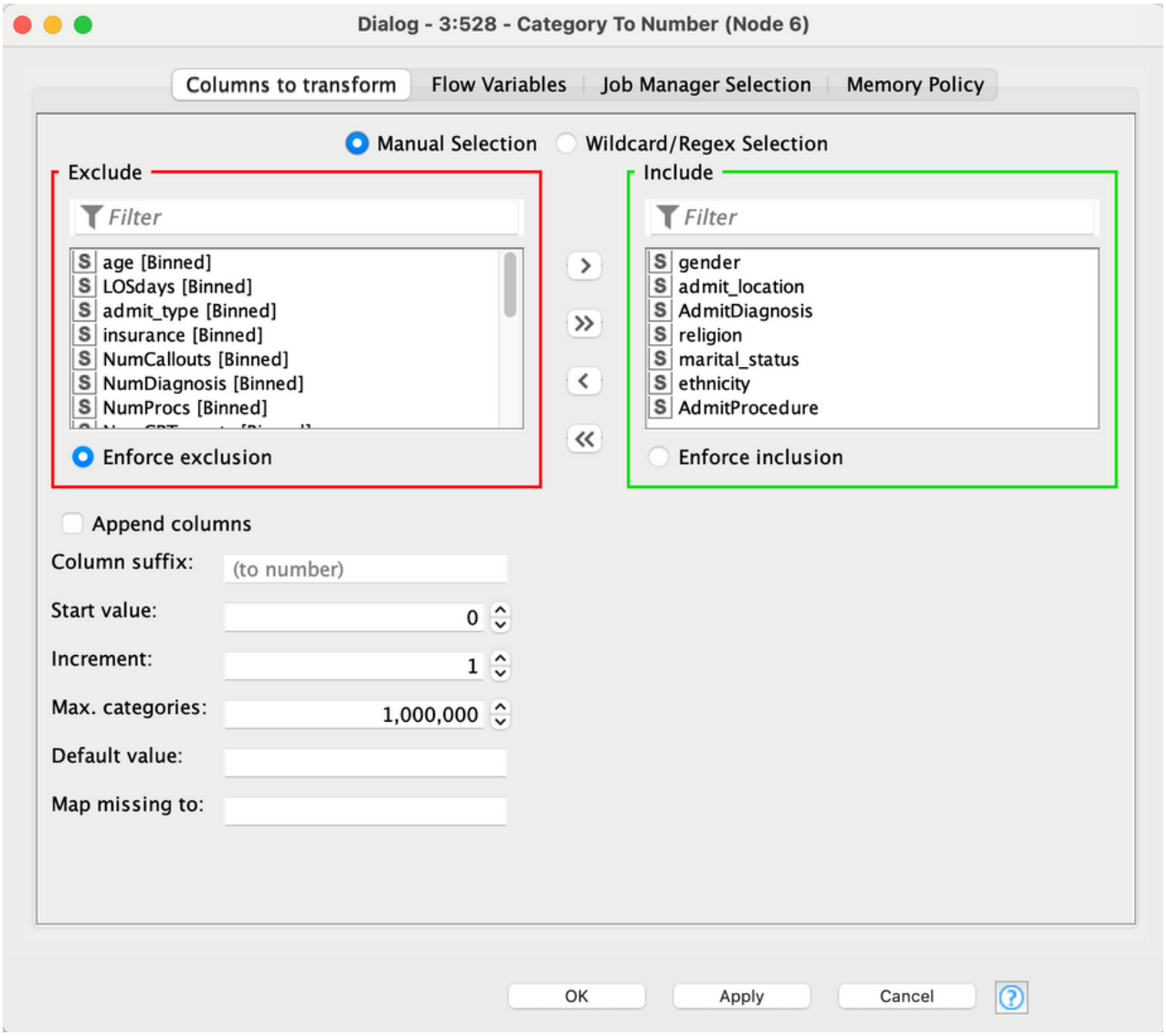
Partitioning is an integral component of data preprocessing that has been done using KNIME. It allows the user to specify the percentage of data for training and testing, where the configuration has been set to 90% training and 10% testing. The method of drawing data samples for training and testing ensures that bias is eliminated and the data can be reproduced in machine learning experiments. It may also assist to handle imbalanced datasets, particularly for the prescribed dataset in assignment 3 where there are significantly more 0s than 1s for 'ExpiredHospital'. However, for SVM, it has been altered to the default settings of 70% training and 30% testing. The random seed has also been selected to ensure reproducibility which allows for same and consistent results, while it is also useful for debugging and testing purposes.

H. String-To-Number (KNIME)



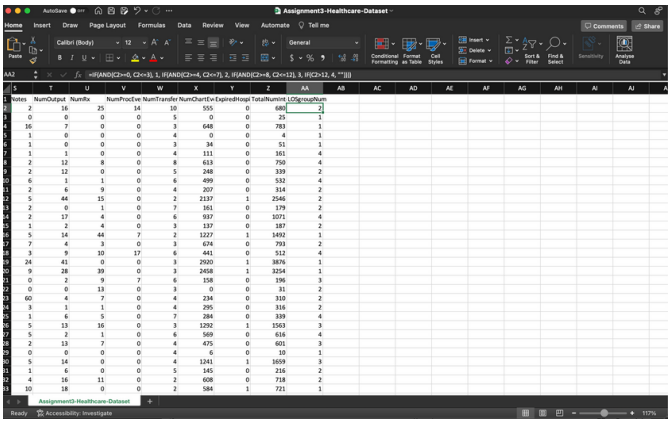
After the model has made its predictions with the unknown dataset, the numbers will be of the string data type. However, the csv writer that assists you to generate the predictions in a Microsoft Excel file requires that the data type is numerical. The 'String-To-Number' node is essential to ensure the convention, and will allow the prediction file to be generated efficiently without error.

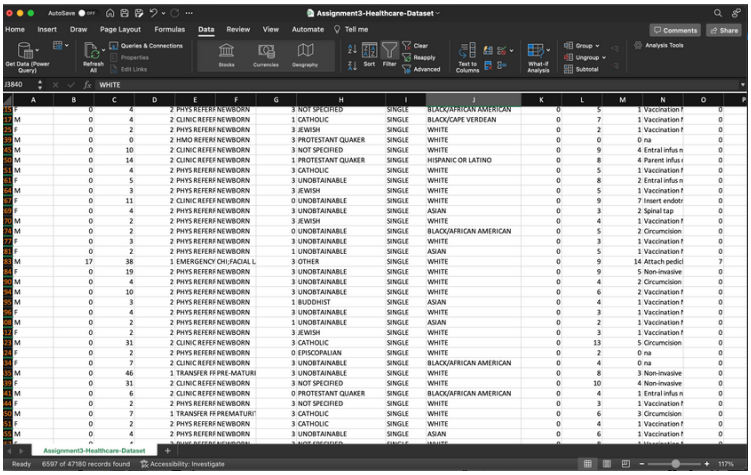
I. Category-To-Number (KNIME)

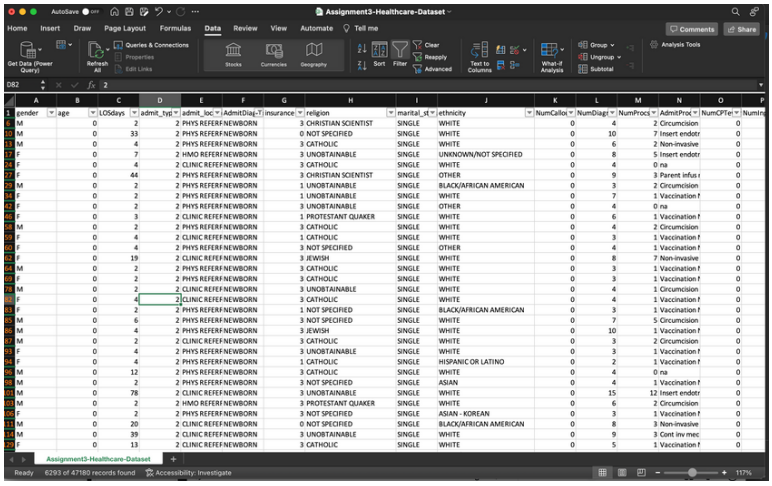


The 'Category-To-Number' node has been implemented where possible to do label encoding on the categorical attributes. This is the only encoding method that worked well in KNIME and was used for a lot of the models except for ones such as SVM, PNN and MLP. It is due to the long computation times for the learner to understand the patterns in the data, and a key essential factor to a model's success is how fast it can output valid results. The 'append columns' function has not been checked so that the transformed values are directly visible in their original locations, not being put right beside it.

J. Microsoft Excel Functions



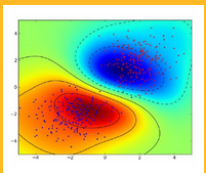




Some of the other data preprocessing functions has been done in Excel. The screenshots here illustrate the formula used to assign appropriate 'LOSgroupNums' to the missing rows, the filters used for locating those patients who are 0 years old and should therefore be labelled as newborns, and those patients who are under 18 years old that should be classified as 'single' without any blanks or 'married'. Tasks like these are simpler and more efficient to be done in Microsoft Excel prior to using KNIME. The decision has ultimately been made to only do the preprocessing that was not possible in Excel in KNIME.

04 - Classification Techniques Used (1)

The first three techniques and models that are typical ones used in data analytics to solve problems are listed below. The comparison of the traits, advantages and disadvantages presents an overview of some background information regarding the methods used.



SVM

- Primarily used for binary classifications.
- Sensitive to input feature scaling; may require data preprocessing with standardisation or normalisation.
- Results in sparse models as only a subset of training data influences decision boundary.

Advantages

- Known for strong geometric interpretation.
- Performs well in high-dimensional spaces; suitable for text classification and image recognition.
- Less prone to data overfitting.

Disadvantages

- Not efficient with large datasets, sensitive to outliers, and may be time consuming.
- Choosing kernels is difficult.
- Parameter tuning is a complex process.

Decision Tree

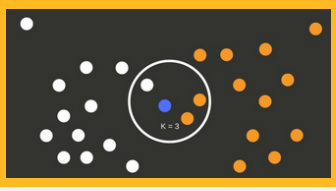
- Are binary trees by default, which can be altered using multiway splits.
- Uses metrics like entropy for measuring the purity of subsets.
- Made on a 'flowchart-like' structure with nodes and branches following classification rules.

Advantages

- Simple to understand and interpret.
- Suitable for both numerical and categorical data as they make no assumptions.
- Making predictions is fast once it has been built.

Disadvantages

- Prone to overfitting; techniques like pruning may be used to mitigate impacts.
- May be too complex with many features.
- Not ideal for continuous data.



K-Nearest Neighbour

- Are binary trees by default, which can be altered using multiway splits.
- Uses metrics like entropy for measuring the purity of subsets.
- Made on a 'flowchart-like' structure with nodes and branches following classification rules.

Advantages

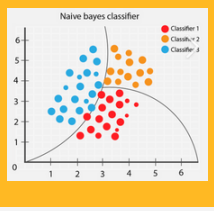
- Easy to understand and interpret.
- No training of data needed.
- Robust to noisy data and outliers.

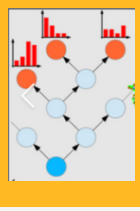
Disadvantages

- Can be computationally expensive.
- Uses a lot of memory with its training dataset.
- Sensitive to feature scaling.

04 - Classification Techniques Used (2)

The next three techniques and models that are typical ones used in data analytics to solve problems are listed below. The comparison of the traits, advantages and disadvantages presents an overview of some background information regarding the methods used.

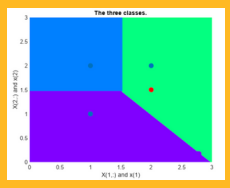


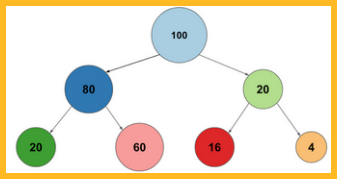


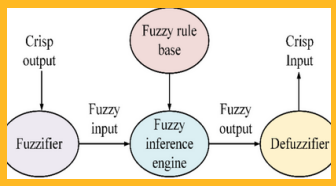
Naive Bayes	Tree Ensemble	Random Forest
• Assigns a probability to each class for a given input then selects the one with the highest. • Used for text classification. • Assumes that features do not affect the presence or absence of another feature; they are conditionally independent.	• Randomization is a key component that makes up the tree ensemble model. • Provides feature importance scores that allow one to see the factors which influences decisions. • Are an example of ensemble learning.	• Introduces random feature subsampling which is considered. • Combines results using majority voting for classification tasks. • Follows the 'Bagging' or 'Bootstrap Aggregating' approach.
Advantages: • Effective with high dimensional feature vectors like the ones in textual data. • Known for its simplicity and speed, even on large datasets. • Can effectively handle missing data as they are computed from the existing values.	**Advantages:** • Robust in handling noisy and high dimensional data. • Are capable to capture complex and non-linear relationships. • Can handle missing values without extensive preprocessing.	**Advantages:** • Uses Decision Trees (DT) that are unpruned and deep that can capture complex data patterns. • Relatively robust to noisy data. • Can handle imbalanced data well and has a high predictive accuracy.
Disadvantages • Not suitable for tasks that have complex relationships or strong feature dependencies. • Sensitive to feature scaling. • Requires enough data for the model to perform adequately.	**Disadvantages** • Gets complex with many trees that are combined. • Can still overfit, even though they may be less prone. • Tuning the hyperparameters can be a lengthy process.	**Disadvantages** • Can be challenging to interpret. • Has difficulty handling structured data with complex relationships. • Sensitive to the order of data.

04 - Classification Techniques Used (3)

The next three techniques and models that are typical ones used in data analytics to solve problems are listed below. The comparison of the traits, advantages and disadvantages, presents an overview of some background information regarding the methods used.



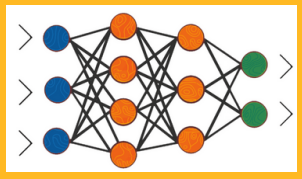




PNN	Gradient Boosted Trees	Fuzzy Rule
• Used for pattern recognition and classification tasks. • A type of Radial Basis Function Network (RBFN). • The hidden layer models probability density functions.	• An ensemble learning technique that incorporates multiple decision trees. • Various techniques can be implemented to address class imbalance in the training data. • Inputs missing values based on the available data.	• Uses linguistic rules to model and make decisions. • Works together with fuzzy sets to handle information. • Designed with human understandable rules.
Advantages: • Known for their fast training process. • Can efficiently 'memorise' the training data. • Offers an alternative to traditional neural network architectures.	**Advantages:** • Relatively robust to noisy data and outliers. • Known for its high predictive accuracy. • The weak-to-strong approach with the trees built result in a strong model.	**Advantages:** • Fuzzy logic can handle uncertain information. • It has great flexibility as it has a wide range of applications. • Can be scaled for larger and more complex problems.
Disadvantages • May lack interpretability due to their strong focus on pattern recognition and probability estimation. • Sensitive to prototype positions. • Number of prototype vectors can affect the model complexity and performance.	**Disadvantages** • Not suitable for structured data with complex feature relationships. • Can be computationally and memory intensive. • Proper hyperparameter tuning is essential.	**Disadvantages** • Need to be fine-tuned for specific datasets. • Limited to specific problem types. • Can be computationally intensive.

04 - Classification Techniques Used (4)

The last model that is used in data analytics to solve problems is MLP, which has been listed below. The comparison of the traits, advantages and disadvantages, presents an overview of some background information regarding the model and its functions.



MultiLayer Perceptron

- Has a feedforward architecture that does not contain cycles.
- Consists of artificial neurons in its layers.
- Has more than one hidden layer between the input and output layers.

Advantages:

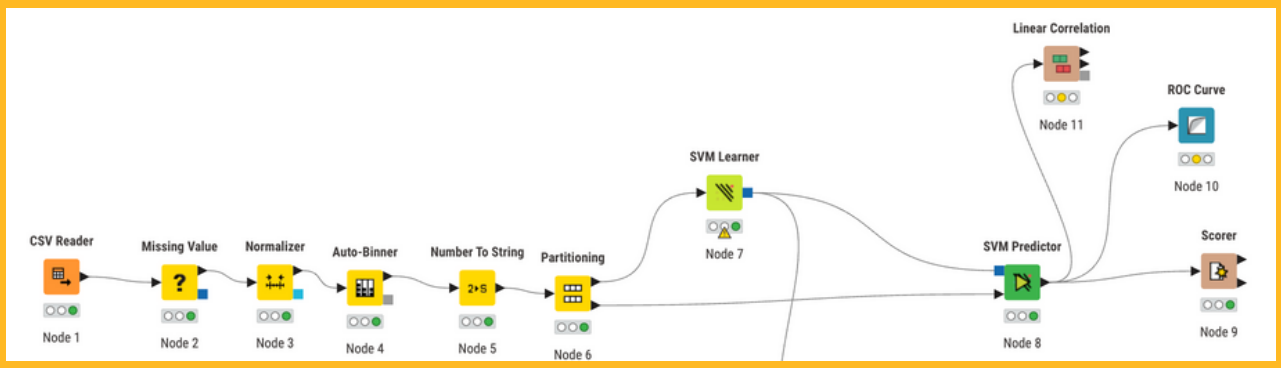
- Capable of approximating a wide range of functions.
- Can learn hierarchical representations and features from raw data.
- Able to handle large and high dimensional datasets.

Disadvantages

- Deep MLP models can get very complex with many parameters.
- Overfitting is a common issue.
- Using large datasets and complex methods can increase training times.

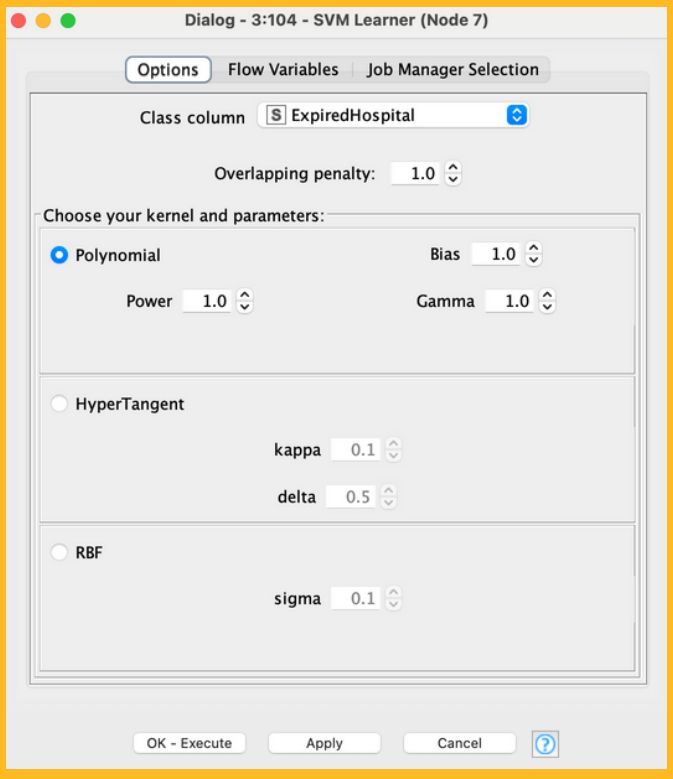
04 - Classification Techniques Used (5)

Support Vector Machines (SVM)



Using the Support Vector Machine (SVM) model, it has been decided to make a conversion from the data with a nonlinear mapping function into a feature space. The nonlinear mapping function typically involves polynomial functions with parameters such as power, gamma and bias. The data was classified effectively without the SMOTE function and preprocessing that does not encode any values, and the partition has also been reduced to the default of 70% training and 30% testing. This enables a more efficient and faster result from the machine learning model, which will otherwise be very lengthy.

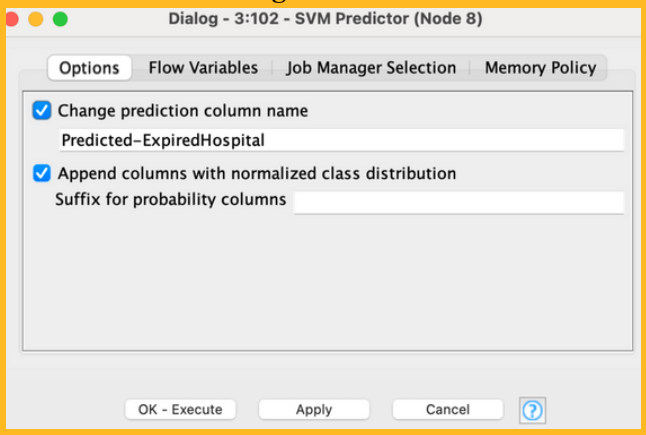
SVM Learner Configuration



This is the configuration for the SVM Learner. The target class column has been set to the target attribute of 'ExpiredHospital', with the overlapping penalty remaining as the default value of 1. The kernel and parameters have been chosen to use the 'polynomial' function, whereby the bias, gamma and power are the default values of 1. Since this is not a very complex configuration, it can contribute to results being outputted faster and more reliably as well.

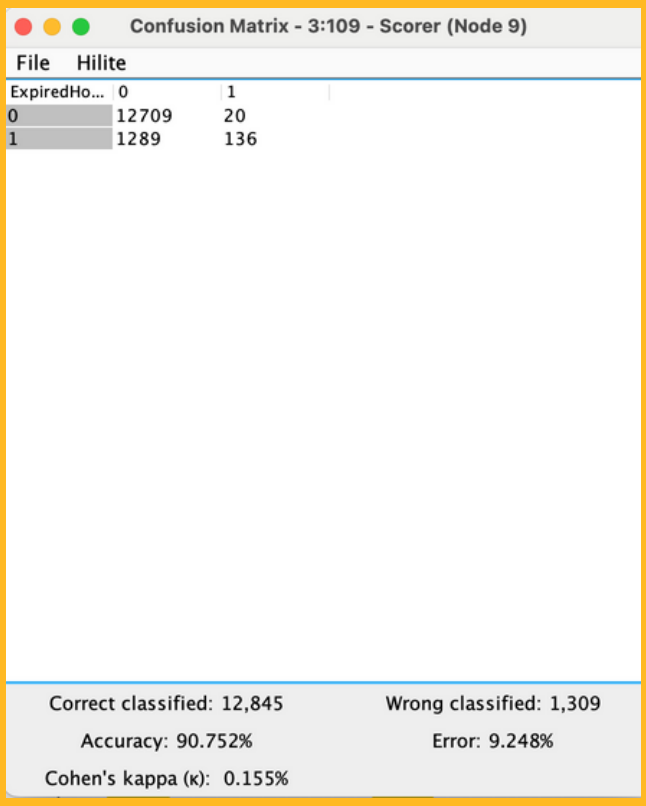
04 - Classification Techniques Used (6)

SVM Predictor Configuration



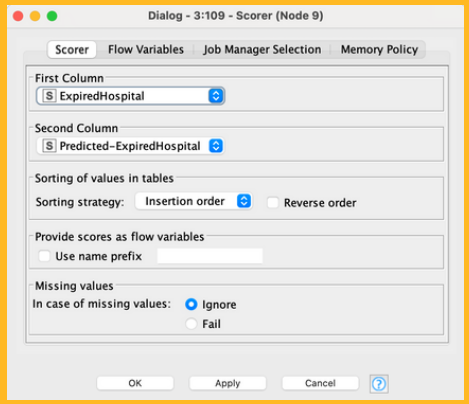
This is the configuration for the SVM Predictor node. The prediction column has been altered to 'Predicted-ExpiredHospital' as per Kaggle submission requirements and for it to match that of the unknown dataset. It has also been set to always append columns with the normalized class distribution as the data mining problem deals with an imbalanced dataset and it may help to improve the model performance.

SVM Scorer



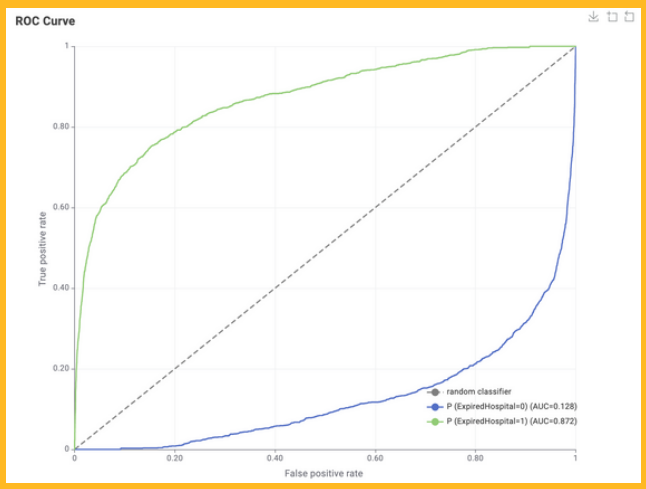
As one can see here, the model has predicted 12709 true negatives, 20 false positives, 1289 false negatives and 136 true positives. The positive class has been set as 1, while the negative class is 0.

For the SVM scorer, the first column was set as 'ExpiredHospital' and the second column was set to 'Predicted-ExpiredHospital'. This will form default settings that are the same throughout my models, alongside the sorting strategy of 'Insertion order'. The setting has been configured to ignore missing values, however, that does not matter too much due to how the preprocessing has already dealt with it. From here, it can also be seen that 90.752% of the train/test data has been correctly classified, and around 9% has been incorrectly classified. The Cohen's Kappa (K) is 0.155%, meaning that the agreement between two raters is slightly better than what would be expected by chance.

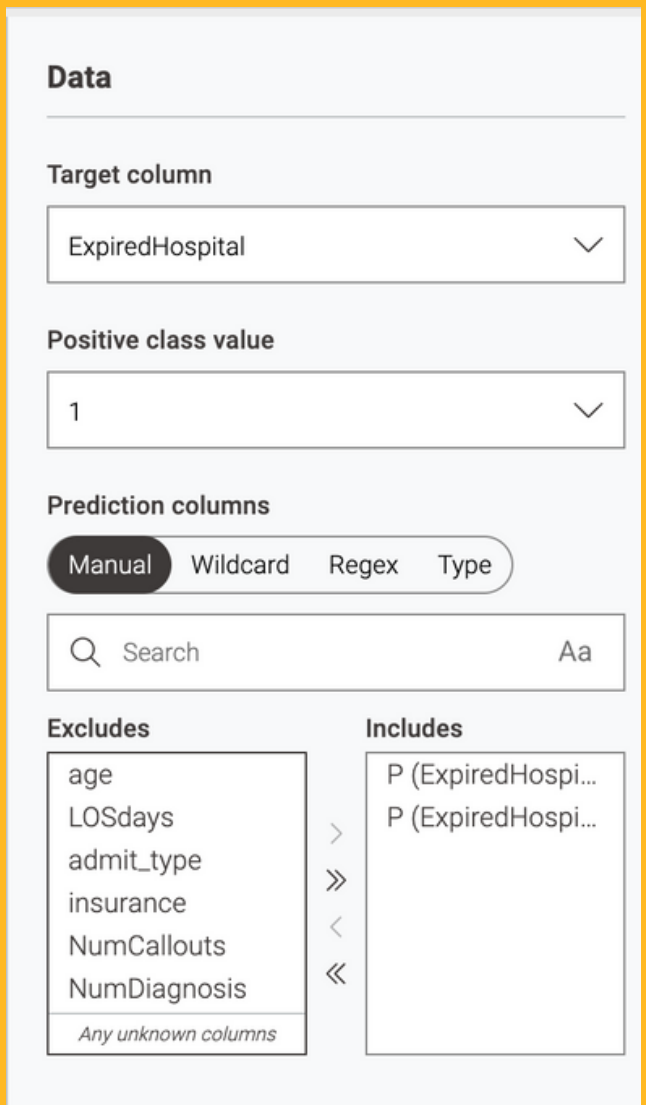


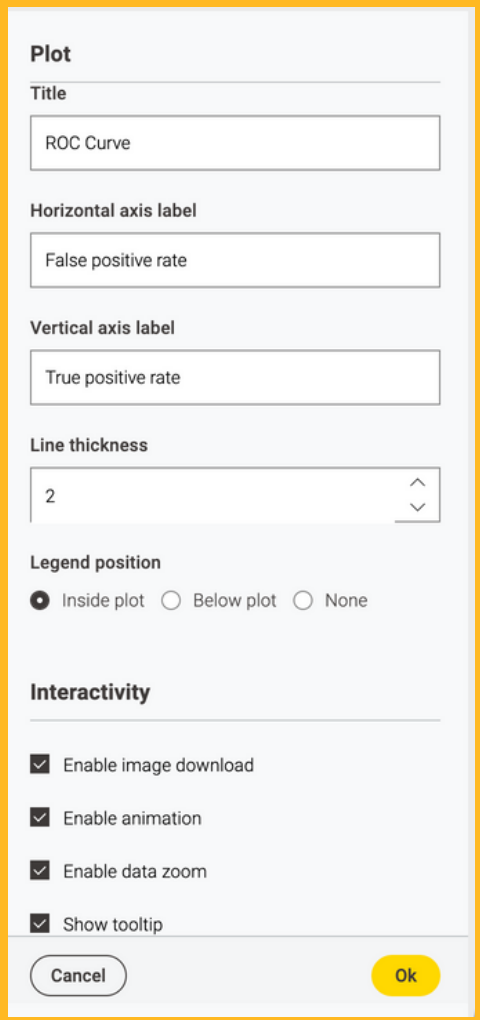
04 - Classification Techniques Used (7)

SVM AUC-ROC Curve



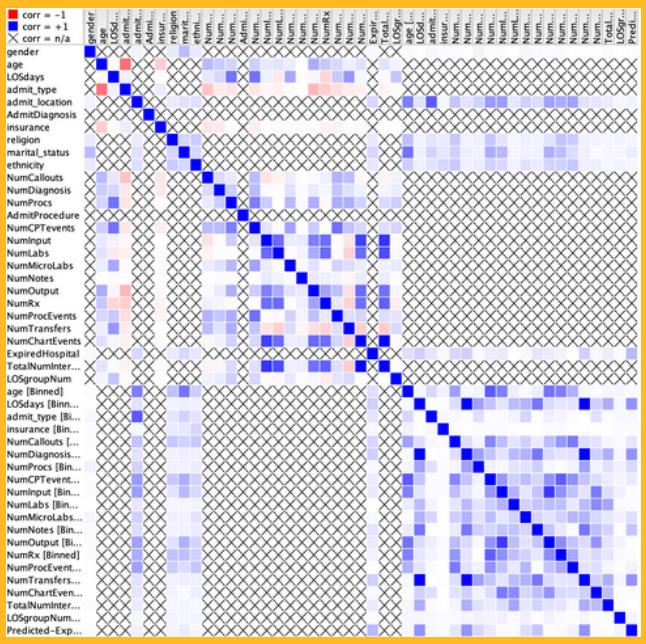
For the AUC-ROC Curve of the SVM model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. For the P (ExpiredHospital = 0), the AUC value was 0.128 and the P (ExpiredHospital = 1) was 0.872. The negative class is a mirror image of the positive. The other attributes have been excluded as the primary focus is the probabilities for each class value of 'ExpiredHospital'.





04 - Classification Techniques Used (8)

SVM Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that most of the attribute pairs have positive correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is between 'NumRx' and 'NumLabs'. However, it can also be seen that the relationships such as the one between 'age' and 'admit_type' is weak. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

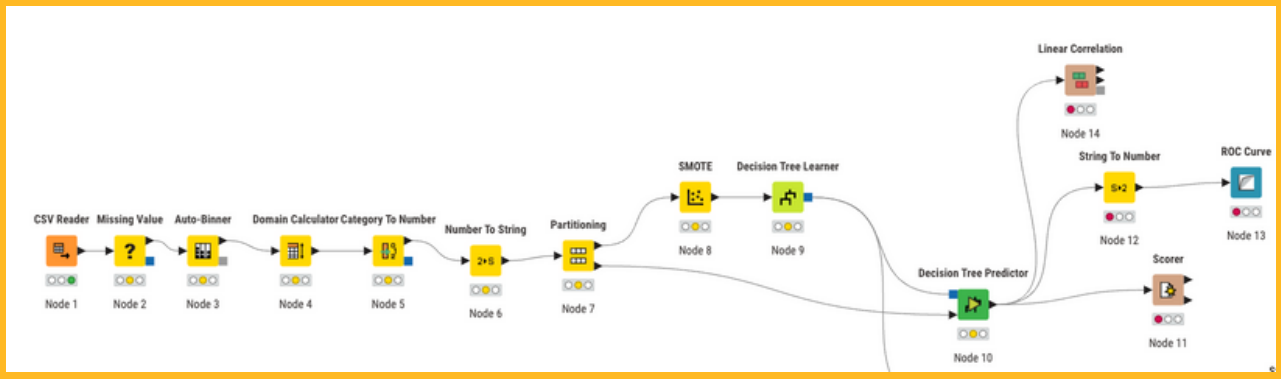
SVM Brief Statistics

Attribute Class	**Recall**	**Precision**	F1
0	0.998	0.908	0.951
1	0.095	0.872	0.172

These are some basic statistics for the Support Vector Machine (SVM) model. In terms of the 'Precision' as a key metric, the model has a precision of 0.908 for the '0' class and a precision of 0.872 for the '1' class. These are both high values that means that when the model predicts a positive instance, it has a high chance of being accurate. The recall values are 0.998 for the '0' class and 0.095 for the '1' class, meaning that while the model is extremely great at capturing most of the positive instances for the '0' class, it struggles to capture them for the '1' class. The 'F-Measure' or F1 scores are 0.951 for the '0' class and 0.172 for the '1' class, highlighting that the model is more capable of predicting actual instances of '0' than '1'.

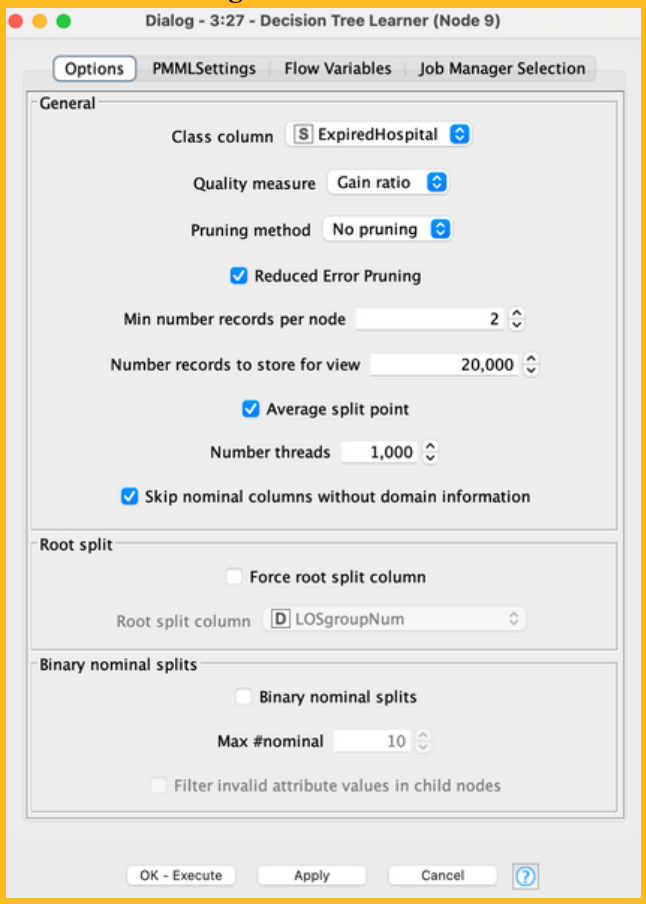
04 - Classification Techniques Used (9)

Decision Trees (DT)



For the Decision Tree model (DT), the data will be transformed into a series of structural trees. It is relatively robust to noisy data, and the model can work well with effective hyperparameter tuning. The SMOTE node has been used to balance the classes a little for the model to be more robust while enhancing its performance, and this did not increase the computation time during testing. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions.

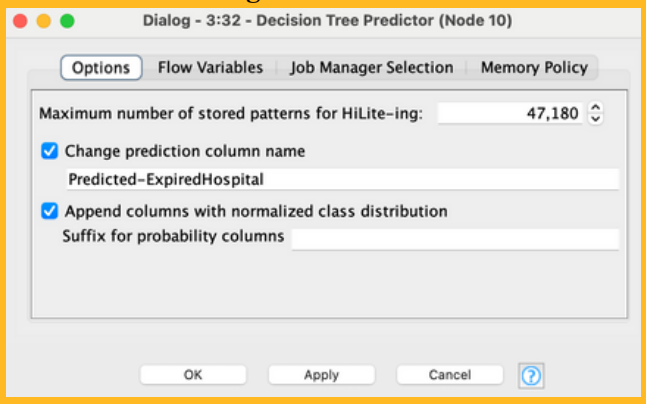
DT Learner Configuration



This is the configuration for the DT Learner. The target class column has been set to the target attribute of 'ExpiredHospital', with the quality measure being set to the 'Gain ratio' as default. The minimum number of records per node and the number of records for view are just to strengthen data performance. The number of threads has been set to 1000 for the model to potentially accelerate the tree building process especially when dealing with large datasets.

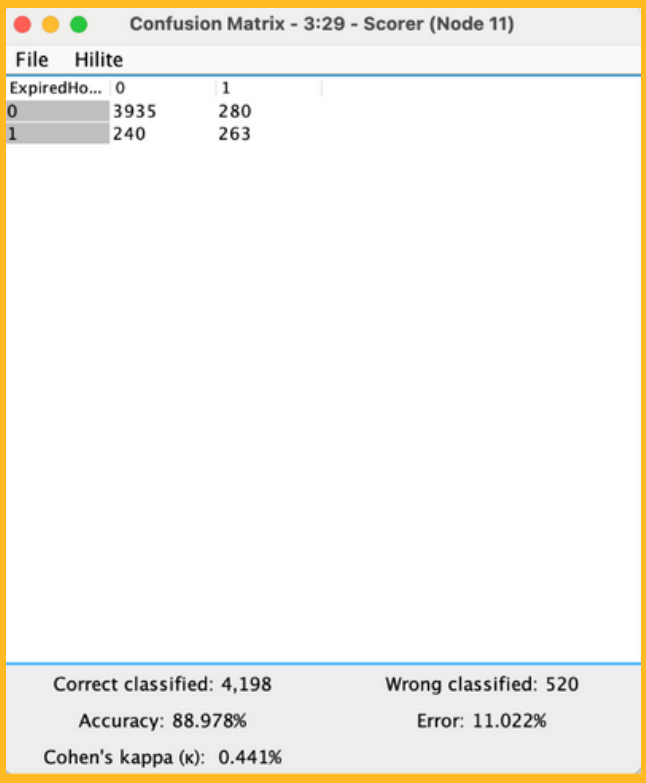
04 - Classification Techniques Used (10)

DT Predictor Configuration



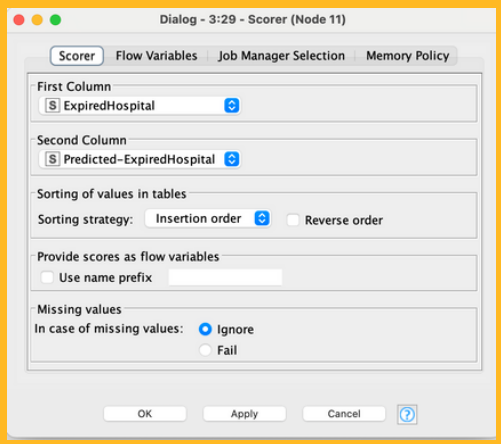
This is the configuration for the DT Predictor node. The prediction column has been altered to 'Predicted-ExpiredHospital' as per Kaggle submission requirements and for it to match that of the unknown dataset. The number chosen for the 'stored patterns for HiLite-ing ensures that all of the rows are considered. It has also been set to append columns with the normalized class distribution as the data mining problem deals with an imbalanced dataset and it may help to improve the model performance.

DT Scorer



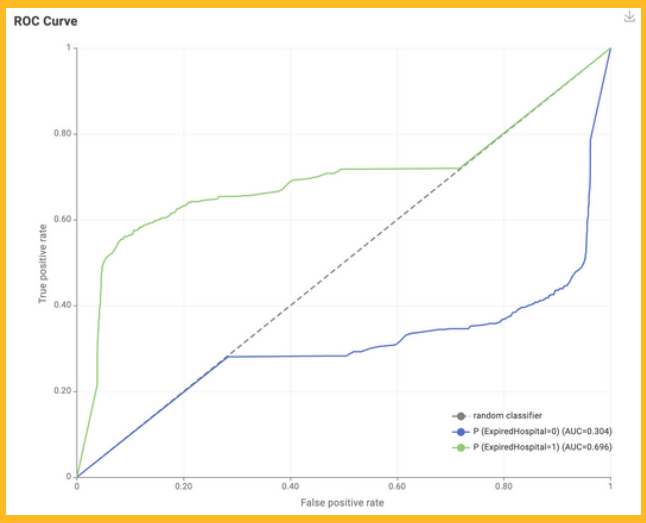
As one can see here, the model has predicted 3935 true negatives, 280 false positives, 240 false negatives and 263 true positives. The positive class has been set as 1, while the negative class is 0.

For the DT scorer, the first column was set as 'ExpiredHospital' and the second column was set to 'Predicted-ExpiredHospital'. This will form default settings that are the same throughout my models, alongside the sorting strategy of 'Insertion order'. The setting has been configured to ignore missing values, however, that does not matter too much due to how the preprocessing has already dealt with it. From here, it can also be seen that 88.978% of the train/test data has been correctly classified, and around 11% has been incorrectly classified. The Cohen's Kappa (K) is 0.441%, meaning that the agreement between two raters is moderate albeit not being the best.

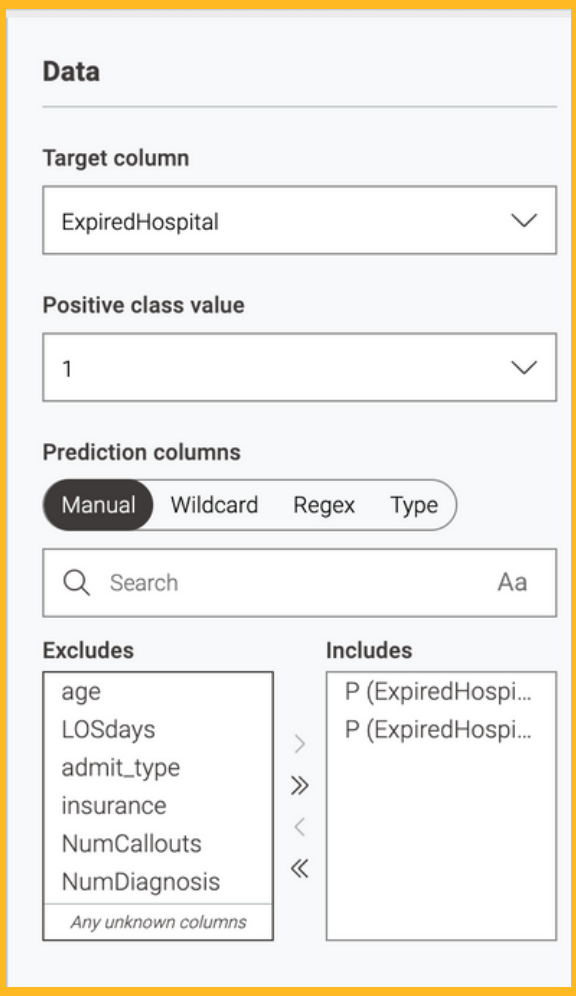


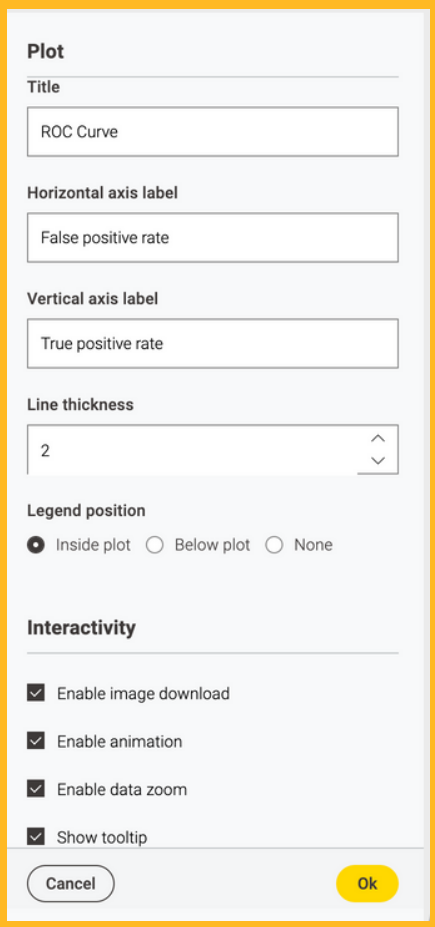
04 - Classification Techniques Used (11)

DT AUC-ROC Curve



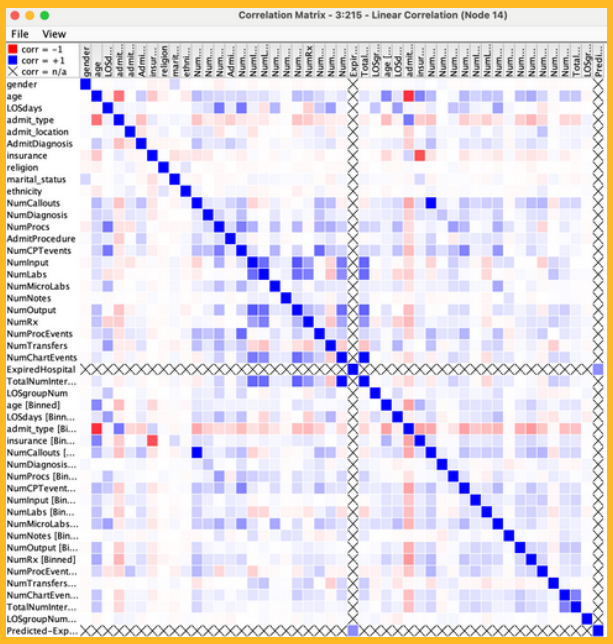
For the AUC-ROC Curve of the DT model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. For the P (ExpiredHospital = 0), the AUC value was 0.304 and the P (ExpiredHospital = 1) was 0.696. The probability of 'ExpiredHospital = 0' is the mirror image of that of 'ExpiredHospital = 1'. The other attributes have been excluded as the primary focus is the probabilities for each class value of 'ExpiredHospital'. It can also be seen that for both positive and negative classes, there's a part that illustrate characteristics of a random classifier.





04 - Classification Techniques Used (12)

DT Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'NumCPTevents' and 'LOSdays'. However, it can also be seen that relationships such as the one between 'age' and 'insurance' is weak. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

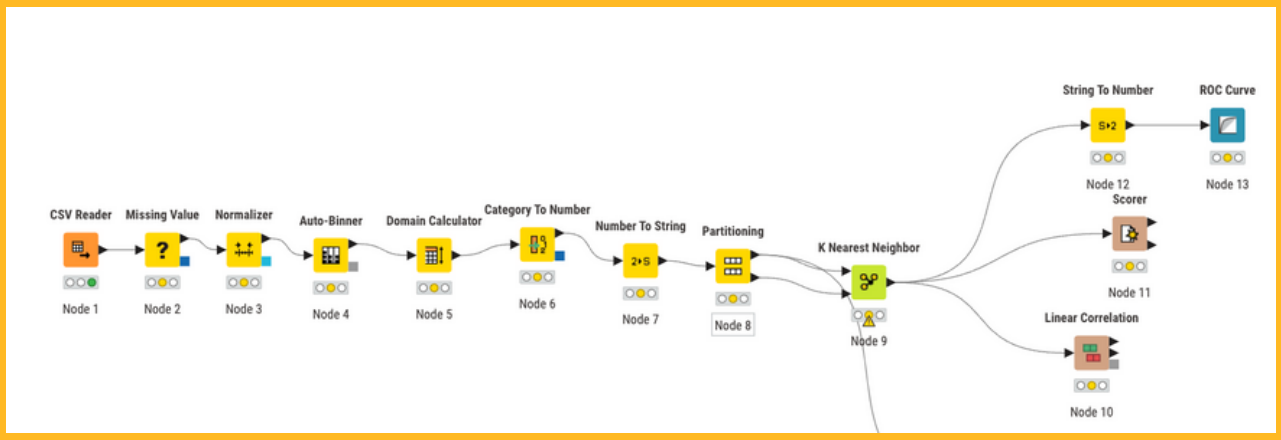
DT Brief Statistics

Attribute Class	**Recall**	**Precision**	F1
0	0.934	0.943	0.938
1	0.523	0.484	0.503

These are some basic statistics for the Decision Tree (DT) model. In terms of the 'Precision' as a key metric, the model has a precision of 0.943 for the '0' class and a precision of 0.484 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is close to random guessing. The recall values are 0.934 for the '0' class and 0.523 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it is equivalent to random guessing for the '1' class. The 'F-Measure' or F1 scores are 0.938 for the '0' class and 0.503 for the '1' class, highlighting that the model is more capable of predicting actual instances of '0' rather than instances of '1'.

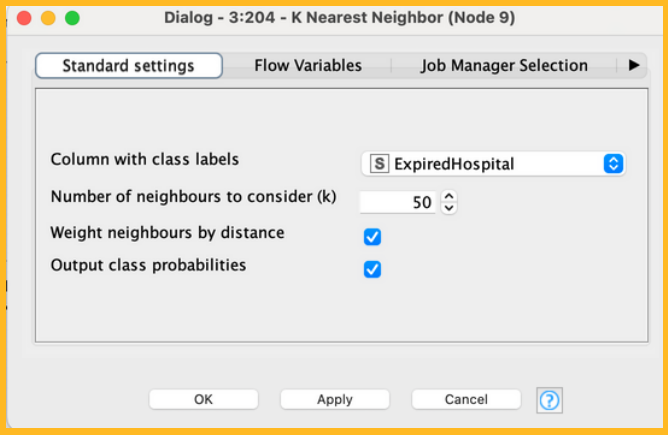
04 - Classification Techniques Used (13)

K-Nearest Neighbour (KNN)



For the K-Nearest Neighbour (KNN) model, the SMOTE node wasn't used in the same way as tree based algorithms such as Decision Trees (DT). The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The KNN node acts as a learner and predictor in one node, partitioning into the non-empty subset where seed points are chosen as centroids.

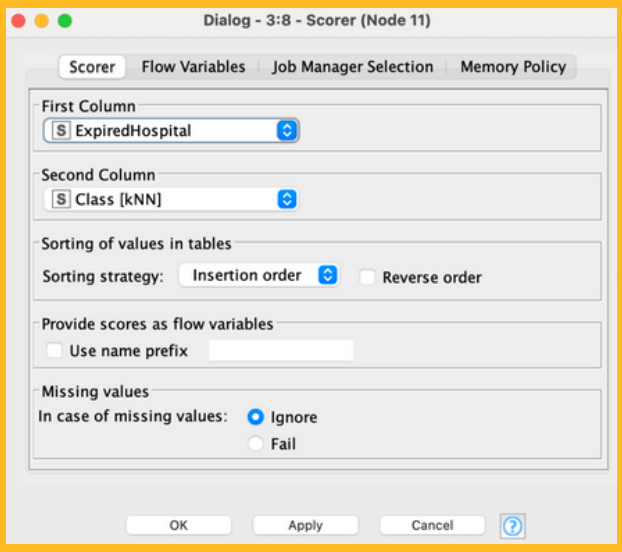
KNN Configuration



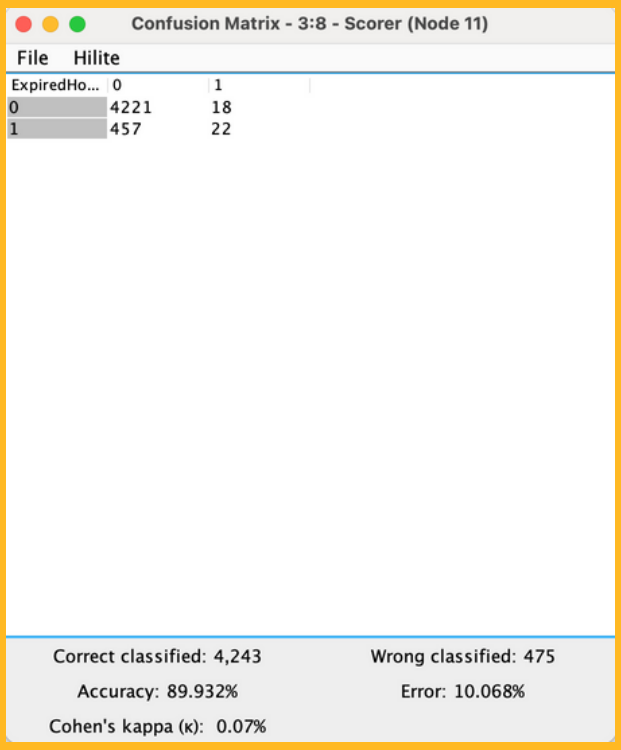
This is the configuration for the main KNN node. The target class column has been set to the target attribute of 'ExpiredHospital', with the number of neighbours to consider being the default value of 50. To ensure that the thorough analysis is done, class probabilities are outputted for the user to view. Also, the option for weighing neighbours by distance is checked so that the ones that are the closest to 'ExpiredHospital' would form values that are the most indicative.

04 - Classification Techniques Used (14)

KNN Scorer



This is the configuration for the scorer node in terms of the KNN model. The first column has been set as 'ExpiredHospital' while the second has been set as 'Class (kNN)'. The second column is where the KNN model will make its predictions. The default sorting strategy is implemented as 'insertion order'. The check for missing values has been set to 'ignore', which does not matter too much as they have already been handled in the preliminary preprocessing.

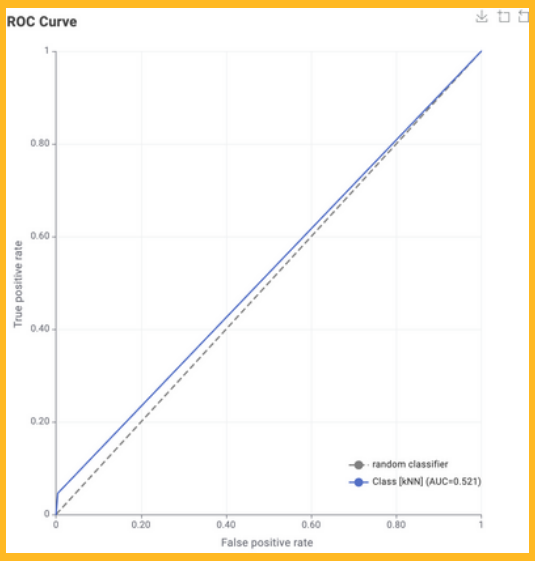


From the output of the scorer node, it can be seen that 89.932% of the train/test data has been correctly classified, and around 10% has been incorrectly classified. The Cohen's Kappa (K) is 0.07%, meaning that the agreement between two raters is about as equivalent as that by chance.

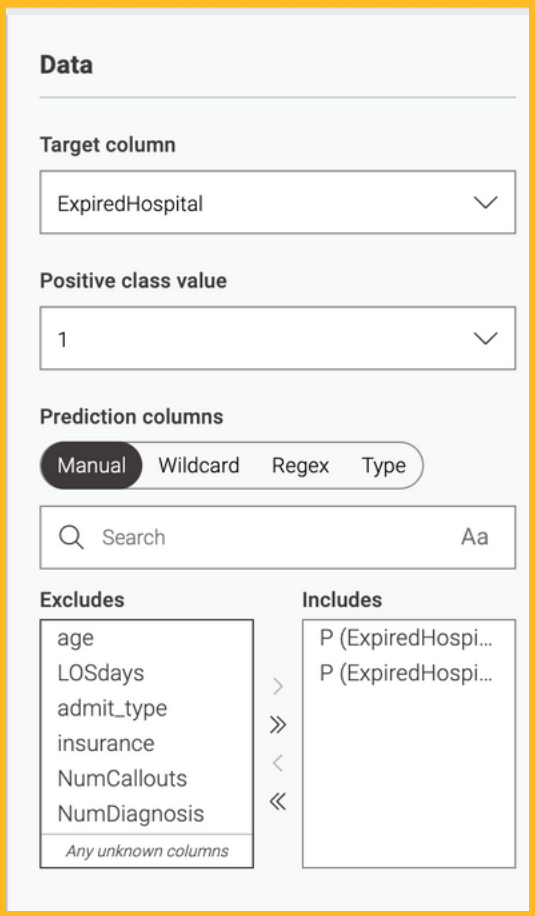
As one can see here, the model has predicted 4221 true negatives, 18 false positives, 457 false negatives and 22 true positives. The positive class has been set as 1, while the negative class is 0.

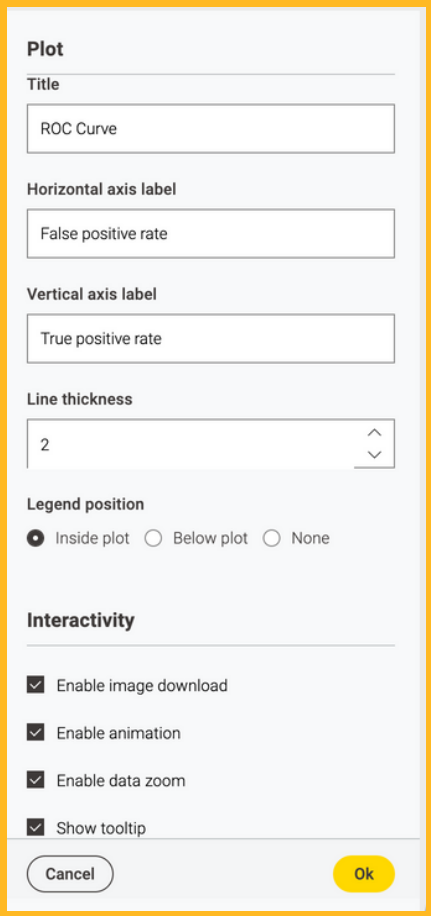
04 - Classification Techniques Used (15)

KNN AUC-ROC Curve



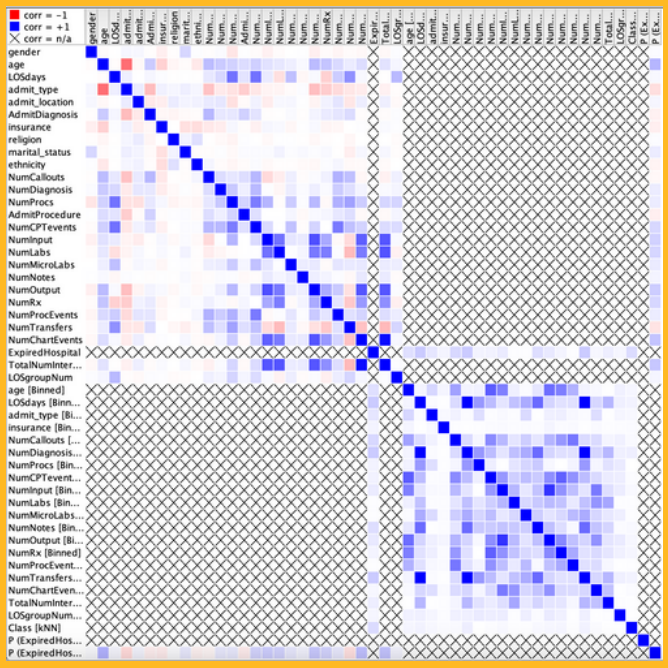
For the AUC-ROC Curve of the KNN model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of KNN yielded an AUC score of 0.521, meaning that it is marginally better than random guessing. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital' (is 'Class(kNN)' in this instance). As the false and true positive rates increase, characteristics of a random classifier becomes apparent as the curve moves closer to the dotted line.





04 - Classification Techniques Used (16)

KNN Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'NumOutput' and 'TotalNumInteract'. However, it can also be seen that relationships such as the one between 'NumTransfers' and 'TotalNumInteract' is weak. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

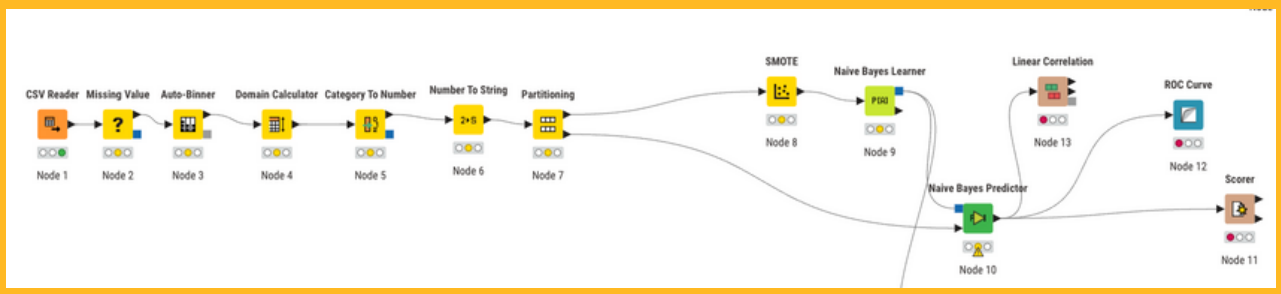
KNN Brief Statistics

Attribute Class	Recall	Precision	F1
0	0.996	0.902	0.947
1	0.046	0.55	0.085

These are some basic statistics for the K-Nearest Neighbours (KNN) model. In terms of the 'Precision' as a key metric, the model has a precision of 0.902 for the '0' class and a precision of 0.55 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is close to random guessing. The recall values are 0.996 for the '0' class and 0.046 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it performs poorly for the '1' class. The 'F-Measure' or F1 scores are 0.947 for the '0' class and 0.085 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' rather than instances of '1'.

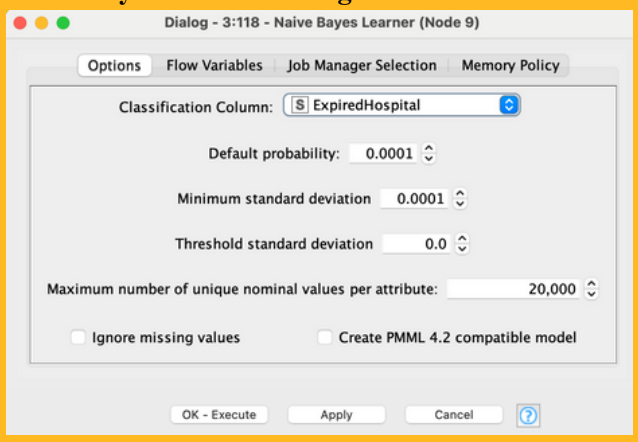
04 - Classification Techniques Used (17)

Naive Bayes



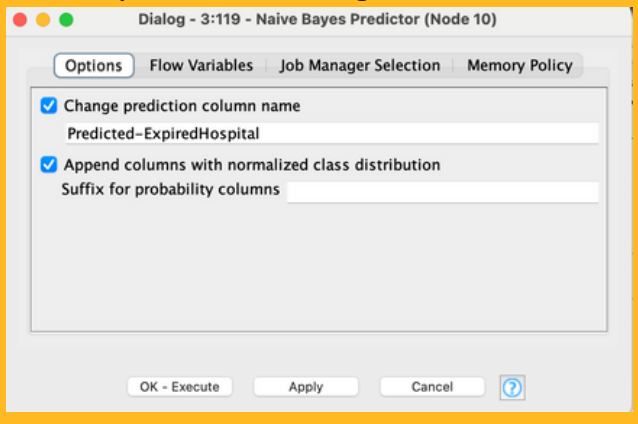
For the Naive Bayes model, the SMOTE node was used to handle class imbalance within the data. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The Naive Bayes Learner node acts as a learner that identifies as many rules and patterns as possible by utilising Bayes Theorem. The Naive Bayes Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

Naive Bayes Learner Configuration



This is the configuration for the Naive Bayes Learner node. The target classification column has been set to the target attribute of 'ExpiredHospital'. The default probability, minimum standard deviation and threshold standard deviation are set as default values. However, the maximum number of unique nominal values per attribute has been set to 20000, so that more complex patterns can be learnt due to it allowing for a high number of categories to be generated.

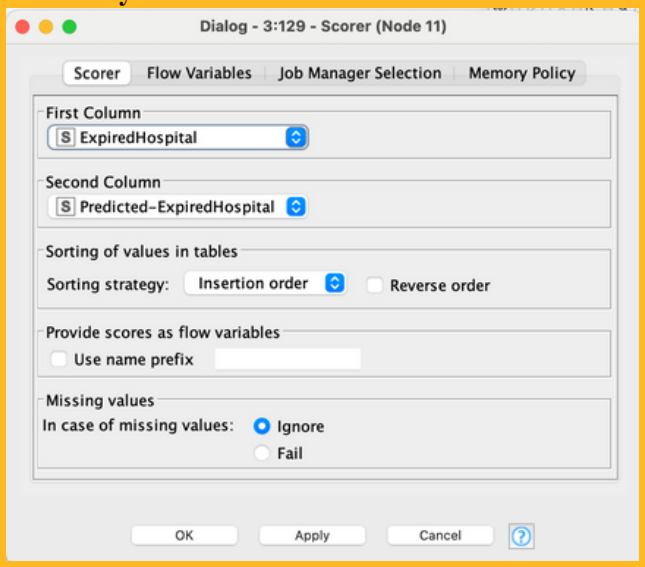
Naive Bayes Predictor Configuration



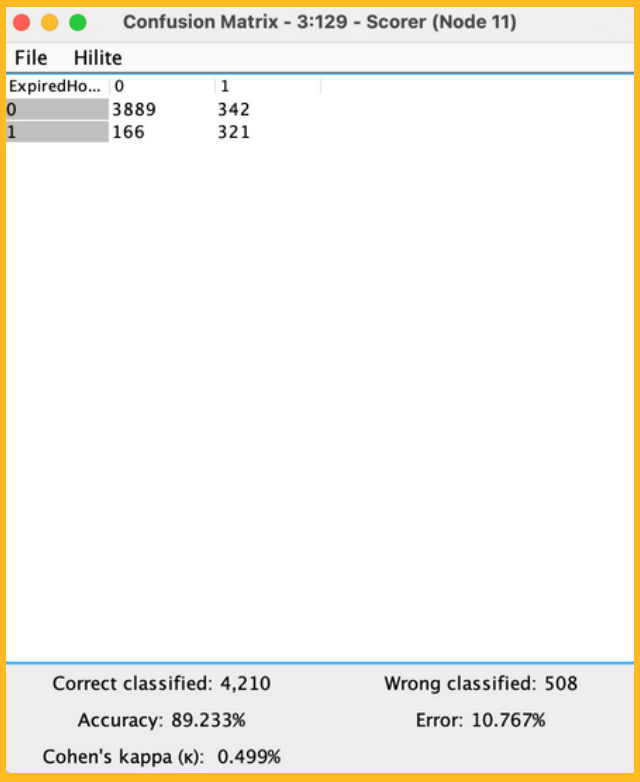
The configuration for the predictor is similar to some of the other models. The prediction column name has been altered to 'Predicted-ExpiredHospital'. It has also been set to append columns with the normalized class distribution as the data mining problem deals with an imbalanced dataset and it may help to improve the model performance.

04 - Classification Techniques Used (18)

Naive Bayes Scorer



This is the configuration for the scorer node in terms of the Naive Bayes model. The first column has been set as 'ExpiredHospital' while the second has been set as 'Predicted-ExpiredHospital'. The second column is where the model will make its predictions. The default sorting strategy is implemented as 'insertion order'. The check for missing values has been set to 'ignore', which does not matter too much as they have already been handled in the preliminary preprocessing.

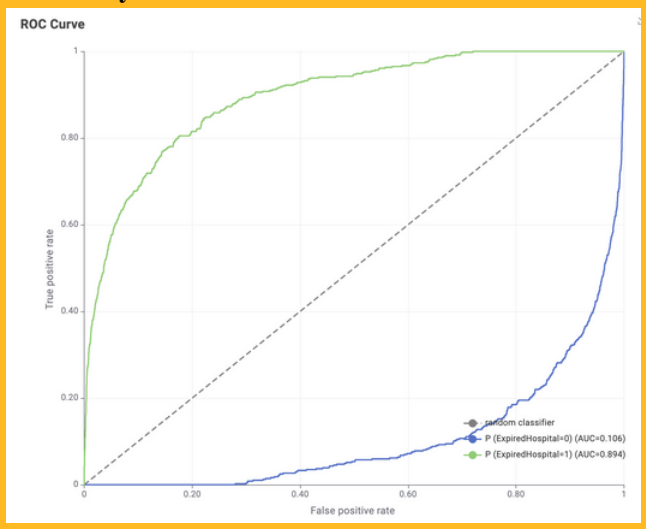


From the output of the scorer node, it can be seen that 89.233% of the train/test data has been correctly classified, and around 11% has been incorrectly classified. The Cohen's Kappa (K) is 0.499%, meaning that the agreement between two raters is better than that of chance and on the upper bounds.

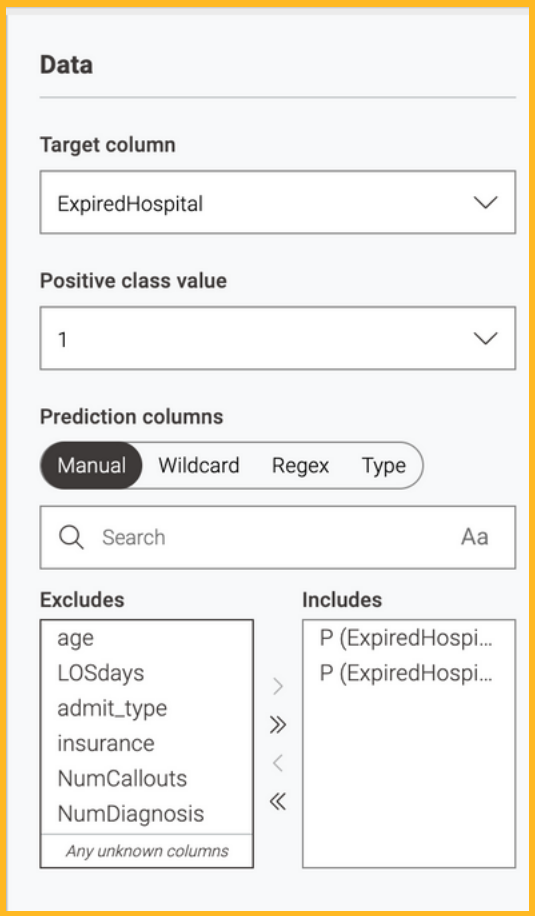
As one can see here, the model has predicted 3889 true negatives, 342 false positives, 166 false negatives and 321 true positives. The positive class has been set as 1, while the negative class is 0.

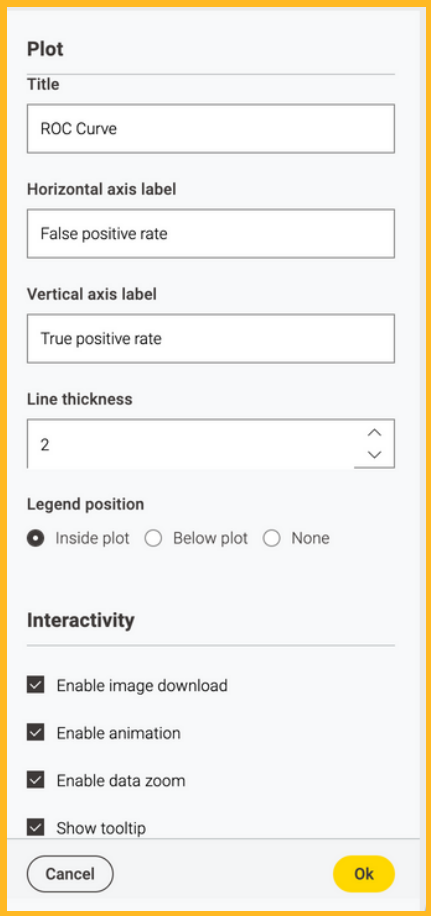
04 - Classification Techniques Used (19)

Naive Bayes AUC-ROC Curve



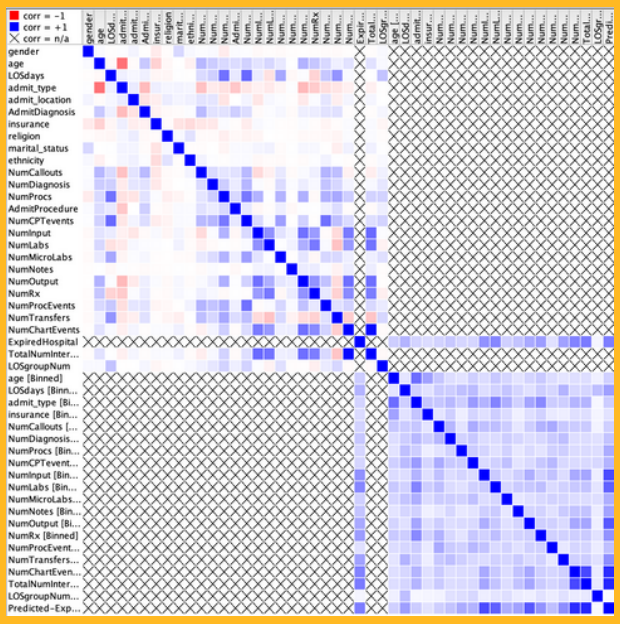
For the AUC-ROC Curve of the Naive Bayes model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for Naive Bayes yielded an AUC score of 0.894, meaning that it is way better than random guessing. The class of 0 yielded an AUC score of 0.106, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (20)

Naive Bayes Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'ExpiredHospital' and 'Predicted-ExpiredHospital'. This means that this model has some degree of validity in predicting the patients who may die while in a hospital during a real-world scenario. However, it can also be seen that relationships such as the one between 'insurance' and 'gender' is weak. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

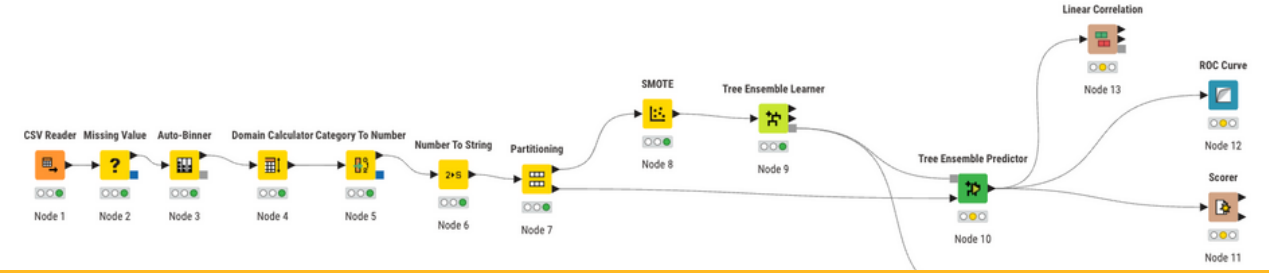
Naive Bayes Brief Statistics

Attribute Class	Recall	Precision	F1
0	0.919	0.959	0.939
1	0.659	0.484	0.558

These are some basic statistics for the Naive Bayes model. In terms of the 'Precision' as a key metric, the model has a precision of 0.959 for the '0' class and a precision of 0.484 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is close to random guessing. The recall values are 0.919 for the '0' class and 0.659 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it performs just moderately well for the '1' class. The 'F-Measure' or F1 scores are 0.939 for the '0' class and 0.558 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' rather than instances of '1' which are close to random guessing.

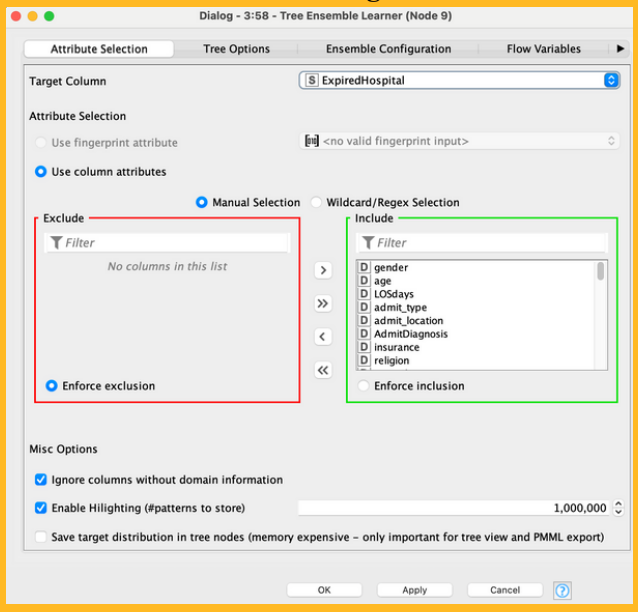
04 - Classification Techniques Used (21)

Tree Ensemble (TE)



For the Tree Ensemble model, the SMOTE node was used to handle class imbalance within the data. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The Tree Ensemble Learner node acts as a learner that identifies as many rules and patterns as possible by utilising multiple Decision Trees that are combined to give the best result possible. The Tree Ensemble Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

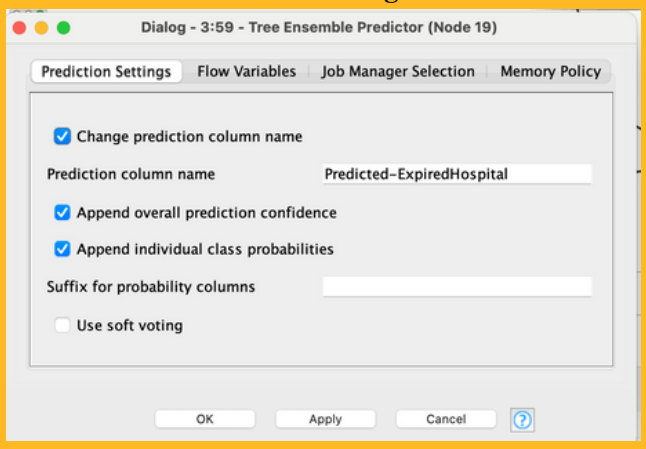
Tree Ensemble Learner Configuration



The tree ensemble learner configuration has been set to focus on the target column of 'ExpiredHospital'. It has been set to learn patterns in data from all of the attributes possible, which may also include binned columns for it to make more accurate predictions. It has been set to ignore columns without domain information as per the default and the patterns to store has been stipulated at 1000000 for it to learn as many patterns as possible.

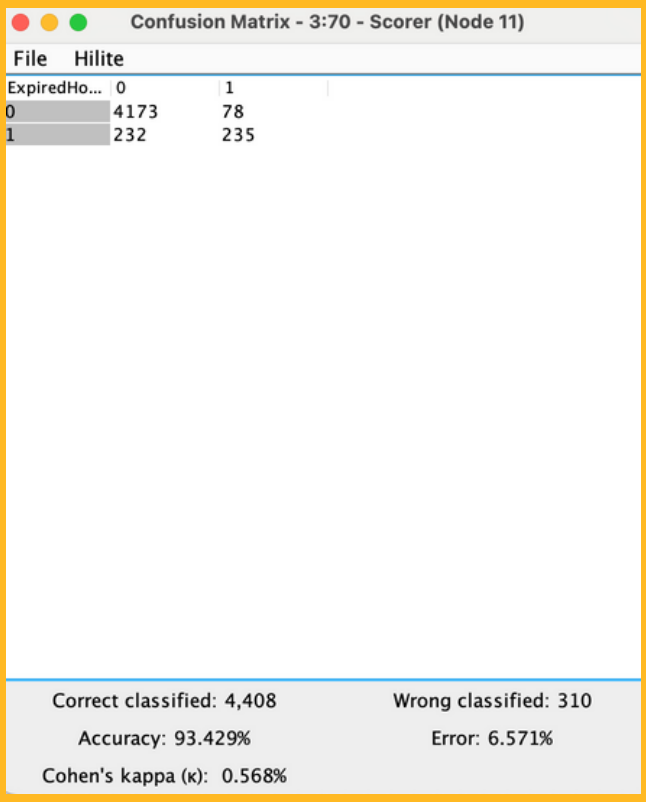
04 - Classification Techniques Used (22)

Tree Ensemble Predictor Configuration



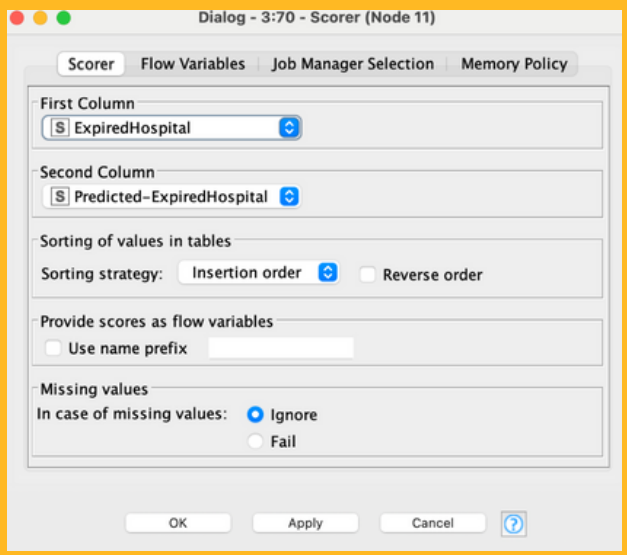
In the tree ensemble predictor configuration, the predicted column has been renamed to 'Predicted-ExpiredHospital'. This is akin to what has been done for the unknown dataset. The choice has been made to append the overall prediction confidence and append individual class probabilities for more effective analysis of the performance as more metrics are being displayed. They may also help to handle the imbalanced dataset while making the model more robust.

Tree Ensemble Scorer



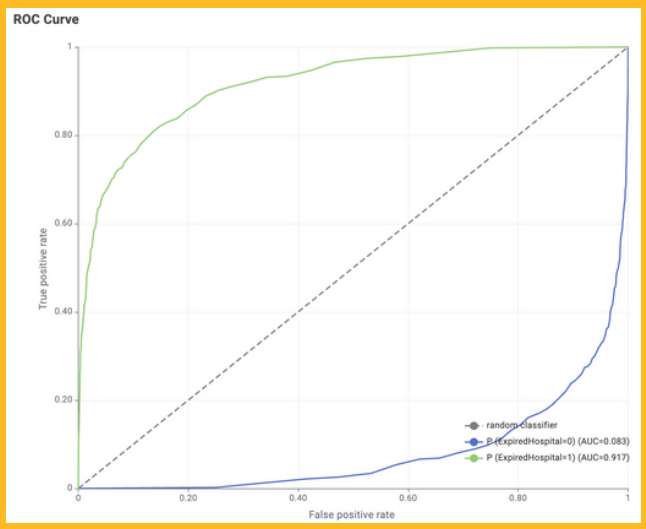
As one can see here, the model has predicted 4173 true negatives, 78 false positives, 232 false negatives and 235 true positives. The positive class has been set as 1, while the negative class is 0.

In the tree ensemble scorer, the first and second columns has the default settings of 'ExpiredHospital' and 'Predicted-ExpiredHospital'. The other settings such as the sorting strategy and ignoring missing values are also some of the default values from previous models. With the model, the scorer yielded an accuracy of 93.429% with an error of around 7%. The Cohen's Kappa (K) is at a 0.568%, which is on the upper bounds of agreement between two raters.

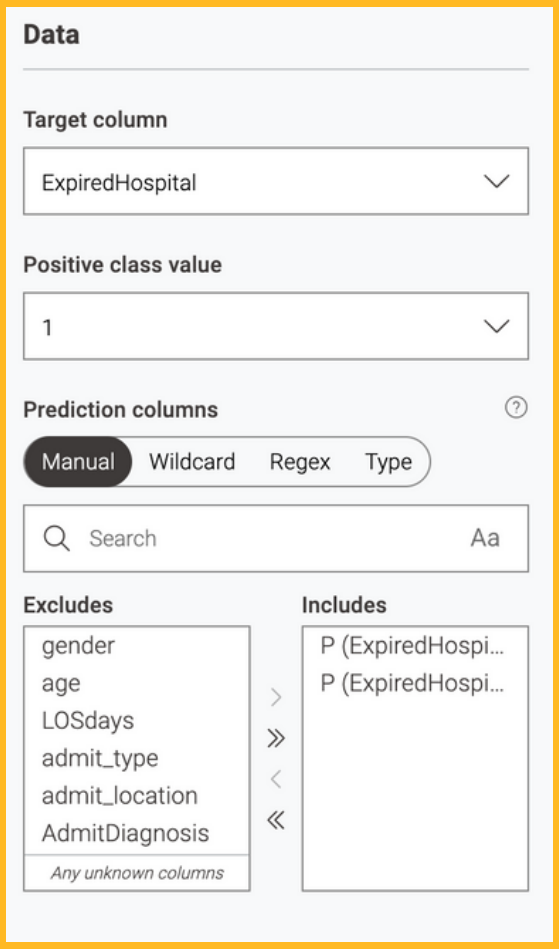


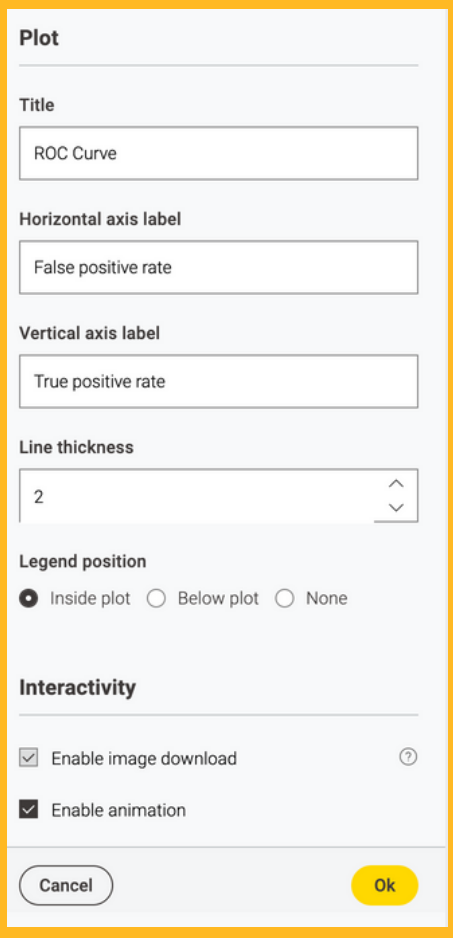
04 - Classification Techniques Used (23)

Tree Ensemble AUC-ROC Curve



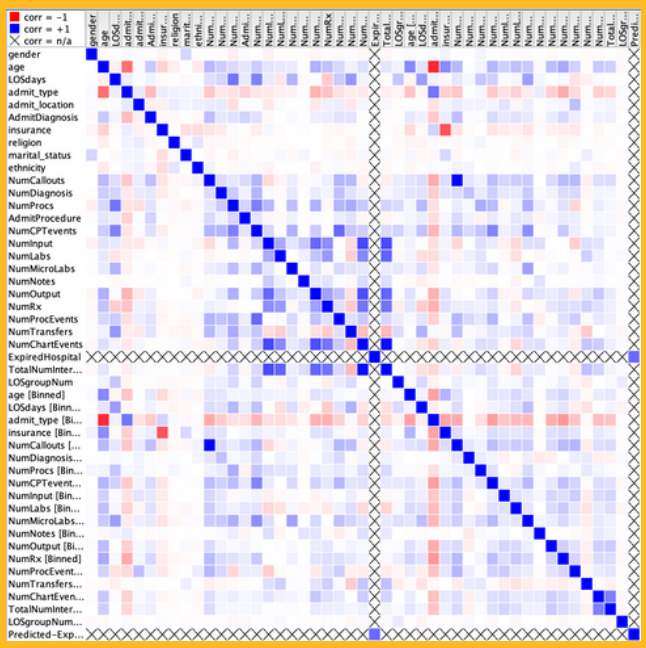
For the AUC-ROC Curve of the Tree Ensemble model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for Naive Bayes yielded an AUC score of 0.917, meaning that it is miles better than random guessing. The class of 0 yielded an AUC score of 0.083, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (24)

Tree Ensemble Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'LOSdays' and 'NumProcs'. However, it can also be seen that relationships such as the one between 'insurance' and 'age' is weak. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

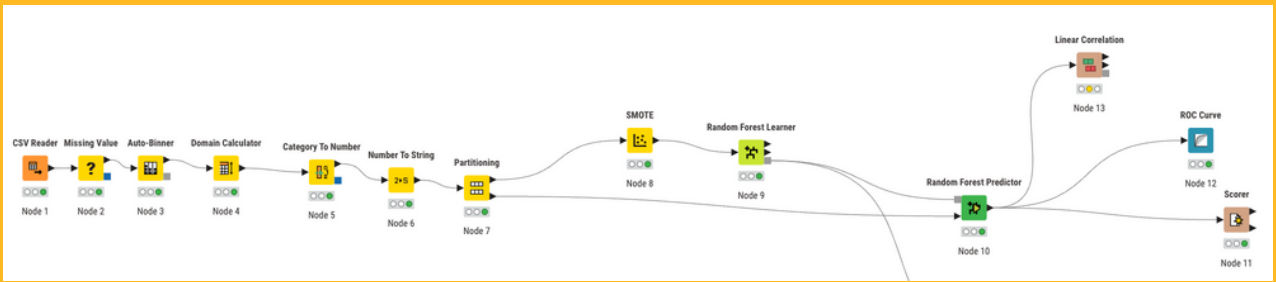
Tree Ensemble Brief Statistics

Attribute Class	Recall	Precision	F1
0	0.982	0.947	0.964
1	0.503	0.751	0.603

These are some basic statistics for the Tree Ensemble model. In terms of the 'Precision' as a key metric, the model has a precision of 0.947 for the '0' class and a precision of 0.751 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is above average. The recall values are 0.982 for the '0' class and 0.503 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it performs randomly for the '1' class. The 'F-Measure' or F1 scores are 0.964 for the '0' class and 0.603 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' and instances of '1' are about an okay standard.

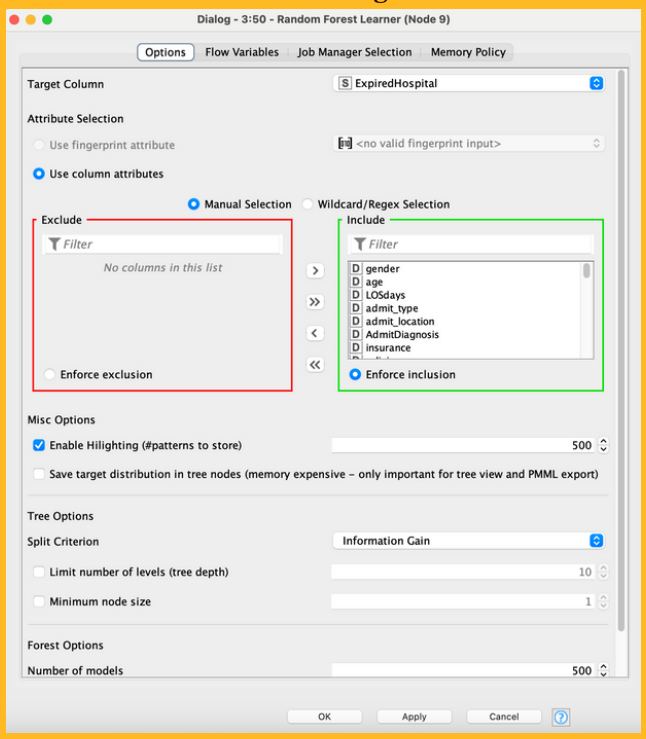
04 - Classification Techniques Used (25)

Random Forest (RF)



For the Random Forest model, the SMOTE node was used to handle class imbalance within the data. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The Random Forest Learner node acts as a learner that identifies as many rules and patterns as possible by utilising a random subset of the training data and features that reduces overfitting while enhancing predictive accuracy. The Random Forest Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

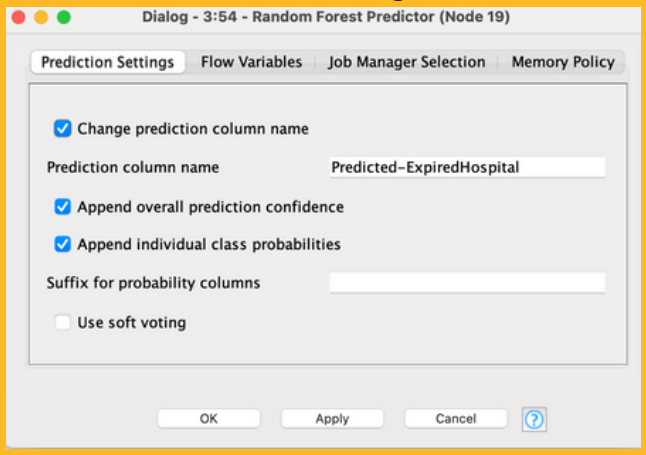
Random Forest Learner Configuration



The random forest learner configuration has been set to focus on the target column of 'ExpiredHospital'. It has been set to learn patterns in data from all of the attributes possible, including the binned columns. The split criterion has been set to use information gain as that yielded better results during testing. It has been set to generate 500 models while storing 500 patterns, to maximise the amount of information stored without the over-usage of memory. The learner has also been set to use the static random seed so that the results generated are consistent.

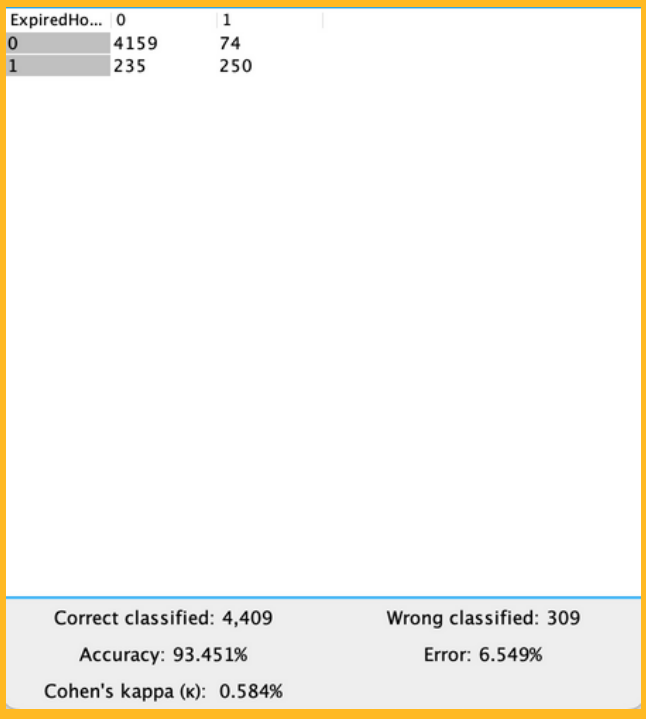
04 - Classification Techniques Used (26)

Random Forest Predictor Configuration



In the random forest predictor configuration, the predicted column has been renamed to 'Predicted-ExpiredHospital'. This is the same as to what has been done for the unknown dataset. The choice has been made to append the overall prediction confidence and append individual class probabilities for more effective analysis of the performance as more metrics are being displayed. They may also handle the imbalanced dataset while making it more robust.

Random Forest Scorer



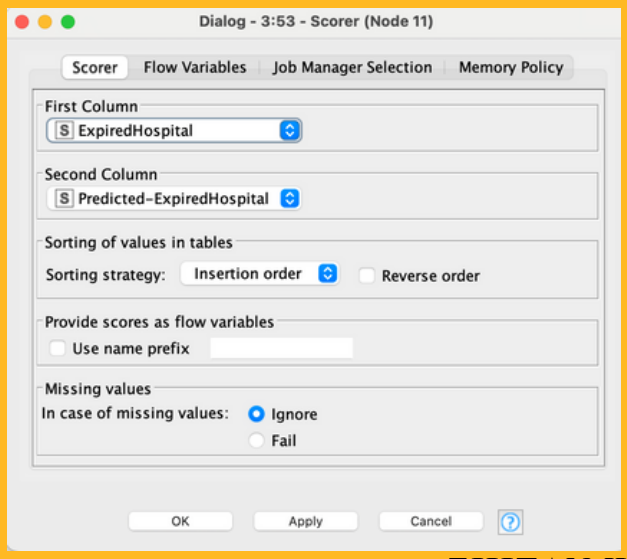
ExpiredHo...	0	1
0	4159	74
1	235	250

Correct classified: 4,409 Wrong classified: 309

Accuracy: 93.451% Error: 6.549%

Cohen's kappa (κ): 0.584%

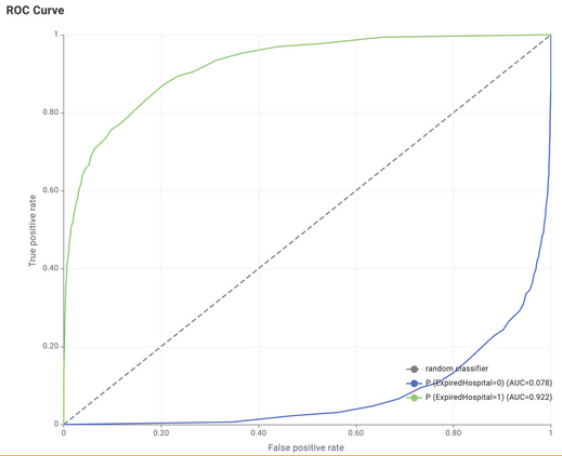
In the tree ensemble scorer, the first and second columns has the default settings of 'ExpiredHospital' and 'Predicted-ExpiredHospital'. The other settings such as the sorting strategy and ignoring missing values are also some of the default values from previous models. With the model, the scorer yielded an accuracy of 93.451% with an error of around 7%. The Cohen's Kappa (K) is at a 0.584%, which is on the upper bounds of agreement between two raters.



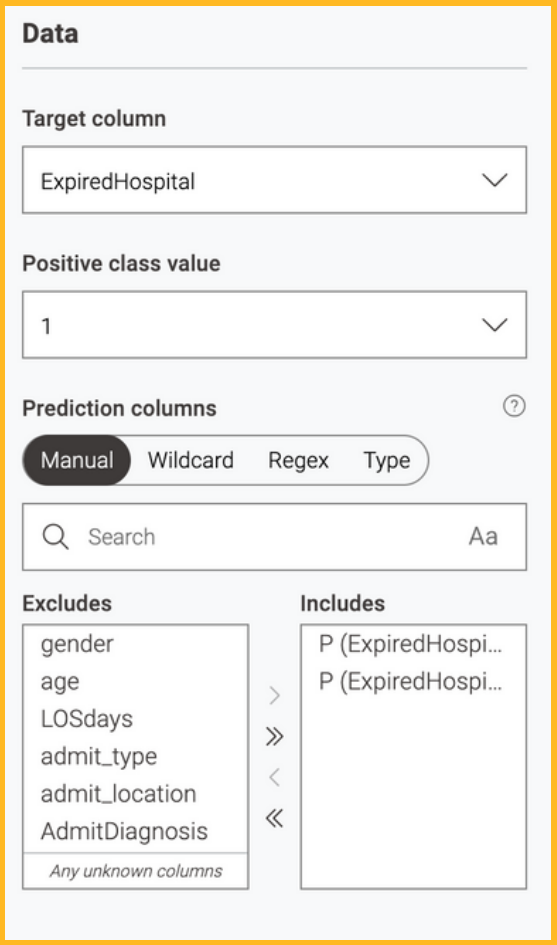
As one can see here, the model has predicted 4159 true negatives, 74 false positives, 235 false negatives and 250 true positives. The positive class has been set as 1, while the negative class is 0.

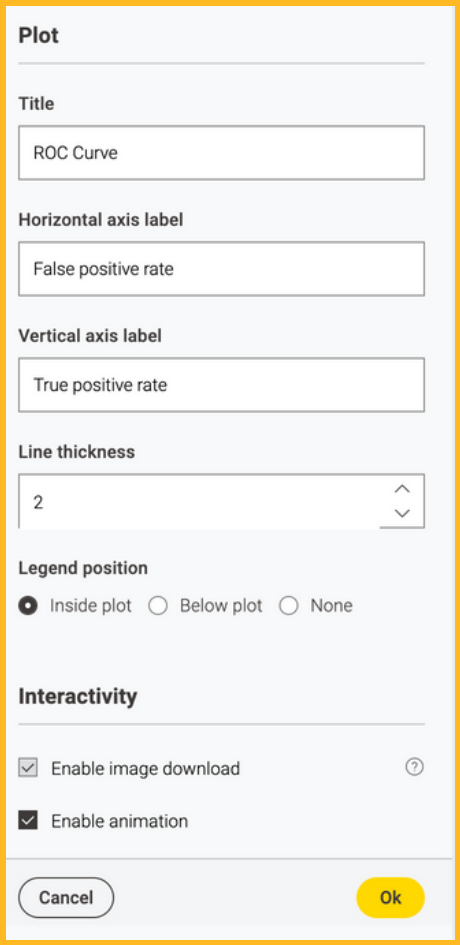
04 - Classification Techniques Used (27)

Random Forest AUC-ROC Curve



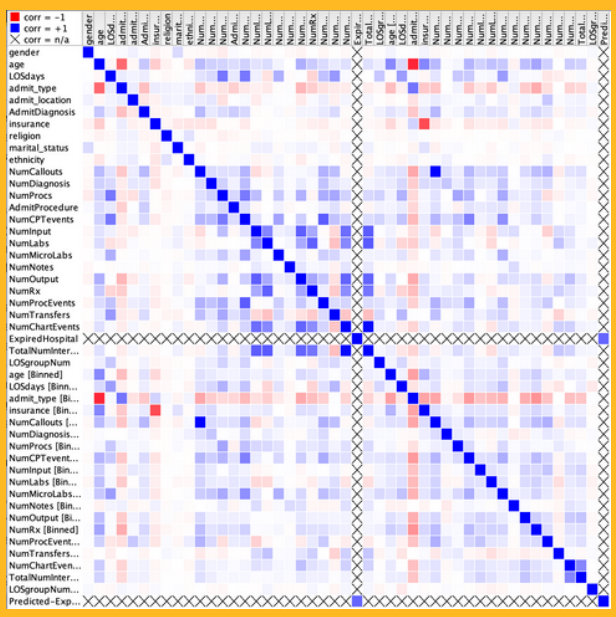
For the AUC-ROC Curve of the Random Forest model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for Random Forest yielded an AUC score of 0.922, meaning that it is miles better than random guessing. The class of 0 yielded an AUC score of 0.078, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (28)

Random Forest Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'NumDiagnosis' and 'AdmitDiagnosis'. However, it can also be seen that the relationships such as the one between 'gender' and 'NumProcs' is weak. One interesting correlation that can also be seen is how the correlation between 'ExpiredHospital' and 'gender' is 'N/A', meaning that gender has no influence on whether someone survives or dies during their treatment in the hospital. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

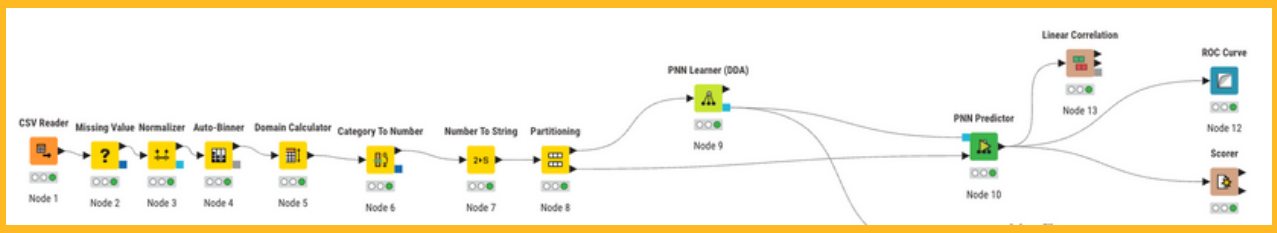
Random Forest Brief Statistics

Attribute Class	Recall	Precision	F1
0	0.979	0.949	0.964
1	0.54	0.749	0.628

These are some basic statistics for the Random Forest model. In terms of the 'Precision' as a key metric, the model has a precision of 0.949 for the '0' class and a precision of 0.749 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is acceptable. The recall values are 0.979 for the '0' class and 0.54 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it performs close to being random for the '1' class. The 'F-Measure' or F1 scores are 0.964 for the '0' class and 0.628 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' and instances of '1' are about an okay standard.

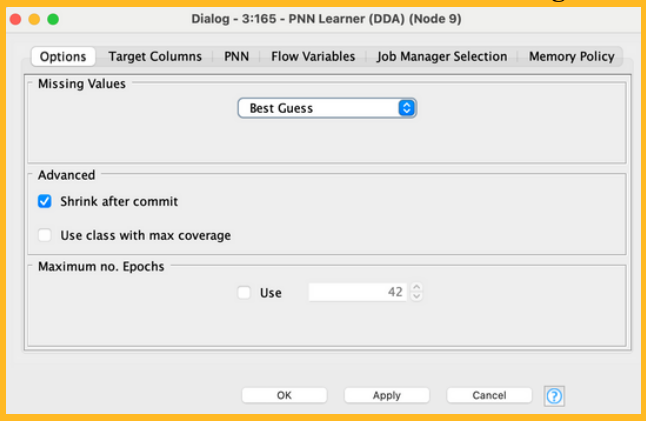
04 - Classification Techniques Used (29)

Probabilistic Neural Network (PNN)



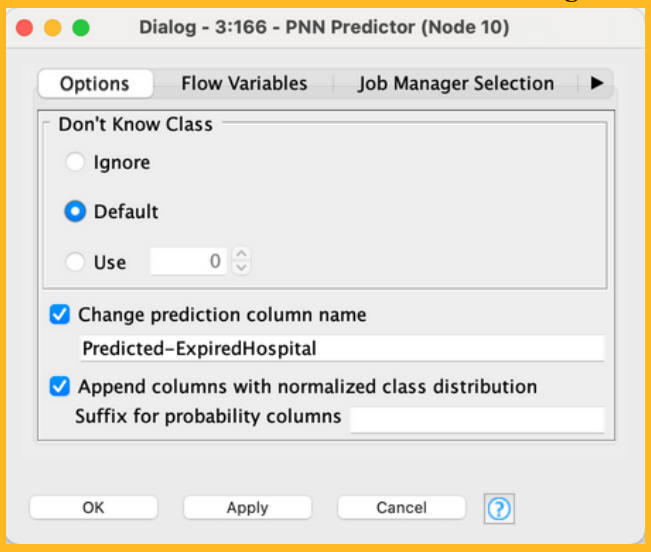
For the Probabilistic Neural Network model, the SMOTE node was not used to handle class imbalance within the data. This aims to make it similar to the MLP model due to how they are all Neural Network models. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The PNN Learner node acts as a learner that gets trained to identify as many rules and patterns as possible from the input, hidden and output layers. The PNN Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

Probabilistic Neural Network Learner Configuration



The PNN learner configuration has been set to focus on the target column of 'ExpiredHospital'. It has been set to fill in the possible missing values with the best guess, as a method to possibly enhance accuracy. The criteria for it to 'shrink after commit' has been checked so that it shrinks the model for it to be more interpretable and efficient, while reducing the risk of overfitting.

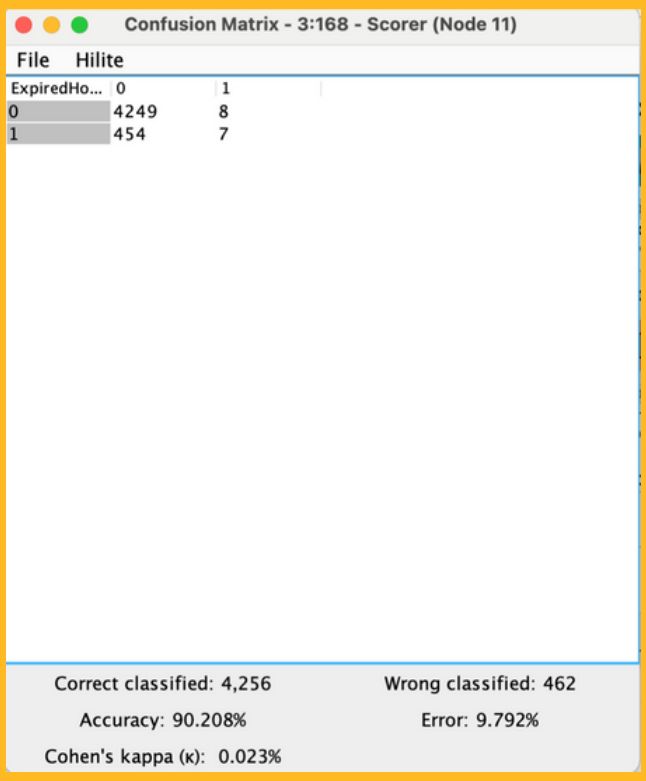
Probabilistic Neural Network Predictor Configuration



The PNN predictor has the 'don't know class' being set as the default value. The prediction column name has been altered to 'Predicted-ExpiredHospital', matching that of the unknown dataset. The final option to 'append columns with normalized class distribution' has been checked so that thorough analysis can be better performed since statistics are outputted.

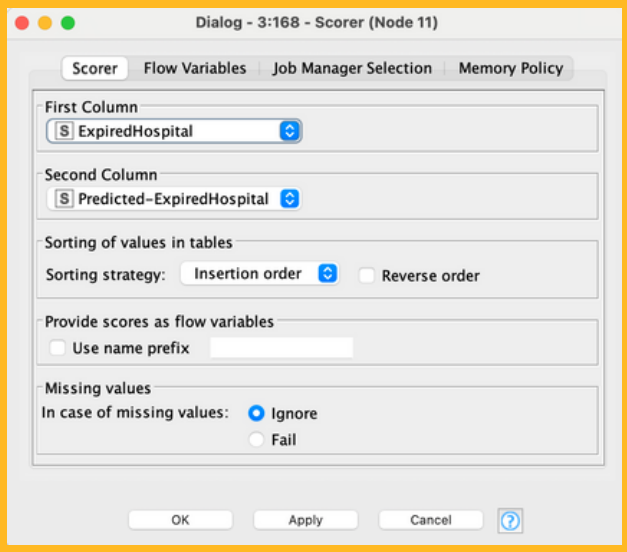
04 - Classification Techniques Used (30)

Probabilistic Neural Network Scorer



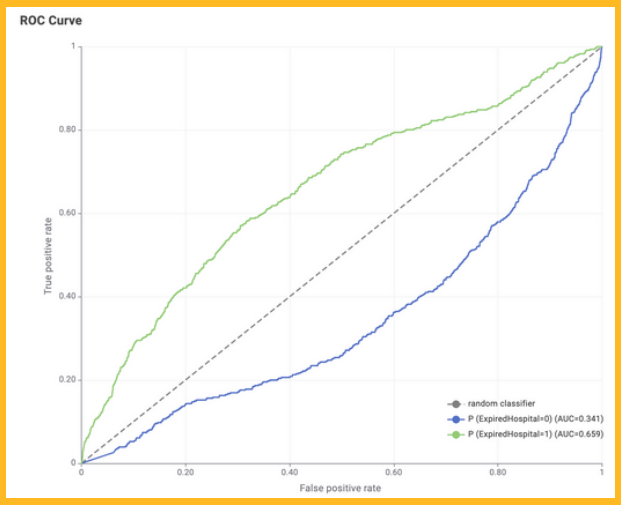
In the tree ensemble scorer, the first and second columns has the default settings of 'ExpiredHospital' and 'Predicted-ExpiredHospital'. The other settings such as the sorting strategy and ignoring missing values are also some of the default values from previous models. With the model, the scorer yielded an accuracy of 90.208% with an error of around 10%. The Cohen's Kappa (K) is at a 0.023%, which is equivalent to random chance for the agreement between two raters.

As one can see here, the model has predicted 4249 true negatives, 8 false positives, 454 false negatives and 7 true positives. The positive class has been set as 1, while the negative class is 0.

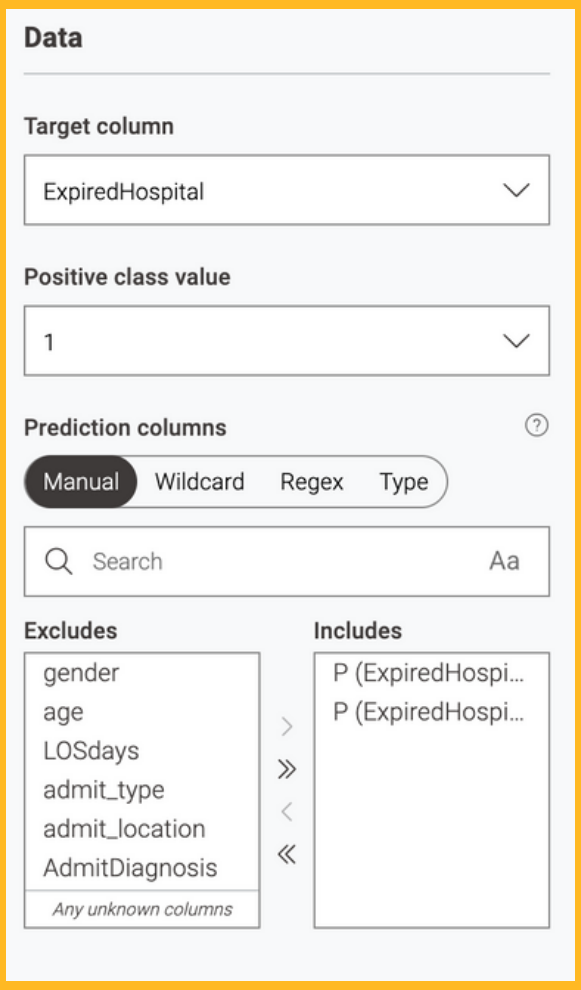


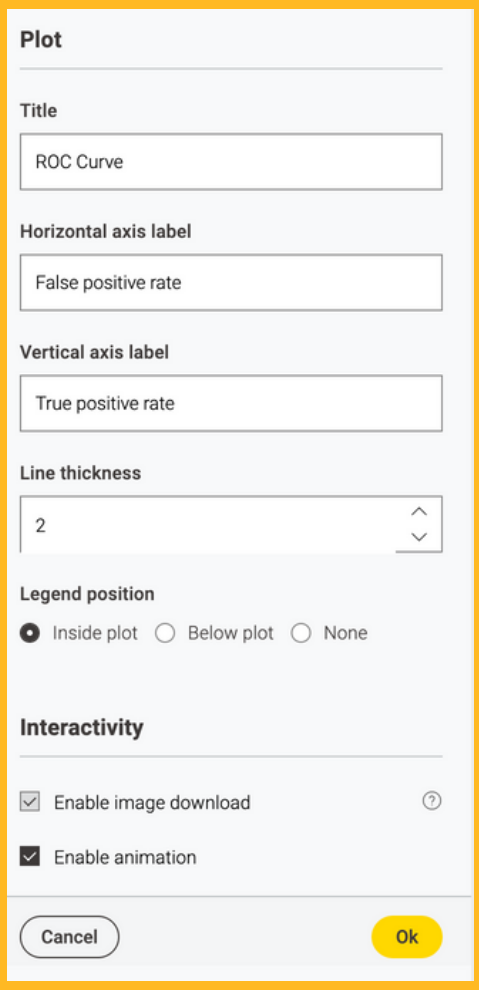
04 - Classification Techniques Used (31)

Probabilistic Neural Network AUC-ROC Curve



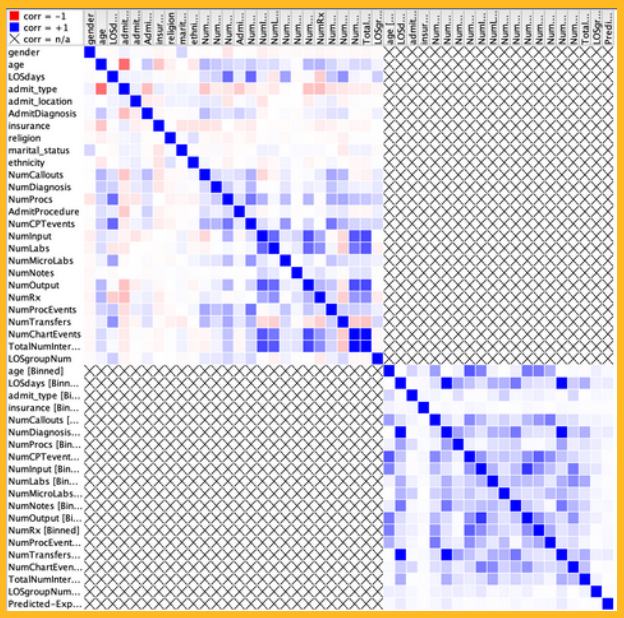
For the AUC-ROC Curve of the Probabilistic Neural Network model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for the Probabilistic Neural Network yielded an AUC score of 0.659, meaning that the values that it can predict are alright but not spectacular in terms of accuracy. The class of 0 yielded an AUC score of 0.341, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (32)

Probabilistic Neural Network Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'LOSdays' and 'NumTransfers'. However, it can also be seen that the relationships such as the one between 'NumChartEvents' and 'NumTransfers' is weak. One interesting detail that can be noticed is that for PNN, the binned columns does not have any correlation with the original attribute as they are denoted by 'N/A'. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

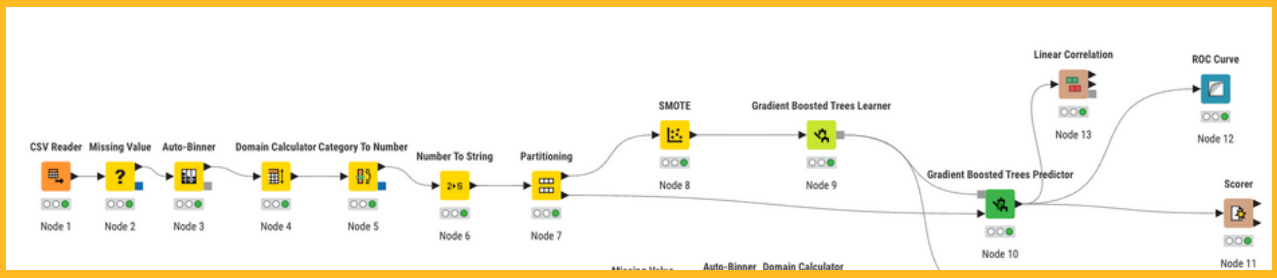
Probabilistic Neural Network Brief Statistics

Attribute Class	Recall	Precision	F1
0	0.998	0.903	0.948
1	0.015	0.467	0.029

These are some basic statistics for the Probabilistic Neural Network model. In terms of the 'Precision' as a key metric, the model has a precision of 0.903 for the '0' class and a precision of 0.467 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is close to random guessing. The recall values are 0.998 for the '0' class and 0.015 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it is particularly poor for the '1' class instance. The 'F-Measure' or F1 scores are 0.948 for the '0' class and 0.029 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' and instances of '1' are of a poor standard.

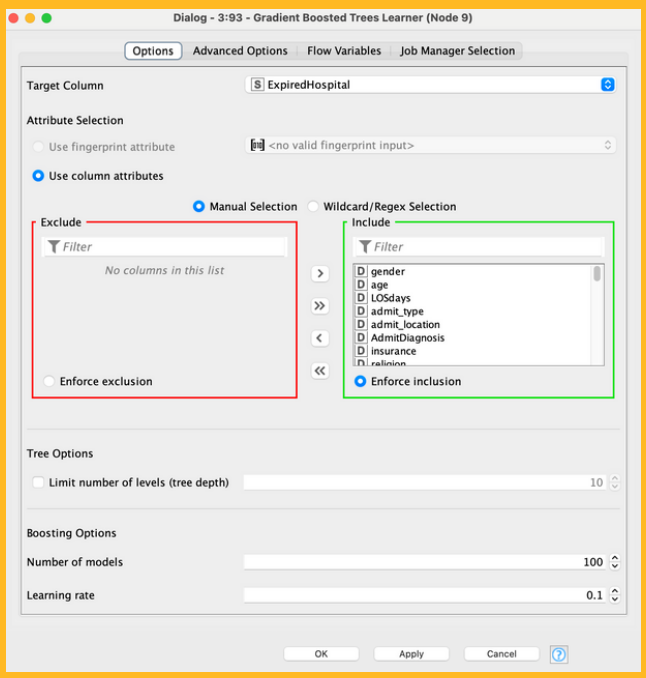
04 - Classification Techniques Used (33)

Gradient Boosted Tree (GBT)



For the Gradient Boosted Tree model, the SMOTE node was used to handle class imbalance within the data. This aims to make it similar to the Random Forest and Decision Tree models due to how they are all tree-based models. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The GBT Learner node acts as a learner that gets trained to identify as many rules and patterns as possible with the gradient descent optimisation algorithm during training. The GBT Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

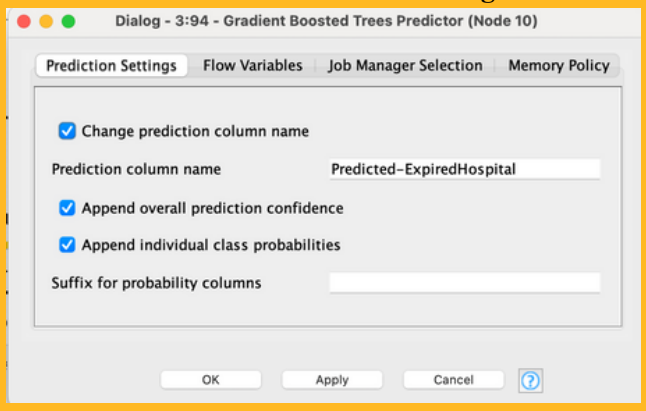
Gradient Boosted Tree Learner Configuration



The GBT learner configuration has been set to focus on the target column of 'ExpiredHospital'. It has been set to include all of the attribute columns, including those that has been encoded and those that has been binned. This ensures that the model learns from all of the features that are present for it to potentially make more accurate predictions. The other settings such as the number of models and learning rate are kept as default values.

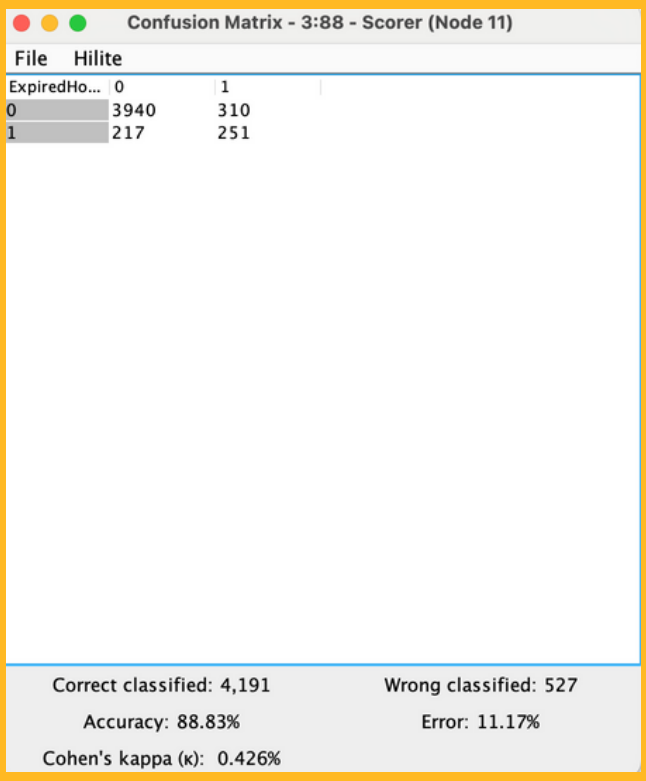
04 - Classification Techniques Used (34)

Gradient Boosted Tree Predictor Configuration



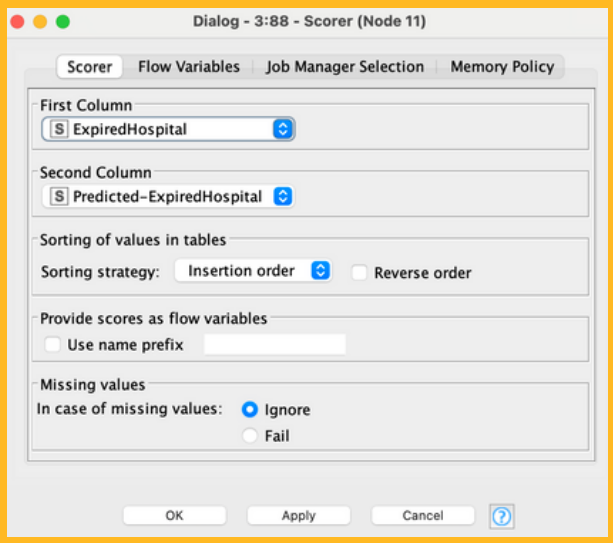
The GBT predictor has the prediction column name being altered to 'Predicted-ExpiredHospital', cohering with that of the unknown dataset. The following options to append overall prediction confidence and individual class probabilities has all been checked so that more thorough analysis can be done with the statistics.

Gradient Boosted Tree Scorer



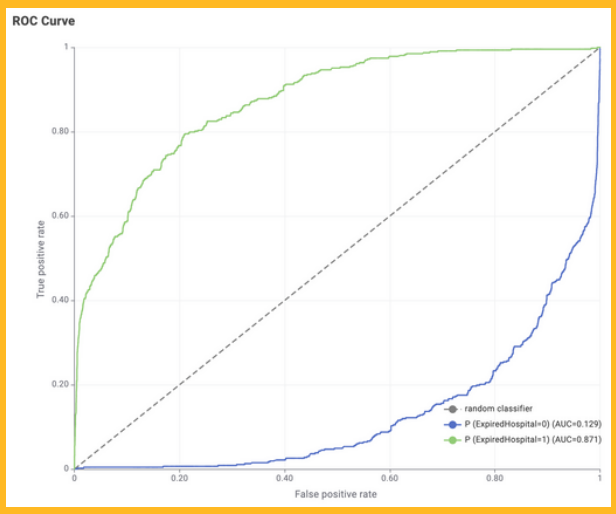
In the gradient boosted tree scorer, the first and second columns has the default settings of 'ExpiredHospital' and 'Predicted-ExpiredHospital'. The other settings such as the sorting strategy and ignoring missing values are also some of the default values from previous models. With the model, the scorer yielded an accuracy of 88.83% with an error of around 11%. The Cohen's Kappa (K) is at a 0.426%, which is equivalent to above average for the agreement between two raters.

As one can see here, the model has predicted 3940 true negatives, 310 false positives, 217 false negatives and 251 true positives. The positive class has been set as 1, while the negative class is 0.

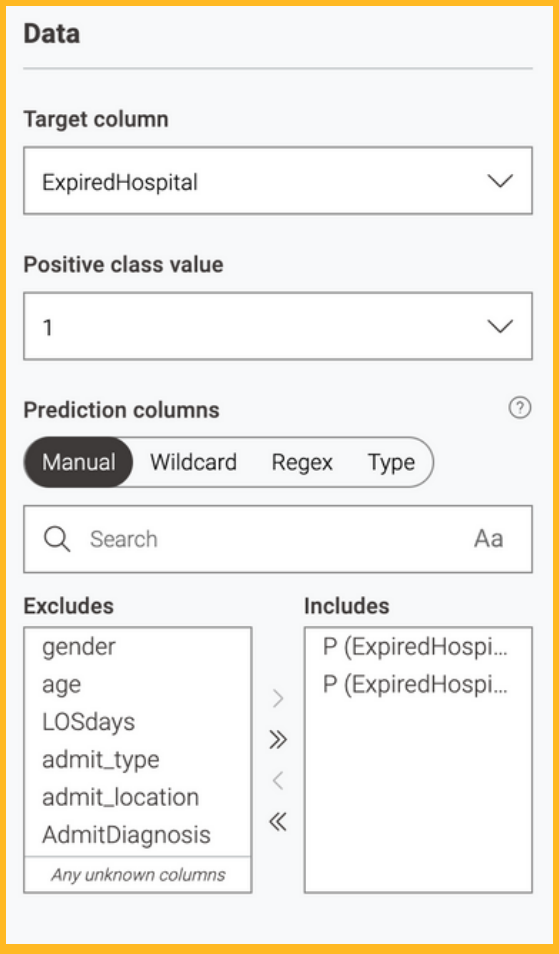


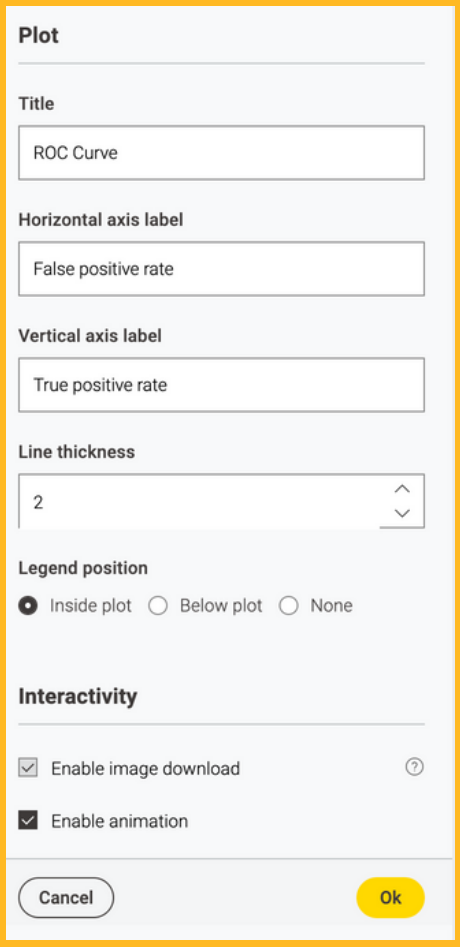
04 - Classification Techniques Used (35)

Gradient Boosted Tree AUC-ROC Curve



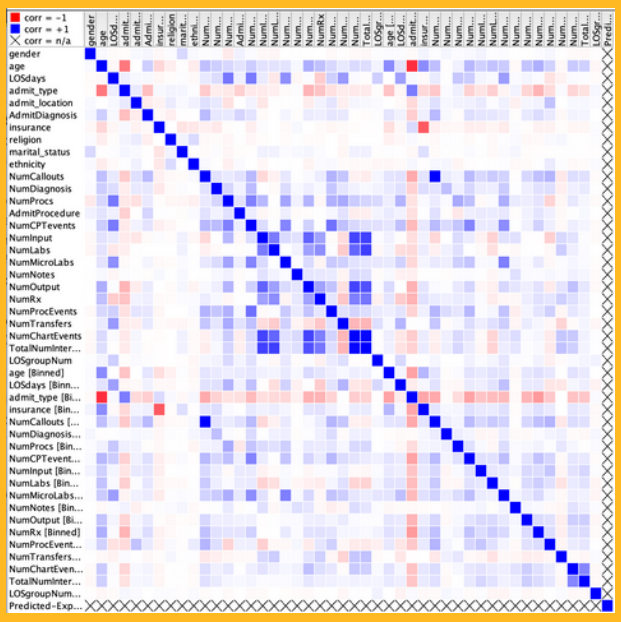
For the AUC-ROC Curve of the Gradient Boosted Tree model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for the Gradient Boosted Tree yielded an AUC score of 0.871, meaning that the values that it can predict are quite accurate but not perfect. The class of 0 yielded an AUC score of 0.129, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (36)

Gradient Boosted Tree Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'TotalNumInteract' and 'NumRx'. However, it can also be seen that the relationships such as the one between 'NumCallouts' and 'NumOutput' is weak. One interesting detail that can be noticed is that for GBT, every attribute column has a correlation with another attribute except for 'Predicted-ExpiredHospital' that relates to itself. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

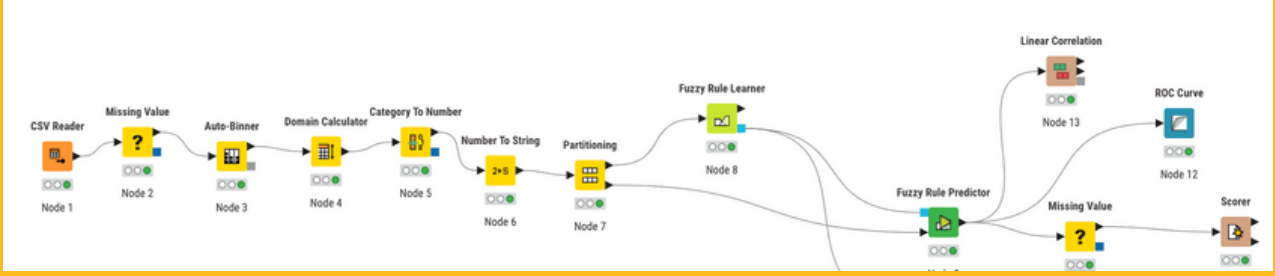
Probabilistic Neural Network Brief Statistics

Attribute Class	Recall	Precision	F1
0	0.927	0.948	0.937
1	0.536	0.447	0.488

These are some basic statistics for the Probabilistic Neural Network model. In terms of the 'Precision' as a key metric, the model has a precision of 0.948 for the '0' class and a precision of 0.447 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is close to random guessing. The recall values are 0.927 for the '0' class and 0.536 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it is only a bit better than random guessing for the '1' class instance. The 'F-Measure' or F1 scores are 0.937 for the '0' class and 0.488 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' and instances of '1' are guessed approximately.

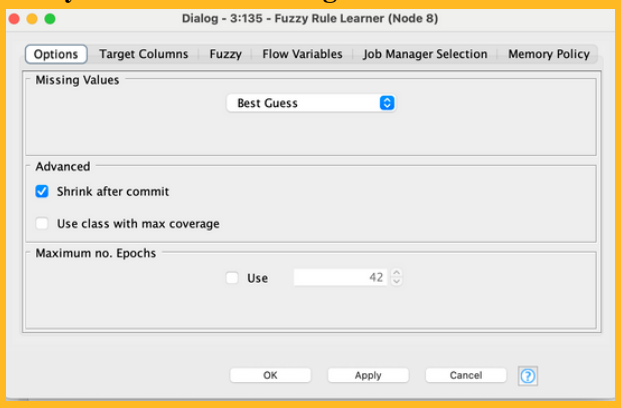
04 - Classification Techniques Used (37)

Fuzzy Rule (FR)



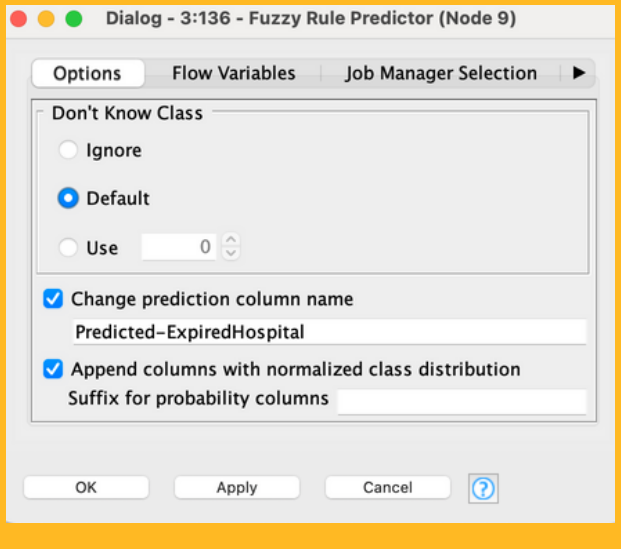
For the Fuzzy Rule model, the SMOTE node was not used to handle class imbalance within the data. This was done so that computation times are greatly reduced, however, the categorical attributes were still encoded using label encoding. The steps here were the same as what has been done to the PNN model. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The FR Learner node acts as a learner that gets trained to identify as many rules and patterns as possible with predefined Fuzzy Logic during training. The FR Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

Fuzzy Rule Learner Configuration



The FR learner configuration is the same as the PNN learner. It has been set to focus on the target column of 'ExpiredHospital'. It has been set to fill in the possible missing values with the best guess, as a method to possibly enhance accuracy. The criteria for it to 'shrink after commit' has been checked so that it shrinks the model for it to be more interpretable and efficient, while reducing the risk of overfitting.

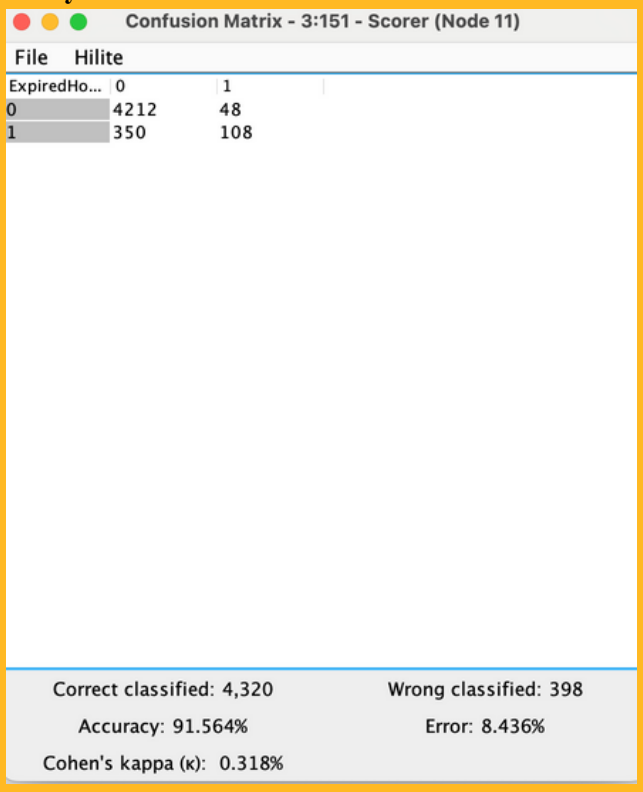
Fuzzy Rule Predictor Configuration



The FR predictor has the 'don't know class' being set as the default value. The prediction column name has been altered to 'Predicted-ExpiredHospital', matching that of the unknown dataset. The final option to 'append columns with normalized class distribution' has been checked so that thorough analysis can be better performed since statistics are outputted.

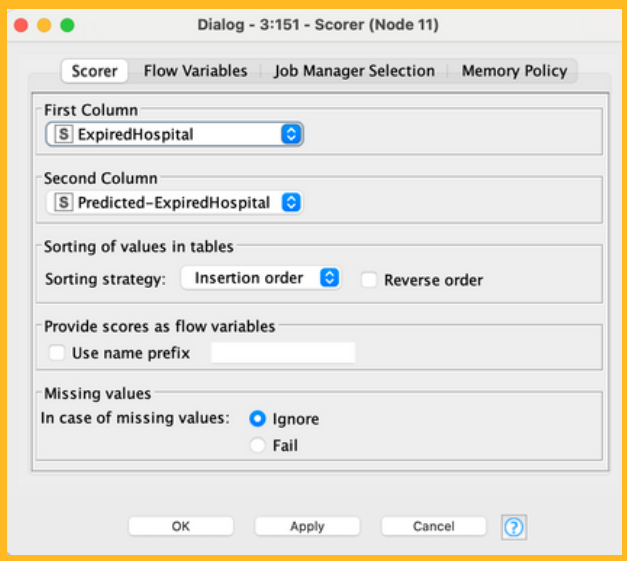
04 - Classification Techniques Used (38)

Fuzzy Rule Scorer



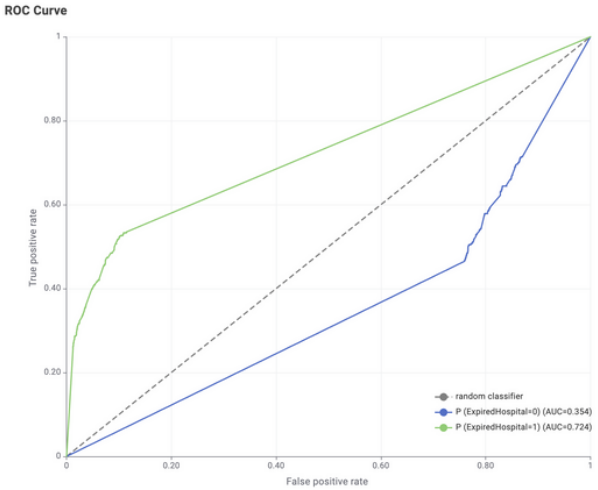
As one can see here, the model has predicted 4212 true negatives, 48 false positives, 350 false negatives and 108 true positives. The positive class has been set as 1, while the negative class is 0.

In the tree ensemble scorer, the first and second columns has the default settings of 'ExpiredHospital' and 'Predicted-ExpiredHospital'. The other settings such as the sorting strategy and ignoring missing values are also some of the default values from previous models. With the model, the scorer yielded an accuracy of 91.564% with an error of around 8%. The Cohen's Kappa (K) is at a 0.318%, which is better than random for the agreement between two raters.

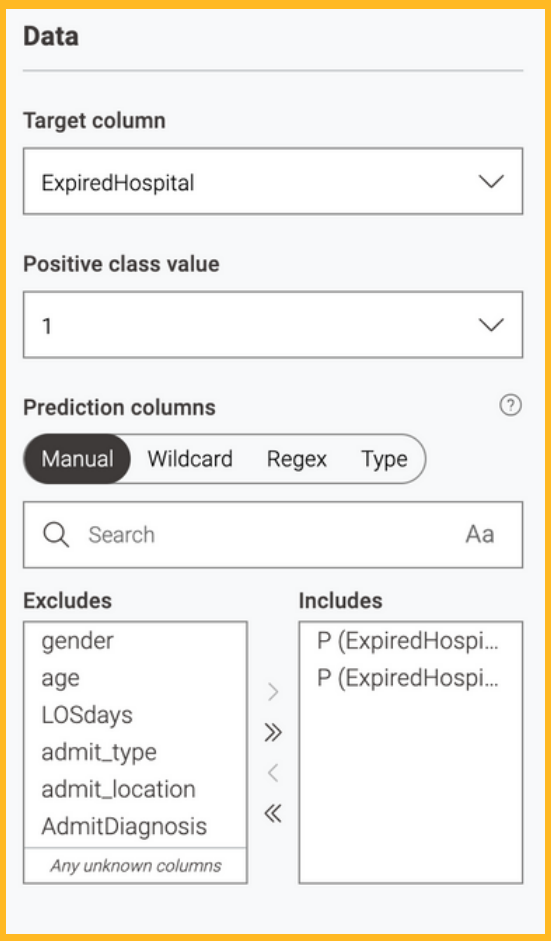


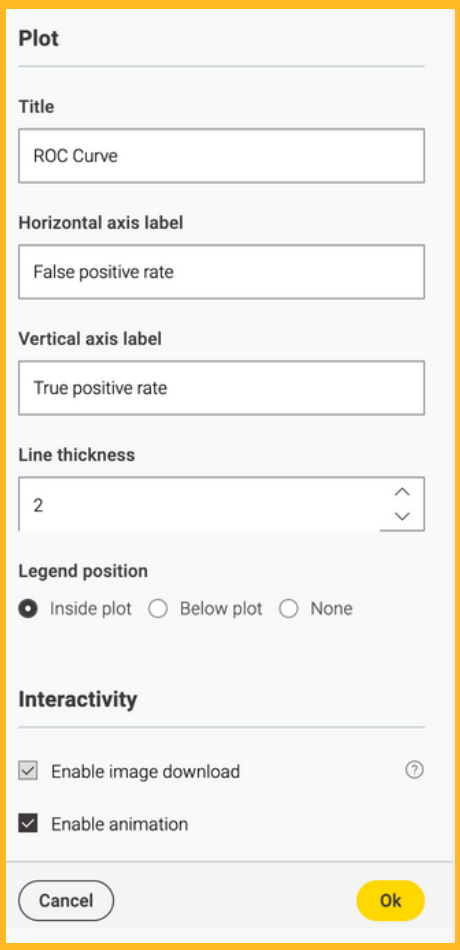
04 - Classification Techniques Used (39)

Fuzzy Rule AUC-ROC Curve



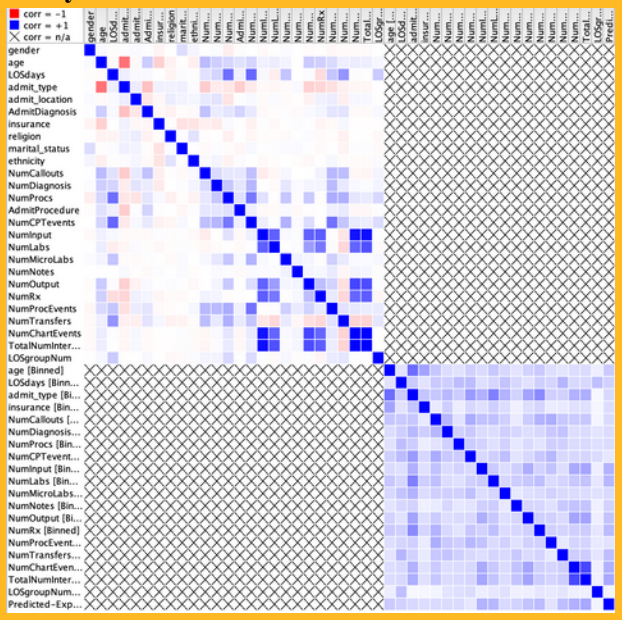
For the AUC-ROC Curve of the Fuzzy Rule model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for the Gradient Boosted Tree yielded an AUC score of 0.724, meaning that the values that it can predict are accurate although there is still room for improvement. The class of 0 yielded an AUC score of 0.354, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (40)

Fuzzy Rule Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'NumOutput' and 'NumProcEvents'. However, it can also be seen that the relationships such as the one between 'NumTransfers' and 'TotalNumInteract' is weak. One interesting detail that can be noticed is that for FR, there are no valid correlations between any binned attribute column with an original one. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

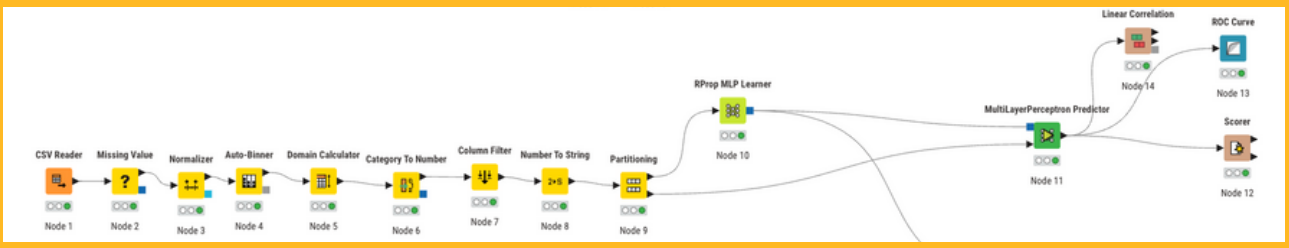
Fuzzy Rule Basic Statistics

Attribute Class	Recall	Precision	F1
0	0.989	0.923	0.955
1	0.236	0.692	0.352

These are some basic statistics for the Fuzzy Rule model. In terms of the 'Precision' as a key metric, the model has a precision of 0.923 for the '0' class and a precision of 0.692 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is fair. The recall values are 0.989 for the '0' class and 0.236 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it is poor for the '1' class instance. The 'F-Measure' or F1 scores are 0.955 for the '0' class and 0.352 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' and the capacity for instances of '1' is below that of random guessing.

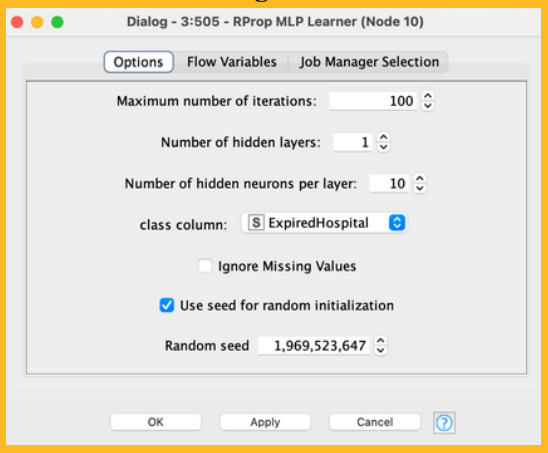
04 - Classification Techniques Used (41)

MultiLayer Perceptron (MLP)



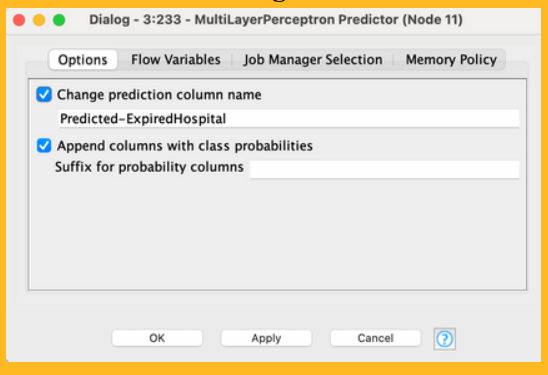
For the MLP model, the SMOTE node was not used to handle class imbalance within the data. This was done so that computation times are greatly reduced, however, the categorical attributes were still encoded using label encoding. The steps here were the same as what has been done to the PNN model. The partitioning of the data has been set as 90% training and 10% testing, so that the model can learn more patterns for it to make more accurate predictions. The MLP Learner node acts as a learner that gets trained to identify as many rules and patterns as possible with predefined interconnected neurons during training. The MLP Predictor node utilises the rules and data from the associated learner to make predictions on the given dataset.

MLP Learner Configuration



The MLP Learner has been configured to have an acceptable number of iterations, one that allows the model to run as thoroughly as possible. The hidden layer has been set to 1 for it to function simply and the hidden neurons has been set to 10 so that it would not have to use a lot of memory and running time. The class column has been set to 'ExpiredHospital', and the random seed initialization has been selected so that results can be reproduced.

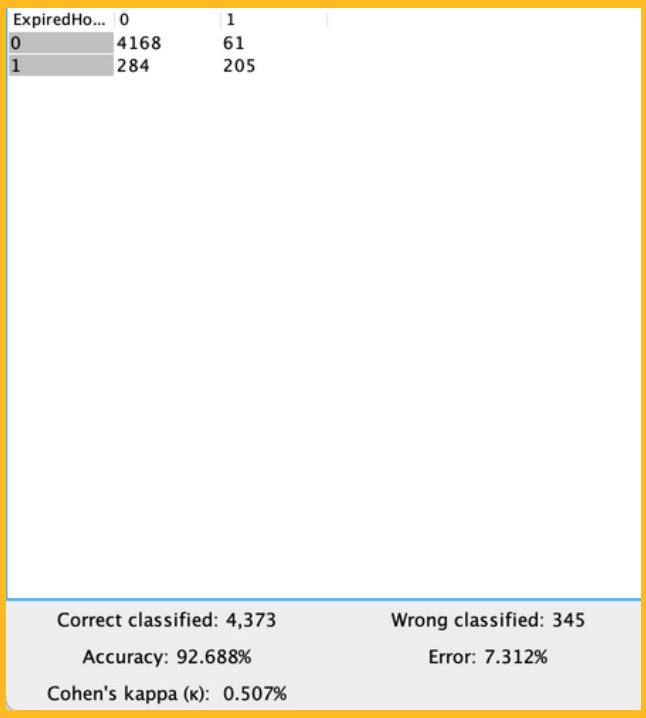
MLP Predictor Configuration



The configuration for the predictor is similar to some of the other models. The prediction column name has been altered to 'Predicted-ExpiredHospital'. It has also been set to append columns with the normalized class distribution as the data mining problem deals with an imbalanced dataset and it may help to improve the model performance.

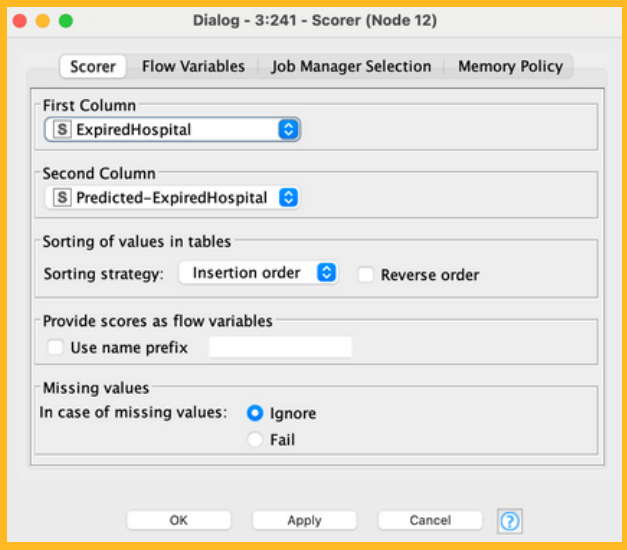
04 - Classification Techniques Used (42)

MLP Scorer



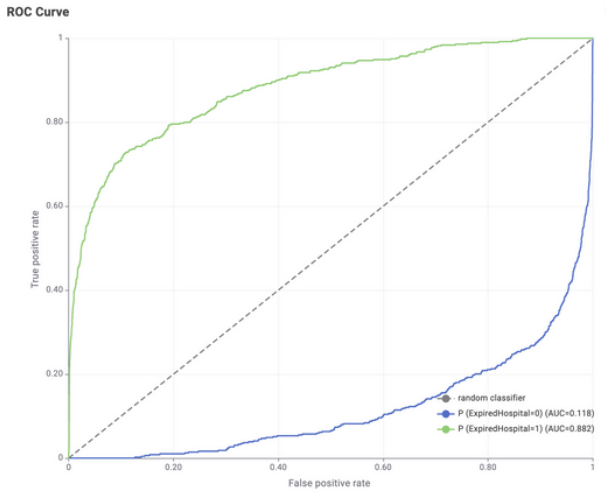
In the MLP scorer, the first and second columns has the default settings of 'ExpiredHospital' and 'Predicted-ExpiredHospital'. The other settings such as the sorting strategy and ignoring missing values are also some of the default values from previous models. With the model, the scorer yielded an accuracy of 92.688% with an error of around 7%. The Cohen's Kappa (K) is at a 0.507%, which is way better than random for the agreement between two raters.

As one can see here, the model has predicted 4168 true negatives, 61 false positives, 284 false negatives and 205 true positives. The positive class has been set as 1, while the negative class is 0.

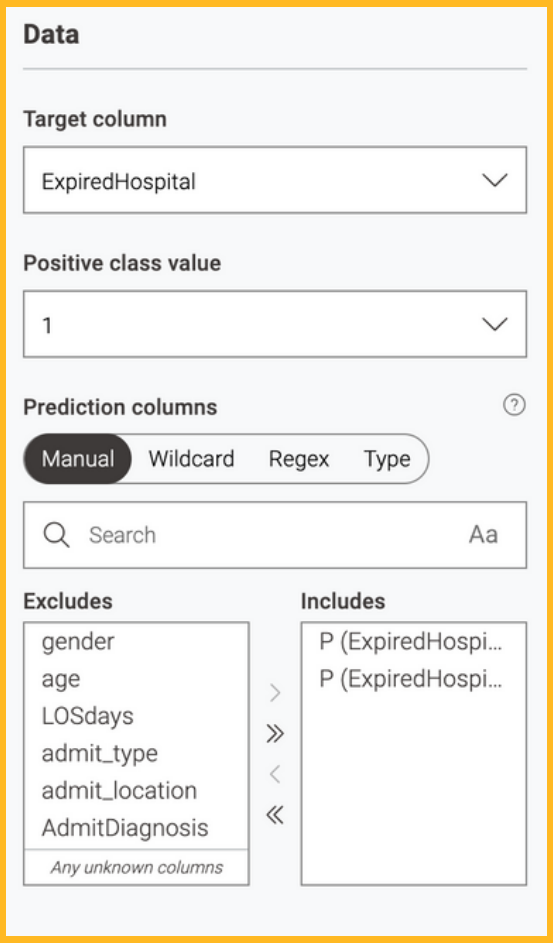


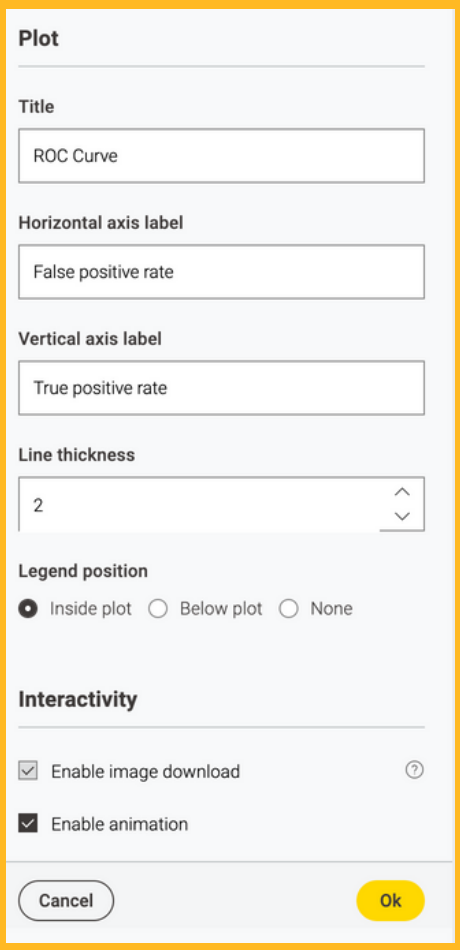
04 - Classification Techniques Used (43)

MLP AUC-ROC Curve



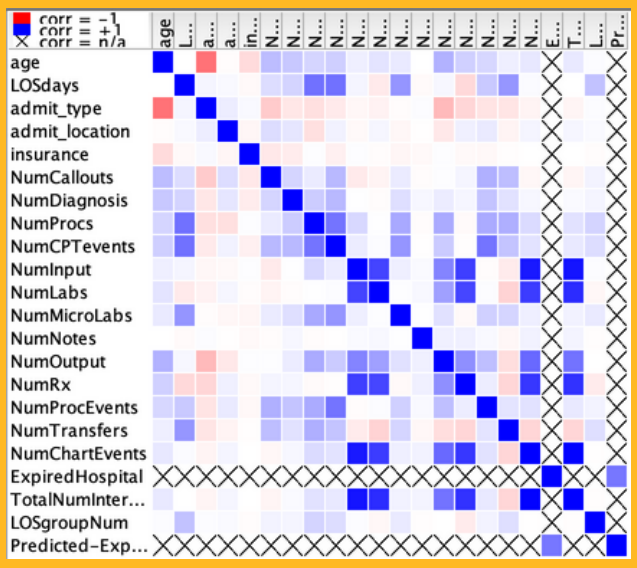
For the AUC-ROC Curve of the MLP model, the positive class value has been set to 1. This specifies that someone has died from the hospital during their treatment. The class of 1 for the MLP model yielded an AUC score of 0.882, meaning that the values that it can predict are quite accurate albeit not being perfect. The class of 0 yielded an AUC score of 0.118, which is the mirror image of the class of 1. The other attributes have been excluded as the primary focus is the target predicted class for 'ExpiredHospital'.





04 - Classification Techniques Used (44)

MLP Linear Correlation



From the linear correlation matrix that has been generated alongside binned columns, it can be seen that the attribute pairs have a mix of positive and negative correlations. It is a powerful tool to see how attributes are related with one another, for example, one of the relations that has a high correlation is what has been illustrated between 'NumLabs' and 'NumRx'. However, it can also be seen that the relationships such as the one between 'LOSdays' and 'NumRx' is weak. One interesting detail that can be noticed is that for MLP, there are no correlations with binned columns. This is possibly due to the lack of compatibility regarding data types. Overall, this is an instrumental tool that can be used to uncover patterns in data so that analysis can be more efficiently conducted.

MLP Basic Statistics

Attribute Class	Recall	Precision	F1
0	0.986	0.936	0.96
1	0.419	0.771	0.543

These are some basic statistics for the MLP model. In terms of the 'Precision' as a key metric, the model has a precision of 0.936 for the '0' class and a precision of 0.771 for the '1' class. This means that when the model predicts a positive instance, it has a high chance of being accurate for the '0' class while the '1' class is not bad. The recall values are 0.986 for the '0' class and 0.419 for the '1' class. It means that while the model is extremely great at capturing most of the positive instances for the '0' class, it is worse than random for the '1' class instance. The 'F-Measure' or F1 scores are 0.96 for the '0' class and 0.543 for the '1' class, highlighting that the model is extremely capable of predicting actual instances of '0' and the capacity for instances of '1' is slightly higher than random guessing.

05 - Best Classifier Selection

Here is a summary of all of the classifiers used and their results in terms of accuracy and AUC-ROC scores in one table. The random forest model was the one that has been chosen to classify the dataset as during testing, it achieved some of the best Kaggle submission scores. It also couples it with a high accuracy for the train/test, and the highest values for the AUC-ROC score and Cohen's accuracy among the models. The precision, recall and f1 scores were also quite high.

Model	Accuracy	AUC-ROC Score % (Positive Class)	Cohen's Kappa
Random Forest	93.451%	92.5%	0.584%
Tree Ensemble	93.429%	91.7%	0.568%
Gradient Boosted Trees	88.83%	87.1%	0.426%
Decision Trees	88.978%	69.6%	0.441%
K Nearest Neighbour	89.932%	52.1%	0.07%
MLP	92.688%	88.2%	0.507%
PNN	90.208%	65.9%	0.023%
Fuzzy Rule	91.564%	72.4%	0.318%
Naive Bayes	89.233%	89.4%	0.499%
SVM	90.752%	87.2%	0.155%

06 - Kaggle Submission/Conclusion

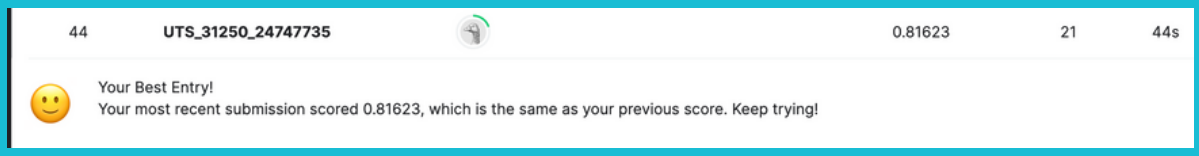
Best Classifier Performance

The best Kaggle submission details for the assignment:

Kaggle Submission Score: 0.81623

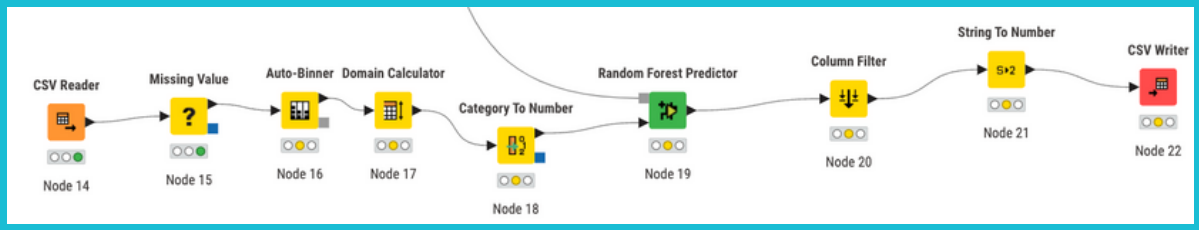
Classifier/Predictor Model Used: Random Forest

Final Rank: 33rd



Conclusion

This diagram illustrates the process of allowing the model to make predicted 'ExpiredHospital' values with the 'Assignment3-Unknown-Dataset.csv' file in the csv reader, using the Random Forest model.



Missing Value - This node fills in the missing values that may be present in the unknown dataset with a user determined value, particularly the ones that Microsoft Excel did not complete.

Auto-Binner - This node bins the attribute columns with the same number of bins as the ones used for the train/test values, ensuring that they match.

Domain Calculator - This defines the range of possible values that label encoding can generate, similar to the preprocessing.

Category-To-Number - This is the main label encoding node, converting the categorical attributes into a set numerical form to match what has been done to the train/test.

Random Forest Predictor - This is the main predictor node for the target attribute of 'ExpiredHospital', which grabs the data and pattern recognition features from the associated 'Learner' node.

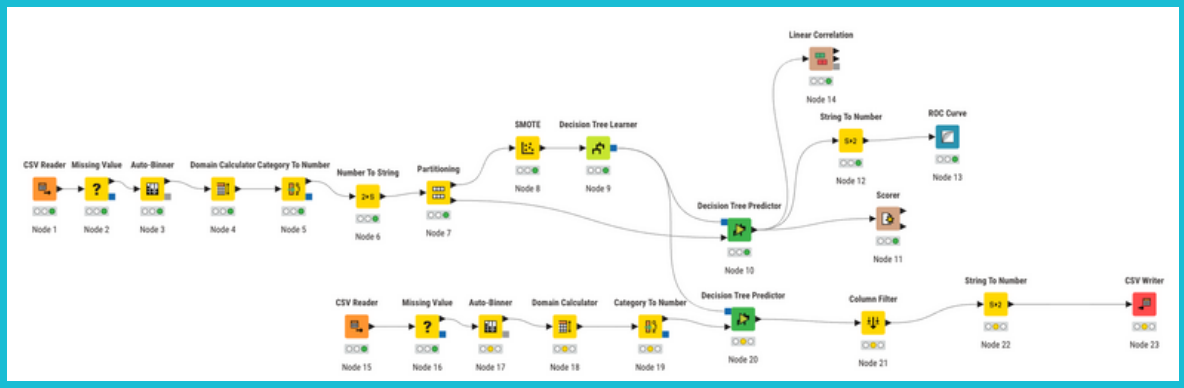
Column Filter - This node filters out the unrelated columns as a result of other preprocessing steps and transformations, leaving the predicted 'ExpiredHospital' as the only one remaining.

String-To-Number - This node converts the predicted values into actual numbers (integers) for the CSV writer node to function properly.

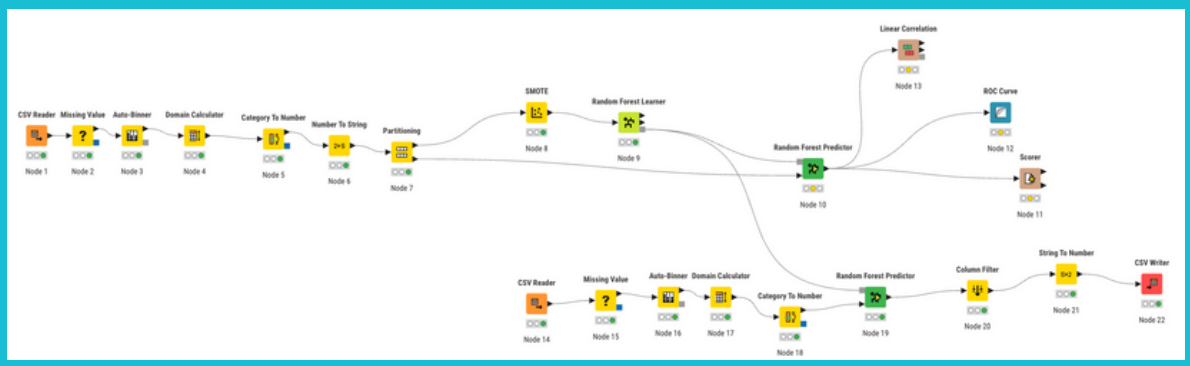
CSV Writer - This node converts the generated predicted values into a CSV file that can be opened by Microsoft Excel. Conditions for column and row ID generation can be specified and a directory to save the file on their computer before executing it can be stipulated.

07 - Appendix

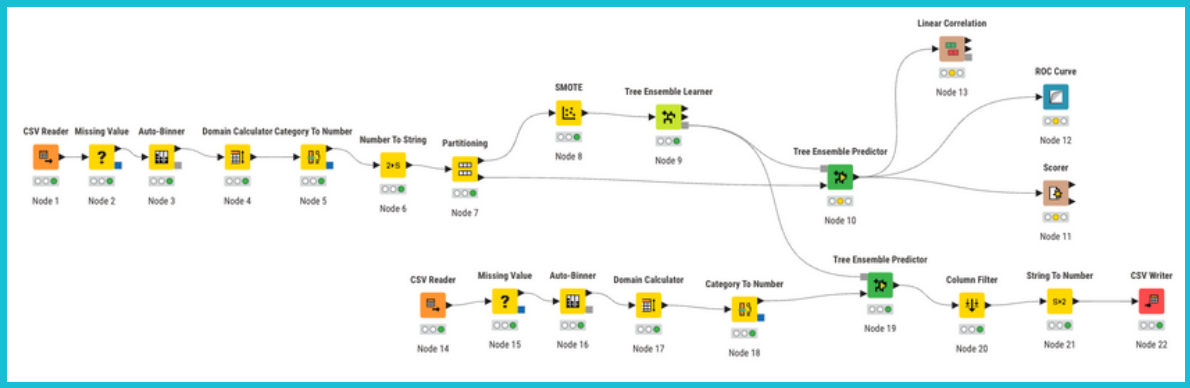
Full KNIME Models



Decision Tree

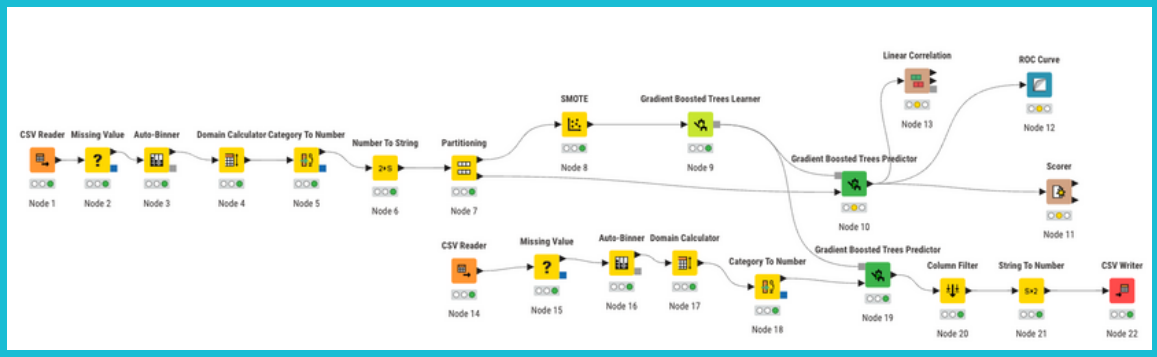


Random Forest

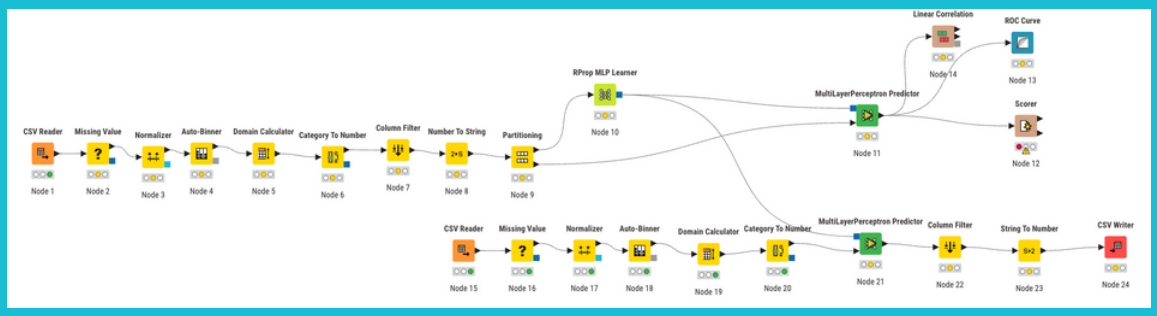


Tree Ensemble

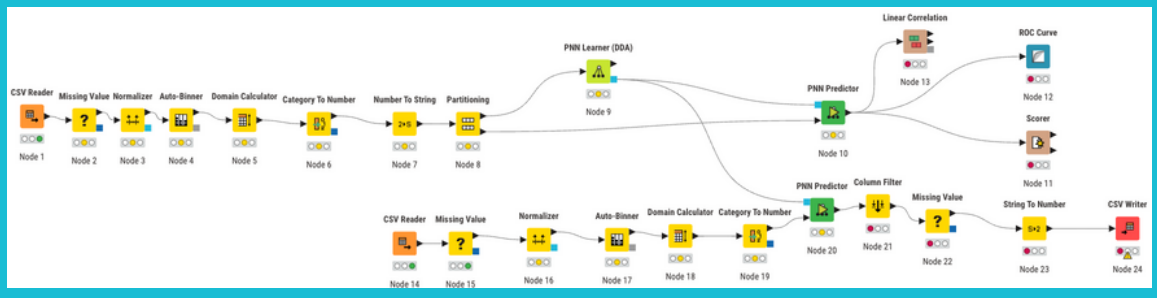
07 - Appendix



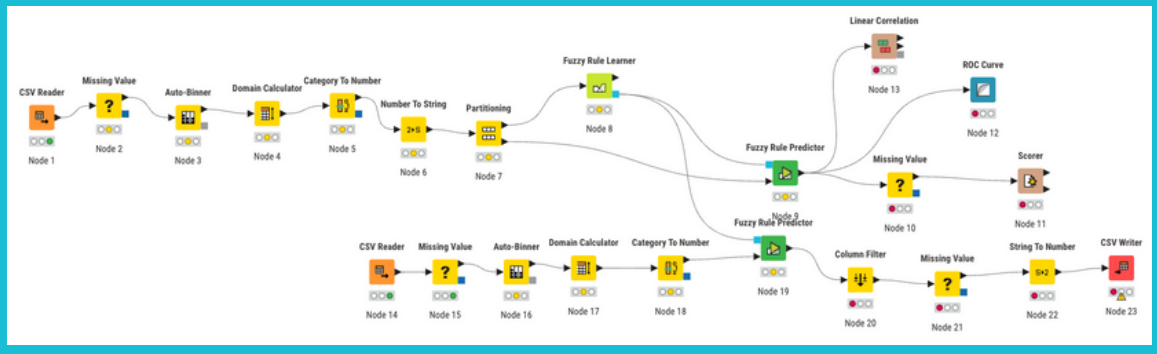
Gradient Boosted Tree



MLP

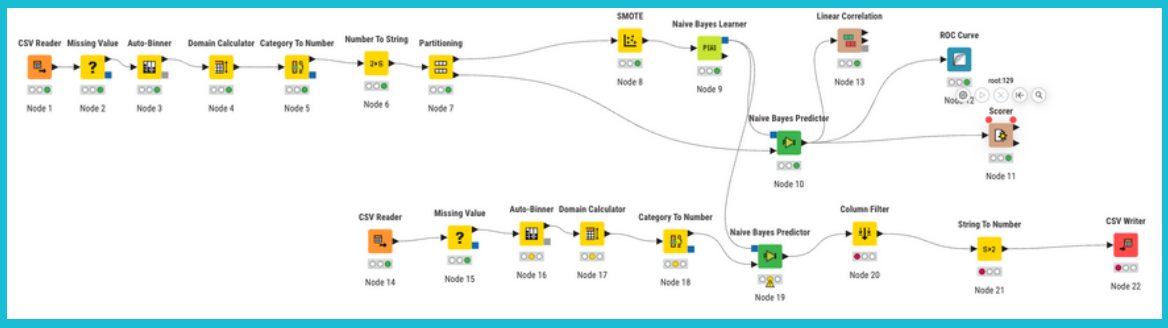


PNN

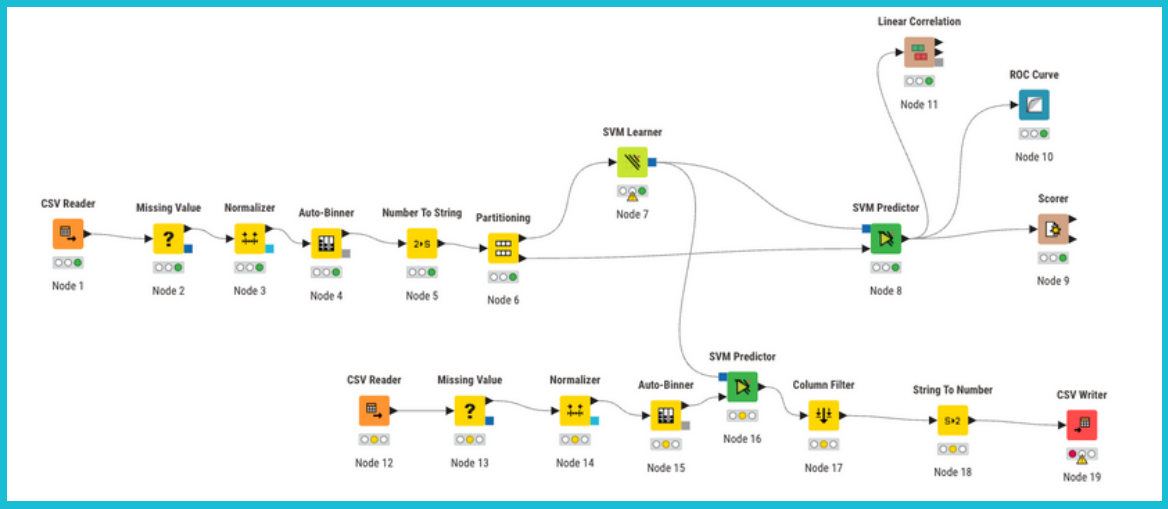


Fuzzy Rule

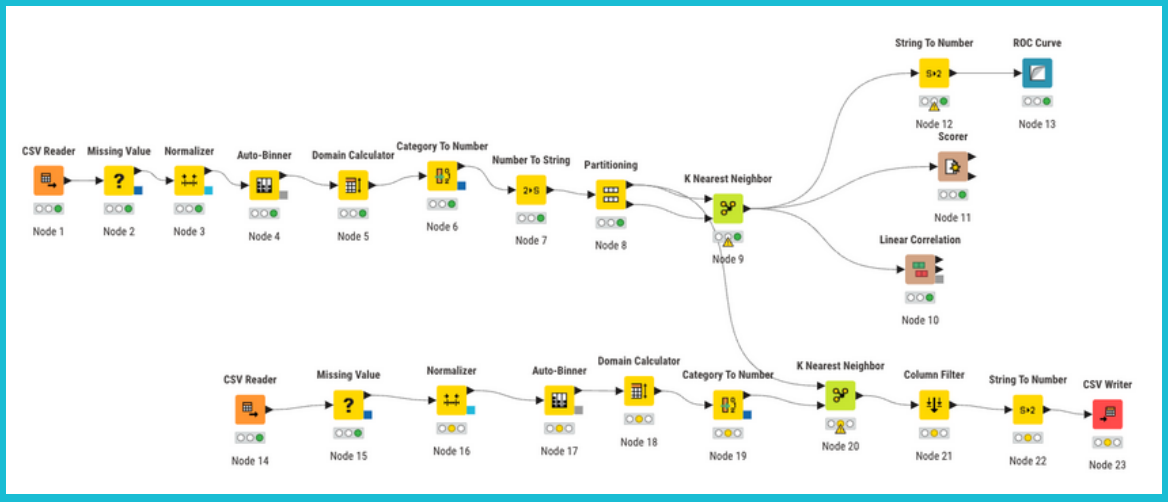
07 - Appendix



Naive Bayes



Support Vector Machine



K-Nearest Neighbour