DATA MINING IN ACTion

UTS Advanced Data Analytics Micro-Credential



Jason Wong

Telstra - Group #2

Table of Contents

[1. Introduction 3](#_Toc121559141)

[1.A The Data Mining Problem 3](#_Toc121559142)

[1.B Training, Testing and Unknown Datasets 3](#_Toc121559143)

[2. Data Pre-processing and Transformation 4](#_Toc121559144)

[2.A Number to String: Converting Target Attribute 4](#_Toc121559145)

[2.B Data Partitioning 4](#_Toc121559146)

[2.C Normalizing Values 4](#_Toc121559147)

[2.D Conversion of String to Date & Time values 5](#_Toc121559148)

[2.E Feature Selection and Missing Values 5](#_Toc121559149)

[3. Solving the Problem & Classifiers Used 8](#_Toc121559150)

[3.A Decision Tree 8](#_Toc121559151)

[3.B Tree Ensemble 8](#_Toc121559152)

[3.C Random Forest 9](#_Toc121559153)

[3.D K-Nearest Neighbour 9](#_Toc121559154)

[3.E Support Vector Machine 9](#_Toc121559155)

[3.F Multi-layer Perceptron (MLP) 10](#_Toc121559156)

[4. Evaluating Classifiers 12](#_Toc121559157)

[4.A Comparison of Classifier Models 12](#_Toc121559158)

[4.B Final Classifier Model Selected 12](#_Toc121559159)

[5. Reflection 13](#_Toc121559160)

[Appendix 14](#_Toc121559161)

[A. Attribute Description 14](#_Toc121559162)

[B. Knime Diagram 13](#_Toc121559163)

Table of Figures

[Figure 1 - Image of the Training/Testing Dataset 3](#_Toc121559164)

[Figure 2 - Number to String 4](#_Toc121559165)

[Figure 3 - Normalizer 5](#_Toc121559166)

[Figure 4 - String to Date&Time 5](#_Toc121559167)

[Figure 5 - Removing Outliers 6](#_Toc121559168)

[Figure 6 – Removing Duplicate attributes and Binarization of Description Attributes 7](#_Toc121559169)

[Figure 7 - SVM model in Knime 10](#_Toc121559170)

[Figure 8 - SVM parameters selected 10](#_Toc121559171)

[Figure 9 - MLP model in Knime 11](#_Toc121559172)

[Figure 10 - MLP Parameters 11](#_Toc121559173)

# Introduction

## 1.A The Data Mining Problem

Presented in this report is an analysis of a property/housing dataset and the steps taken to produce a Machine Learning classifier model that is able to both accurately and reliably predict the discrete ‘Qualified’ attribute of a property in an Unknown dataset.

This report will also outline the pre-processing steps taken to clean-up the dataset, training the classifier models, the evaluation of different classifier models, the effect of different parameter settings and justification for the final classifier model selected.

The training dataset contains details about several thousand properties, each properties features and the respective sales data. The aim of our classifier will be to accurately categorize the ‘Qualified’ attribute of each data entry: 0 if it is not qualified (U), and 1 for qualified (Q).

This analysis was primarily completed using the KNIME data analytics platform, with some initial analysis of the training dataset conducted in Excel.

## 1.B Training, Testing and Unknown Datasets

In this project we have been provided with two datasets: a Housing dataset with known ‘Qualified’ attribute containing 75008 unique data entries and an Unknown dataset containing 32148 unique data entries.

In order to produce our classifier models the Housing dataset has been randomly partitioned into separate Training and Testing sets.

Additionally, a brief description of the attributes in the dataset have also been provided (Appendix A).

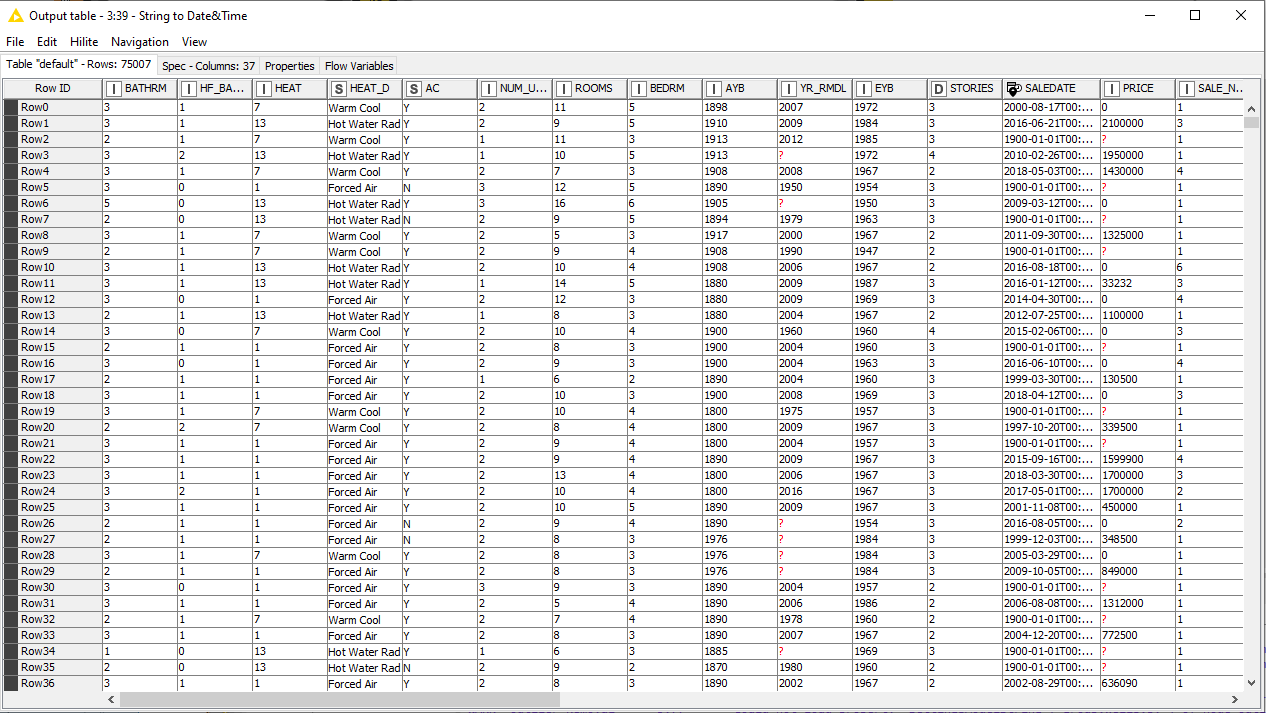


Figure - Image of the Training/Testing Dataset

# Data Pre-processing and Transformation

After loading the training and unknown datasets into Knime, the next step in training our classifier models involves the pre-processing and transformation of the raw dataset. The intent of these steps is to minimize the effect of incomplete or inconsistent data and therefore improve the accuracy of our classifier models.

## 2.A Number to String: Converting Target Attribute

For each classifier learner model, the ‘Number To String’ node is first used to convert the ‘Qualified’ attribute from an integer value to a string value. This is necessary as the target attribute in the class column of our Learner methods must be discrete categorical values.

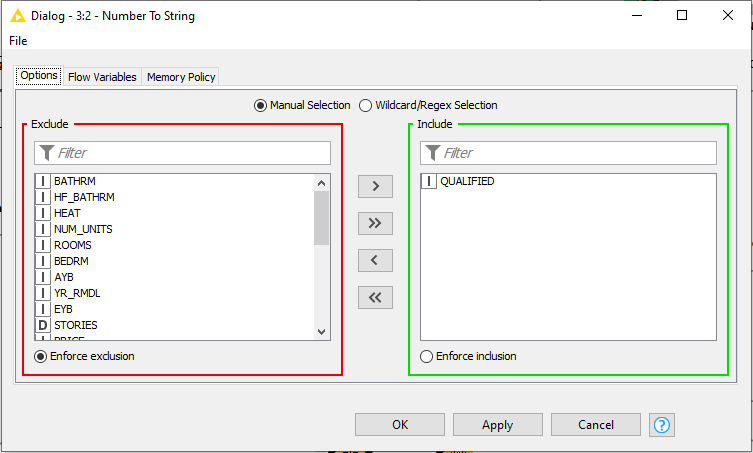


Figure - Number to String

## 2.B Data Partitioning

To validate and refine the classifier models we examine in this report, our training dataset is split into two smaller data subsets: training and testing.

We initially used the default 70-30 Relative % split with data drawn randomly. After evaluating and selecting our final classifier model this Relative split was increased to a 90-10 ratio, this higher ratio was selected to improve the accuracy of the chosen model.

## 2.C Normalizing Values

In order for the Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP) classification methods to work correctly, the Numerical (Integer) values in the dataset need to be Normalized.

This has the effect of improving the accuracy and integrity of the data while also optimising computational efficiency. The ‘Normalizer’ node in Knime is used with Min-Max normalisation.

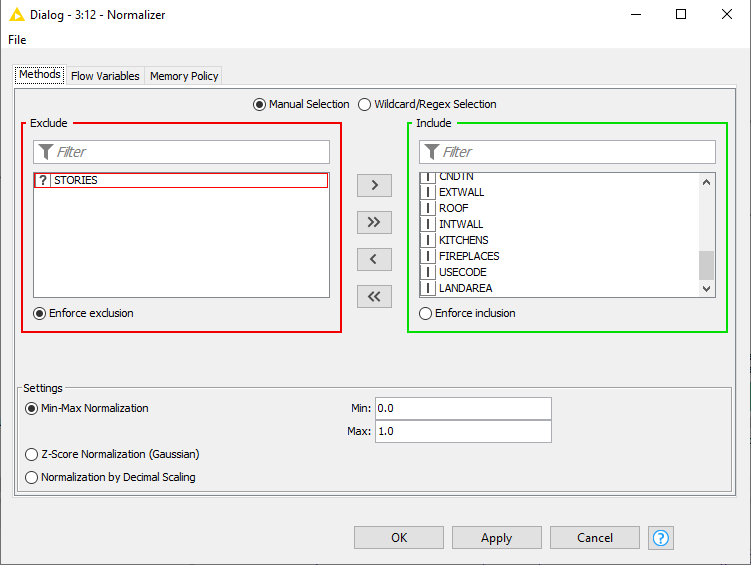


Figure - Normalizer

## 2.D Conversion of String to Date & Time values

As the SALEDATE attribute is an Interval rather than Nominal value, these data entries were converted from string values to a Date Time values. This type conversion allows the ordering of this data and could improve the training process.

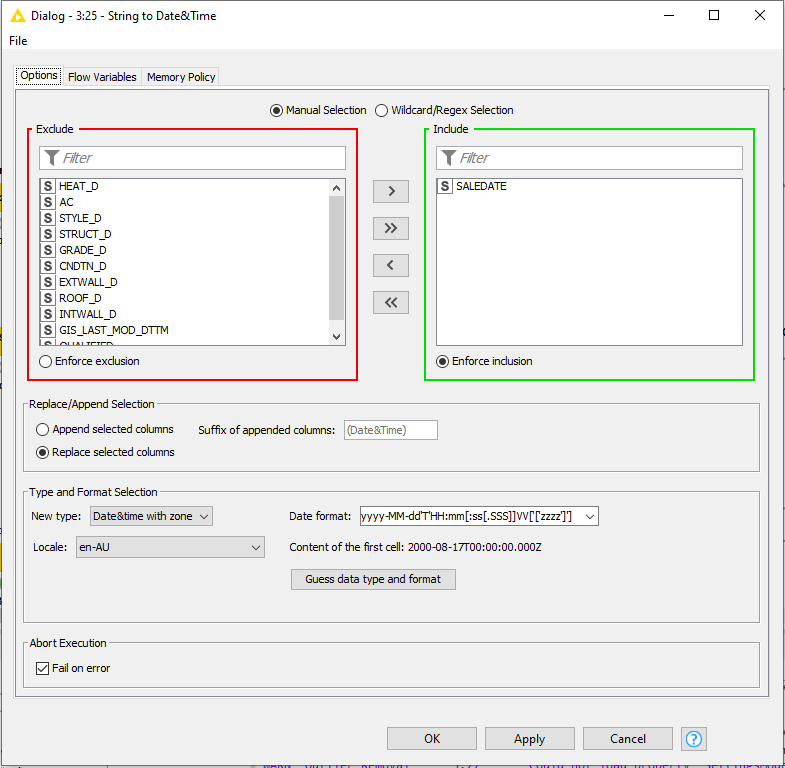


Figure - String to Date&Time

## 2.E Feature Selection and Missing Values

The training dataset was incomplete, containing a large number of missing data values particularly for the PRICE and YR\_RMDL attributes. Careful consideration was given as to how to treat these missing values and outliers.

Some of the classifier models are naturally capable of dealing with missing values treating them as distinct null values. For the classifier models which couldn’t handle missing values, we trialled using the median, mean and most frequent values before opting for the mean value.

20 erroneous data entries were found with blank values for the majority of the attributes, these values were removed from all of Classifiers using the Training dataset.

The effect of outliers in the dataset was also taken into account, using the Outlier Removal Extension node values that were greater than 3 standard deviations from the mean were excluded from the training data. However, removing these outliers was found to have a negligible effect when using the Random Forest classifier, there simply weren’t enough outliers in the dataset to have an effect on the results.

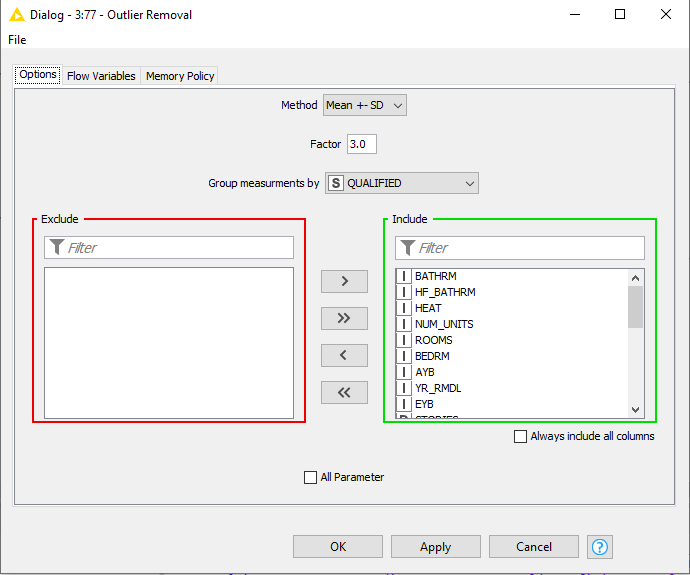


Figure - Removing Outliers

For the SVM and MLP methods matching string attributes (HEAT\_D, STYLE\_D, STRUCT\_D, GRADE\_D, CNDTN\_D, EXTWALL\_D, ROOF\_D, INTWALL\_D) were removed as they simply categorized the values already found in another corresponding attribute column.

Initially, these values were binarized into multiple variables using the ‘One-to-Many’ method in Knime, however, this was found to significantly increase computational effort without much benefit in improving the classifier accuracy.

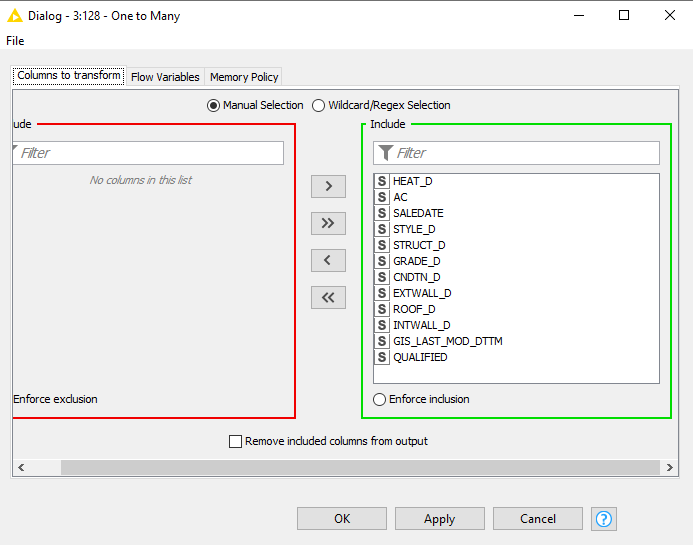
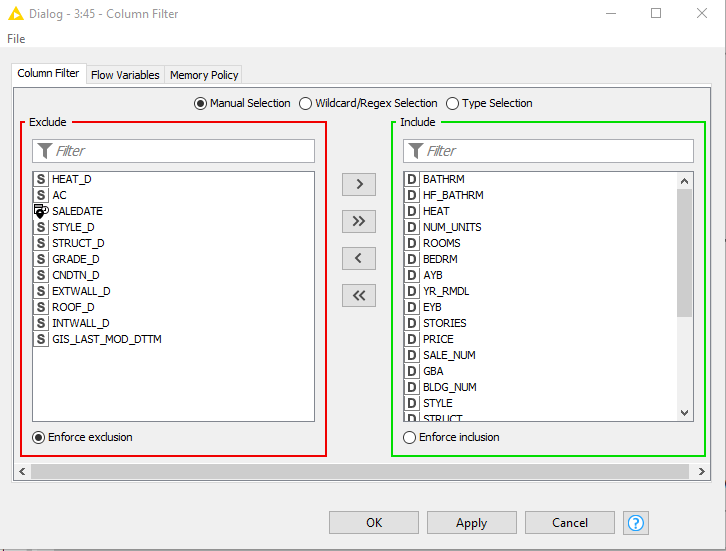


Figure – Removing Duplicate attributes and Binarization of Description Attributes

# Solving the Problem & Classifiers Used

This section of the report briefly explains the classification methods used and the parameters selected for each method.

## 3.A Decision Tree

A relatively simple classification technique, the decision tree creates a tree-like flowchart based on simple decision nodes extrapolated from the dataset. These nodes may be further divided into new nodes or ‘branches’.

The Decision Tree method was used as a quick preliminary model to further understand the dataset but due to the model’s difficulty in handling continuous variables (e.g. the PRICE attribute) and the superior performance of other models, this method was not seriously considered for our final classifier model. The default settings were used in Knime

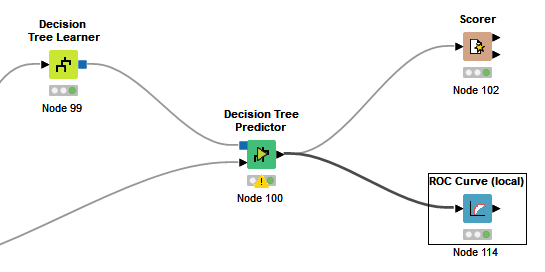


Figure - Decision Tree model

## 3.B Tree Ensemble

Expanding upon the Decision Tree model, the Tree ensemble method combines the results of several independent decision trees together (otherwise known as bagging) to produce a better predictive model.

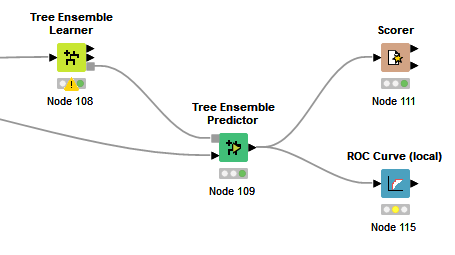


Figure - Tree Ensemble model

## 3.C Random Forest

A further extension on our initial Decision Tree model, the Random Forest method combines several decision trees together much like the Tree Ensemble method, to produce better predictive results. The key difference of the Random Forest method over the Tree Ensemble method is that in addition to ‘bagging’ random subsets of the data, this method uses a random selection of features for each branch split.

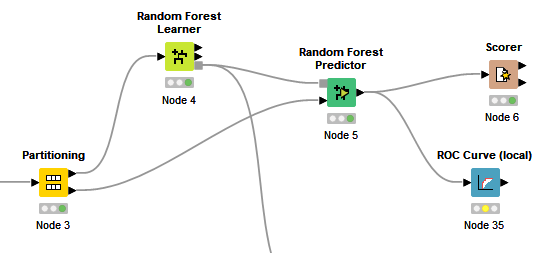


Figure - Random Forest model

## 3.D K-Nearest Neighbour

Another simple but well understood model, the K-Nearest Neighbours (KNN) algorithm classifies a data point based on its proximity to similar or neighbouring data records.

In the setup for our model, the k-value which represents the number of neighbours considered in the algorithm was set to 4.

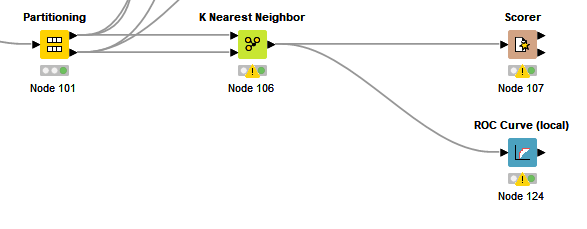


Figure - KNN model

## 3.E Support Vector Machine

Support Vector Machine (SVM) is an example of a Kernel method that attempts to find a formula for a hyperplane which separates positive and negative values to identify values in separate classes.

The SVM model proved to be the most difficult to setup correctly and the most time-consuming model to train. The Radical Basis Function (RBF) kernel was chosen after difficulty getting the Polynomial kernel to run, furthermore the sigma setting was increased to 0.5 to more quickly achieve our results.

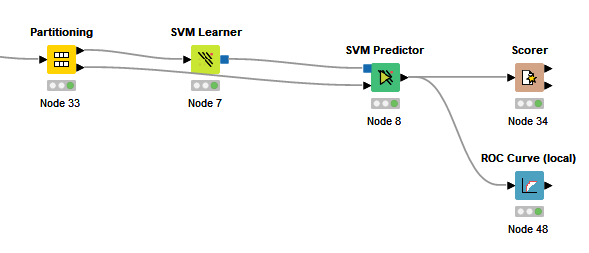


Figure - SVM model in Knime



Figure - SVM parameters selected

## 3.F Multi-layer Perceptron (MLP)

The Multi-layer Perceptron method is an example of neural network that explores the relationship between linear and non-linear data. The method consists of three layers: an input layer, hidden layer and output layer.

A backpropagation algorithm is the mechanism used to iteratively train up the MLP model, the parameters are adjusted after each iteration to minimise the error vector.

In our model we have set the number of hidden layers to 5, max iterations as 100 and the hidden neurons per layer as 20.

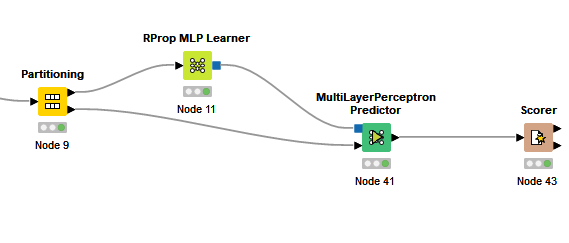


Figure - MLP model in Knime

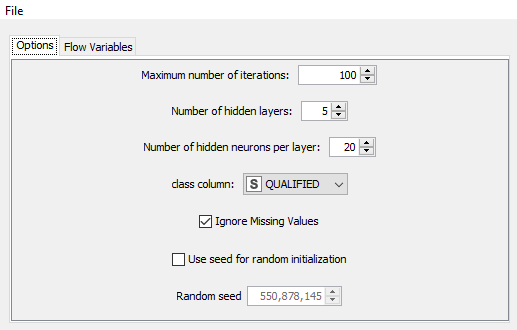


Figure - MLP Parameters

# Evaluating Classifiers

## 4.A Comparison of Classifier Models

Table 1 below gives a brief summary of the Scorer accuracy and Cohen’s Kappa achieved for each classifier model when comparing the partition Training and Testing data:

|  |  |  |
| --- | --- | --- |
| **Classifier Model** | **Scorer Accuracy (%)** | **Cohen’s Kappa (%)** |
| Decision Tree | 86.451 | 0.724 |
| Tree Ensemble | 89.728 | 0.794 |
| Random Forest | 90.188 | 0.803 |
| K-nearest Neighbour | 87.083 | 0.739 |
| SVM | 80.188 | 0.586 |
| MLP | 88.145 | 0.734 |

Table - Accuracy of the Classifier Models evaluated

## 4.B Final Classifier Model Selected

After comparing the various classifier models outlined in Section 3 of this report, the Random Forest Model gave us the best scorer results and was therefore selected as the most optimal classifier for use in categorizing the Unknown dataset.

In hindsight, this wasn’t surprising given the large number of missing values and diverse attribute types in the Training dataset.

Compared to the SVM and MLP classifiers, the Random Forest method was able to produce results substantially quicker. Random Forest was able to simply handle the missing data fields, while greater efforts were taken to pre-process the data for SVM/MLP with minimal improvement to the results achieved.

The final Knime Random Forest Model is presented in Appendix B.

# Reflection

In this analysis of a housing/property dataset, several classifier models were created to predict the ‘Qualified’ attribute of an unknown dataset with varying degrees of success.

We have outlined the steps taken to clean-up the dataset through pre-processing and feature selection, briefly explained the theory and mechanisms behind the classifier models examined, compared the results of training our models and finally given justification for the final classifier model chosen.

Although we found that all of the models were able to produce somewhat accurate results, we ultimately selected the Random Forest model as the most optimal classifier. Our final Kaggle result with this model was 0.89665 accuracy. As explained earlier, this was due to its ability to produce accurate results quickly and its flexibility in handling a very large number of attributes with differing data types.

In regards to this as a data mining learning exercise – it was great! The open-ended nature of the assessment meant there was a great degree of flexibility in both choosing which classifier models to use and which data analytics tool (Knime/R/Python). I trialled both Knime and Python through its Machine-learning Scikit-learn Library before deciding to use Knime due to various time constraints. However for future data analysis and class prediction problems, the Scikit-learn library appeared to be a more feature-rich tool.

I was surprised that almost all of the classifier models used in this analysis yielded quite reliable results, given more time and computational power I would have explored using the SVM and MLP models further.

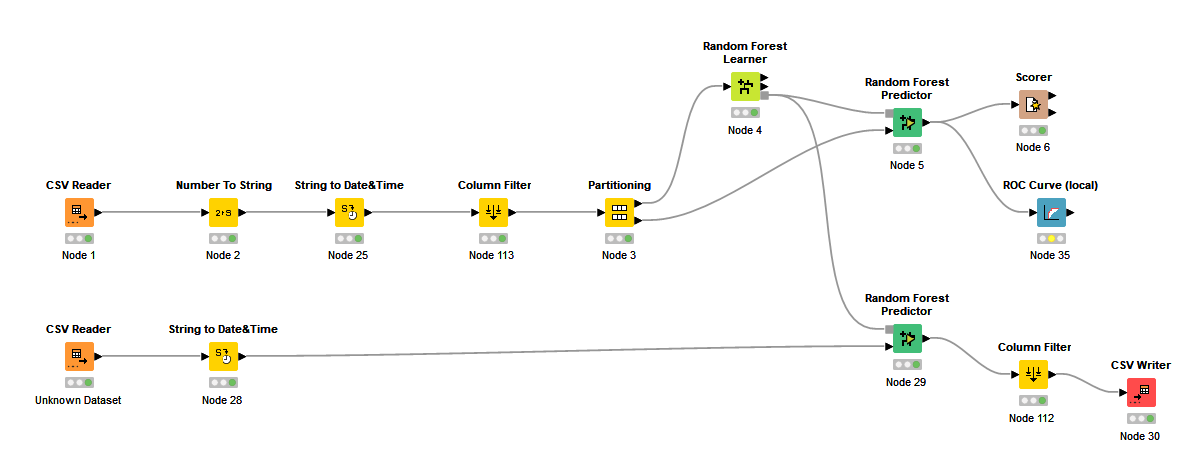
# Appendix

## Attribute Description

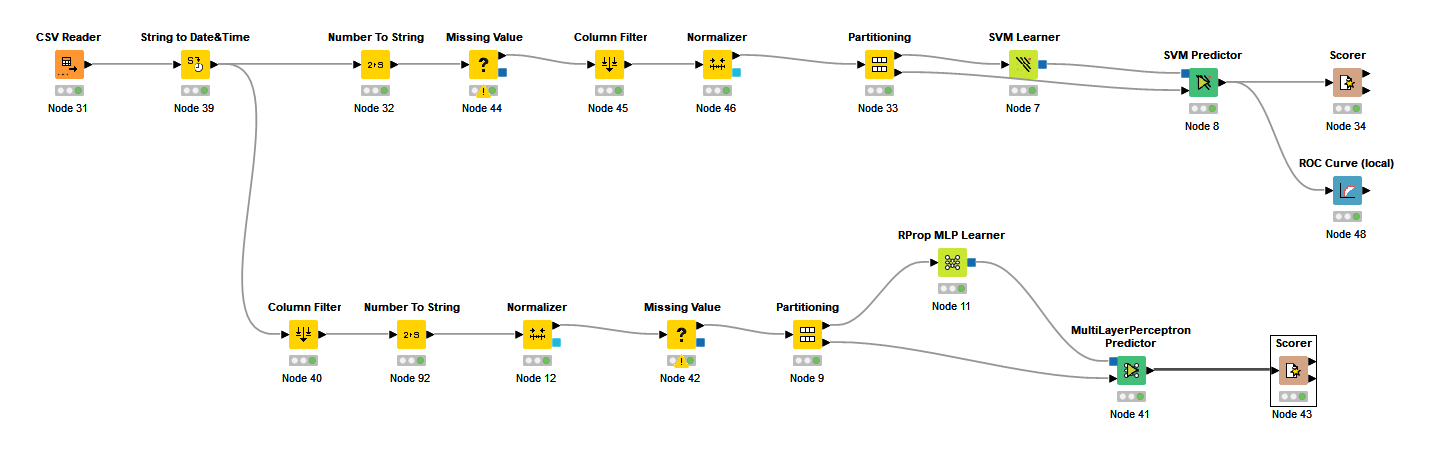
|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Attribute** | **Description** |
| 1 | ROW ID | Auto-generated Unique ID |
| 2 | SSL | Square, suffix, lot ID |
| 3 | BATHRM | Number of full bathrooms |
| 4 | HF\_BATHRM | Number of half bathrooms (no shower or tub) |
| 5 | HEAT | Heating code |
| 6 | HEAT\_D | Heating description |
| 7 | AC | Air conditioning (Y/N) |
| 8 | NUM\_UNITS | Number of units |
| 9 | ROOMS | Number of rooms |
| 10 | BEDRM | Number of bedrooms |
| 11 | AYB | The earliest time the main portion of the building was built. It is not affected by subsequent construction. |
| 12 | YR\_RMDL | Last year residence was remodeled |
| 13 | EYB | The year that an improvement was built that is most often more recent than actual year built. |
| 14 | STORIES | Stories |
| 15 | SALEDATE | Date of most recent sale |
| 16 | PRICE | Price of most recent sale |
| 17 | QUALIFIED | Qualified (Q), unqualified (U) |
| 18 | SALE\_NUM | Sale number |
| 19 | GBA | Gross building area in square feet |
| 20 | BLDG\_NUM | Building number. For parcels where multiple buildings exist, the main residence is assigned a value of 1 |
| 21 | STYLE | Style code |
| 22 | STYLE\_D | Style description |
| 23 | STRUCT | Structure code |
| 24 | STRUCT\_D | Structure description |
| 25 | GRADE | Grade code |
| 26 | GRADE\_D | Grade description |
| 27 | CNDTN | Condition code |
| 28 | CNDTN\_D | Condition description |
| 29 | EXTWALL | Exterior wall code |
| 30 | EXTWALL\_D | Exterior wall description |
| 31 | ROOF | Roof type code |
| 32 | ROOF\_D | Roof type description |
| 33 | INTWALL | Interior wall code |
| 34 | INTWALL\_D | Interior wall description |
| 35 | KITCHENS | Number of kitchens |
| 36 | FIREPLACES | Number of fireplaces |
| 37 | USECODE | Property use code |
| 38 | LANDAREA | Land area of property in square feet |
| 39 | GIS\_LAST\_MOD\_DTTM | Last modified datetime |

## Knime Diagram

Final Random Forest Model



MLP/SVM model:



KNN/Decision-Tree/Tree Ensemble Models:

