1. Objective

To build a decision tree to predict which user will buy services/products offered on the website, based on their characteristics and interactions exhibited.

Data understanding, data preparation, modelling and evaluation processes under the CRISP-DM (CRoss-InduStry Process for Data Mining) will be performed as described in this report.

2. Understand and Prepare Data

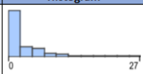
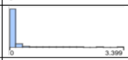
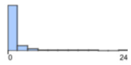
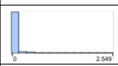
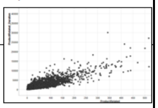
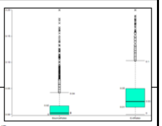
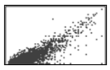
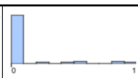
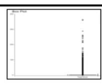
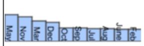
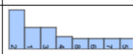
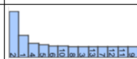
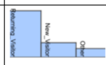
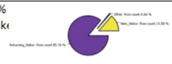
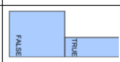
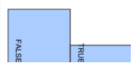
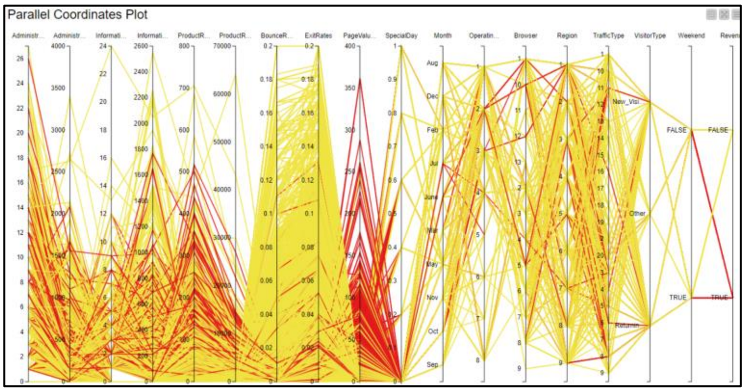
S/N	Variable Name/Label	Data Classification	Classes or values of the Variable	No. missings	Min	Max	Mean	Median	Std. deviation	Histogram	Observation / Distribution
1	Administrative	Discrete		0	0	27	2.32	1.00	3.33		47% (4,588 records) are '0' values. Another 11% (1,116 records) are with value '1'. In fact, the majority of the records (>96%) are with value '0' to '10' (out of max '27'). The distribution is positively skewed.
2	Administrative_Duration	Continuous		0	0	3,399	80.43	8.00	177.24		48% (4,704 records) are 0 values. When Administrative_Duration is > 0, the Administrative is also > 0. The distribution is positively skewed.
3	Informational	Discrete		0	0	24	0.50	0.00	1.26		78% (7,741 records) are '0' values, resulting in median at 0. The remaining 22% are with value '1' (841 records) to value '24' (1 record), except values '15', '17' to '23' (no record). The distribution is positively skewed.
4	Informational_Duration	Continuous		0	0	2,549	34.48	0.00	142.61		7,920 records (~80%) are 0 values. When Informational_Duration is > 0, the Informational is also > 0. The distribution is positively skewed.
5	ProductRelated	Discrete		0	0	705	32.00	18.00	45.27		Almost 100% of records are non zero, with majority of counts (>95%) contributed by 100 rows (out of ~295 rowid). The median is 18 (< mean of 32). The distribution is positively skew. 
6	ProductRelated_Duration	Continuous		0	0	63,974	1,204.49	602.54	1,961.27		Almost 100% of records are non zero. When ProductRelated_Duration is > 0, the Product_Related is also > 0. The distribution is positively skewed. Positive relation between ProductRelated & ProductRelated_Duration
7	BounceRates	Continuous		0	0	0.20	0.02	0.00	0.05		4,381 records (~44%) are 0 values, resulting in median at 0.003 and many outliers, as shown in first column of box plot. The distribution is positively skewed. 
8	ExitRates	Continuous		0	0	0.20	0.04	0.03	0.05		99% of records are non zero values. The IQR are from 0.01 to 0.05, with median 0.03 as shown in second column of the box plot. The distribution is positively skewed. With the scatter mattrix, ExitRates & BounceRates has a positive correlation. 
9	PageValues	Continuous		0	0	361.76	5.75	0.00	18.35		2,172 records (~22%) are with 0 values. The distribution of data is centralised around 0, with many outliers, as shown in the box plot. The distribution is positively skewed. 
10	SpecialDay	Continuous		0	0	1.00	0.06	0.00	0.20		8,852 records (~90%) are with 0 values. The distribution is positively skewed.
11	Month	Nominal		0							The highest is May with ~2,700 records, follow by Nov, Mar, Dec and rest of the months. There is no record for Jan and Apr.
12	Operating Systems	Nominal	Coded as '1' to '8', where each value represent an operating system type	0							The highest is with Operating System '2', follow by '1' and '3'. The record count for these 3 operating systems added up to ~95% of the total counts.
13	Browser	Nominal	Coded as '1' to '13', where each value represent a browser type	0							The highest is with Browser '2' which add upto ~65% of the total counts, followed by Browser '1' & '4' which add up to another ~26%.
14	Region	Nominal	Coded as '1' to '9', where each value represent a region	0							The highest is from region '1' (39%), follow by region '3' (19%), etc.
15	TrafficType	Nominal	Code as '1' to '20', where each value represent a traffic type	0							The highest is via traffic type '2' (32%), follow by type '1' and '3' (37%)
16	VisitorType	Nominal	Either 'Returning_Visitor', 'New_Visitor', or 'Other'	0							The 'Returning_visitor'make up ~86% (in purple), while 'New_visitors' make up slighly less than 14% (in yellow) 
17	Weekend	Binary (Nominal)	Either 'True' or 'False'	0							77% of the records are not from weekend, and 23% are from weekend.
18	Revenue	Binary (Nominal)	Either 'True' or 'False'	0							The proportion of the 'False' class is ~85% (in red), against the 'True' class of 15% (in green). As the target for prediction, this unbalanced ratio will likely result in biasness of the model. 

Table 1: Metadata and statistics consolidated from KNIME

Summary:

i. Target variable has unbalanced proportion of 'False' (85%) and 'True' (15%). The high 85% of users with no purchases might cause biasness to the model.
ii. All quantitative variables are positively skewed, with Median < Mean.
iii. There are outliers for PageValues, BounceRates and ExitRates attributes.

Parallel Coordinates Plot to better understand the relation between the target Revenue variable with the rest of the attributes were generated and the observations documented below:



Observations:
Users making purchases (Revenue = 'True' as red lines) have higher PageValues, whereas users not making purchases have higher BounceRates and ExitRates (as yellow lines).

Diagram 1: Exploring training data, and observations noted

The following data cleansing, transformation and integrity checks were carried out, prior to building the model:

Process/Activity	Action/ Reason for no action taken
Data Cleansing – Formatting	Attributes OperatingSystem, Browser, Region and Traffic Type were converted from integer to string as they are code types.
Data Cleansing – Missing values, inconsistencies	There are no missing values, and the code values has no spelling errors nor differ in format. Thus, no changes required.
Data Cleansing – Abnormalities, Imputation	Although there are outliers for PageValues, BounceRates, ExitRates, no imputations were carried out. The outliers can help in understanding data and identifying classifications.
Data Transformation – Discretization, Rescaling, etc	Variables such as BounceRates, ExitRates are already on the similar scale. No discretization performed on continuous values to avoid loss of information.
Data Quality Checks	The binary variables only contain the 'True' and 'False' values, variables on duration align with the labels, etc. The data passed the integrity checks.

Table 2: Data preparation activities carried out

3. Data Mining – Building the Decision Tree

Decision Tree was chosen for the below reasons:

(i) It can handle both continuous and categorical variables provided in the data set
(ii) It provides clear indication on fields that are important for prediction and is easily understood

Below are the KNIME workflows constructed for Training with cross validation and hyper parameter tuning, as well as the final testing using the tunned hyperparameters:

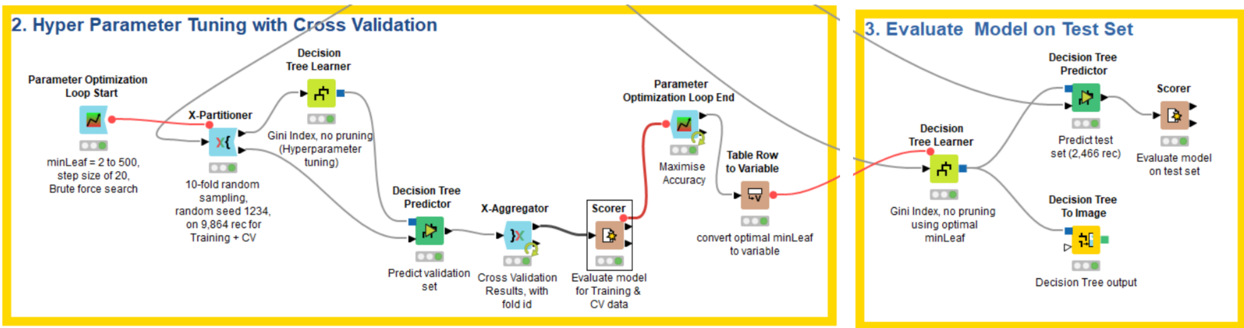


Diagram 2: Workflows on training and cross validation (left), final testing using optimal parameter (right)

The main configurations specified in the training with cross validation workflow were summarised below:

<table>
<tr><th>Node & Configurations</th><th>Purpose / Remarks</th></tr>
<tr><td>

Parameter Optimization Loop Start:

- Add parameter minLeaf, start 2 to 500 with step 20
- Brute force Search

Parameter Optimization Loop End:

- Maximise Accuracy

</td><td>

The optimisation is configured to Brute Force search strategy to loop and find all the possible parameter combinations, within the intervals and step sizes, for the best minLeaf hyperparameter with maximum accuracy.

The initial interval specified for the minimum number of records per node (minLeaf) is between 2 to 500, at step of 20 (24 iterations). KNIME will advise an optimal minLeaf value at the loop end node.

Hyperparameter minLeaf refers to the minimum number of records at least required in each node. If the number of records is > minLeaf, the tree will stop 'growing' (a pre-pruning method).

</td></tr>
<tr><td>

X-Partitioner:

- Connect to get 9,864 records for training + cross validation
- 10 fold
- Random sampling and Random seed '1234'

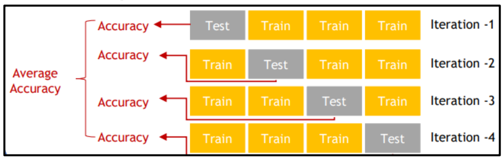
X- Aggregation:

- Target & predict Revenue column
- Check to add column on fold id

</td><td>

All nodes between the X-Partitioner and X-Aggregator nodes are iterated for cross validations.

As 10-fold is configured, the 9,864 records are partitioned into 10 sets, where 9 sets are used to train the model and 1 set is used to validate the trained model for each iteration, as illustrated in the below diagram:



With the above, each record will appear in both the training and validation test sets.

Below are the results collected by X-Aggregator for each of the 10 iterations:

<table>
<tr><th>Row ID</th><th>D Error in %</th><th>I Size of Test Set</th></tr>
<tr><td>fold 0</td><td>7.497</td><td>987</td></tr>
<tr><td>fold 1</td><td>10.649</td><td>986</td></tr>
<tr><td>fold 2</td><td>9.929</td><td>987</td></tr>
<tr><td>fold 3</td><td>12.17</td><td>986</td></tr>
<tr><td>fold 4</td><td>9.939</td><td>986</td></tr>
<tr><td>fold 5</td><td>14.488</td><td>987</td></tr>
<tr><td>fold 6</td><td>11.765</td><td>986</td></tr>
<tr><td>fold 7</td><td>10.436</td><td>987</td></tr>
<tr><td>fold 8</td><td>11.866</td><td>986</td></tr>
<tr><td>fold 9</td><td>10.548</td><td>986</td></tr>
</table>

</td></tr>
<tr><td>

Decision Tree Learner:

- Gini index
- With reduced error pruning

</td><td>

Training and validating the Decision Tree Model, to predict the nominal target variable Revenue.

</td></tr>
</table>

Node & Configurations	Purpose / Remarks
• Flow variable, minLeaf as minNumberRecordsPerNode **Decision Tree Predictor**	

Table 3: Configurations performed in the KNIME's training with cross validation workflow

Result of hyperparameter tuning:
The optimised hyper parameter on minimum number of records per node is 62, with objective value of 0.901.

Using the optimal hyperparameter, the trained model was fed with the final test set for prediction. Below are the results of the 2 Scorer nodes, for the training + cross validation, and final test sets:

<table>
<tr><th></th><th>For Training + Cross Validation</th><th>For Final Testing</th></tr>
<tr><td>Confusion Matrix</td><td>
<table>
<tr><td colspan="2"># of records =9,864</td><td colspan="2">Predicted</td></tr>
<tr><td></td><td></td><td>TRUE</td><td>FALSE</td></tr>
<tr><td>Actual</td><td>TRUE</td><td>962</td><td>554</td></tr>
<tr><td></td><td>FALSE</td><td>524</td><td>7,824</td></tr>
</table>
</td><td>
<table>
<tr><td colspan="2"># of records =2,466</td><td colspan="2">Predicted</td></tr>
<tr><td></td><td></td><td>TRUE</td><td>FALSE</td></tr>
<tr><td>Actual</td><td>TRUE</td><td>232</td><td>160</td></tr>
<tr><td></td><td>FALSE</td><td>78</td><td>1,996</td></tr>
</table>
</td></tr>
<tr><td>Accuracy</td><td>0.891</td><td>0.903</td></tr>
<tr><td>Sensitivity, Specificity, Precision</td><td>
<table>
<tr><td></td><td>Sensitivity</td><td>Specificity</td><td>Precision</td></tr>
<tr><td>TRUE</td><td>0.635</td><td>0.937</td><td>0.647</td></tr>
<tr><td>FALSE</td><td>0.937</td><td>0.635</td><td>0.934</td></tr>
</table>
</td><td>
<table>
<tr><td></td><td>Sensitivity</td><td>Specificity</td><td>Precision</td></tr>
<tr><td>TRUE</td><td>0.592</td><td>0.962</td><td>0.748</td></tr>
<tr><td>FALSE</td><td>0.962</td><td>0.592</td><td>0.926</td></tr>
</table>
</td></tr>
</table>

Table 4: Metrics from the KNIME's Score Nodes, for the training + cross validation, and final test sets

4. Model Evaluation

(i) Accuracy or Sensitivity

Accuracy is expressed as $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FN+FP}}$, whereas Sensitivity (or Recall) is $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$.

Accuracy metrics include both the measurements on model correctly predicted the positive scenarios (TP), and the negative scenarios (TN). High Accuracy can be due to very high TP, or very high TN or both.

Sensitivity measures the model correctly predicted the positive scenarios (TP), against all actual positive scenarios. High Sensitivity can only be high due to high TP.

As the objective of this work is to predict which user will purchase (which is on positive TP scenarios), Sensitivity is the preferred performance metric.

Sensitivity is preferred especially when the training data has unbalanced proportion of 'False' (or negative) and 'True' (or positive) scenarios, as Accuracy can be biased due to the dominating negative scenarios. As summarised under section 1, the proportion of 'False' and 'True' scenarios is 85% to 15% which is significantly unbalanced. This further explained Sensitivity or Recall metric is preferred.

(ii) Accuracy of dummy model with all records predicted as 'False'
The accuracy of the dummy model is 0.874.

(iii) Comparison and deployment of the trained decision tree model and the dummy model

The Accuracy for the trained decision tree model on the final test set is 0.903, as shown in Table 4. This Accuracy (valued 0.903) is higher than that of the dummy model (valued 0.874). In addition, the Sensitivity for the decision tree model is 0.592. Considering both the Accuracy and Sensitivity, it is worthwhile to deploy the decision tree model.

(iv) Improving Sensitivity with balancing the 'False' and 'True' classes of the Revenue Target variable

One way to improve the Sensitivity is to use the 'Equal Size Sampling' node in KNIME to balance the 'False' classes, with the minimum number of 'True' classes. The decision on performing balancing is to be made with considerations on the objective and which metric is more important.

(v) Characterise users who have a positive purchase intent (REVENUE=TRUE)

Based on the decision tree model, it characterised the users with positive purchase intent as having PageValues > 0.9448 and BounceRates < 8.11E-5. This is aligned with observation in Diagram 1 that "Users made purchases have higher PageValues, whereas users not making purchases have higher BounceRates and ExitRates".

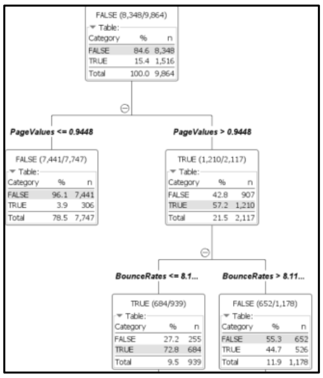


Diagram 3: Decision Tree Model

(vi) Cost Matrix

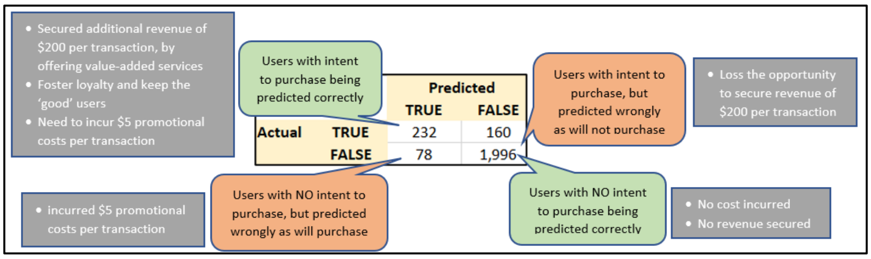


Diagram 4: Cost Matrix for the trained decision tree

Reference to Diagram 4, the cost matrix for the trained decision tree model = (232* ($200-$5)) - (78*$5) – (160*$200) + 0 = + $12,850.

If the decision tree model is deployed, it will benefit with a value of +

<< End of Report >>