1. Data Understanding and Pre-processing:

1.1 Problem Statement:

An automobile company has plans to enter new markets with its existing products. After intensive market research, they deduced that the behaviour of the new market is like their existing market. In their existing market, the sales team has classified all customers into 4 segments (A, B, C, D). Then, they performed segmented outreach and communication for a different segment of customers. This strategy has worked exceptionally well for them. They plan to use the same strategy for the new markets. So, we are building two classification models to predict customer segmentation.

1.2 Data Collection:

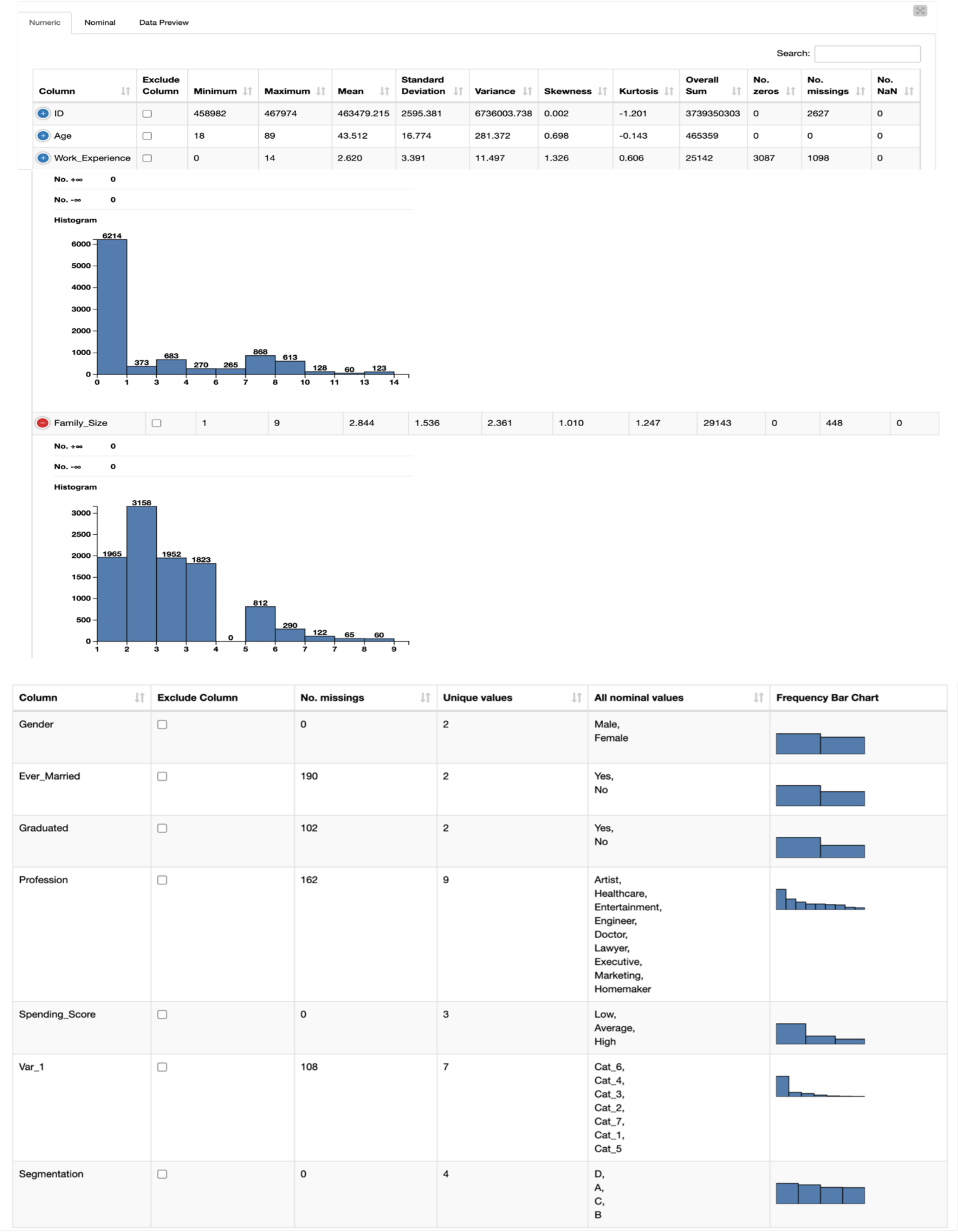
The Data set provided for this study is "customers.csv" from the Analytics Vidhya hackathon. It contains 10965 rows of data with 11 features. The individual feature descriptions are explained in Table 1 as follows.

Feature	Description
ID	Unique ID
Gender	Gender of the customer
Ever_Married	Marital status of the customer
Age	Age of the customer
Graduated	Is the customer a graduate?
Profession	Profession of the customer
Work_Experience	Work Experience in years
Spending_Score	Spending score of the customer
Family_Size	Number of family members for the customer (including the customer)
Var_1	Anonymized Category for the customer
Segmentation	(Target) Customer Segment of the customer

Table 1 – Feature names with Descriptions

1.3 Data Loading & Exploration:

In this study, we are using KNIME version 4.7.4 software for Data mining and understanding. The CSV file containing the customer data is loaded into KNIME using the "CSV Reader" node. The output table of this node has 10695 rows of records and 11 columns. The output table is next fed to the "Data Explorer" node to explore and analyze the dataset with respect to its structure, distribution, and characteristics in a graphical user interface [5]. The interactive table view of this node is shown in Figure 1. We can observe that the columns ID, Work_Experience, Family_Size, Ever_Married, Graduated, Profession, and Var_1 had the null values. In addition, we can see the distribution of each feature in the Histogram. We must treat these null values to prepare the data for applying the classification algorithms. The ways of handling these null values are discussed in the next step.



Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum	No. zeros	No. missings	No. NaN
ID	□	458982	467974	463479.215	2595.381	6736003.738	0.002	-1.201	3739350303	0	2627	0
Age	□	18	89	43.512	16.774	281.372	0.698	-0.143	465359	0	0	0
Work_Experience	□	0	14	2.620	3.391	11.497	1.326	0.606	25142	3087	1098	0

No. +∞ 0

No. -∞ 0

Histogram

Column	Exclude Column	Minimum	Maximum	Mean	Standard Deviation	Variance	Skewness	Kurtosis	Overall Sum	No. zeros	No. missings	No. NaN
Family_Size	□	1	9	2.844	1.536	2.361	1.010	1.247	29143	0	448	0

No. +∞ 0

No. -∞ 0

Histogram

Column	Exclude Column	No. missings	Unique values	All nominal values	Frequency Bar Chart
Gender	□	0	2	Male, Female	
Ever_Married	□	190	2	Yes, No	
Graduated	□	102	2	Yes, No	
Profession	□	162	9	Artist, Healthcare, Entertainment, Engineer, Doctor, Lawyer, Executive, Marketing, Homemaker	
Spending_Score	□	0	3	Low, Average, High	
Var_1	□	108	7	Cat_6, Cat_4, Cat_3, Cat_2, Cat_7, Cat_1, Cat_5	
Segmentation	□	0	4	D, A, C, B	

Figure 1 – Graphical representation of features in the Data Exploration node.

1.4 Data Preprocessing:

The sequence of steps involved in Data Preprocessing is shown in Figure 2.

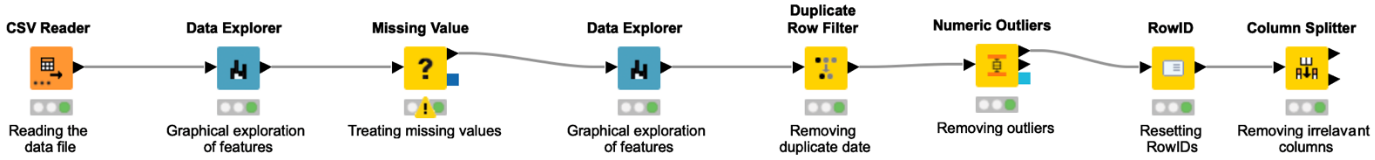


Figure 2 – Workflow of Data Preprocessing

The Data preprocessing consists of the following steps:

- **Handling Missing Values:**
  In this step, we replaced the missing values in the integer columns such as the 'Age' column with the rounded mean, 'Work_Experience' and 'Family_Size' with Mode. For string data columns, we replaced the null values with fixed value - Unknown in the 'ID' and 'Profession' columns. Removed the null value rows in the 'Ever_Married', 'Graduated', and 'Var_1' columns [3]. The output table of the 'Missing Value' node consists of 8243 rows of records and 11 features.

- **Removing Duplicates:**
  The Output table from the previous node is fed to the 'Duplicate Row Filter' node. It deleted all the records with duplicates. The filtered table of this node consists of 8222 records and 11 features.

- **Removing Outliers:**
  The filtered table from the last node is fed to the "Numeric Outliers" node. In this node, the features 'Age', 'Work_Experience', and 'Family_Size' were checked against the outliers by an R_4 estimate using a 1.5 IQR multiplier with 'Remove outlier rows' treatment. The output of this node consists of 6842 records and 11 features.

- **Removing irrelevant features**
  By using the 'Column Splitter' node, we removed irrelevant features like 'ID', 'Ever_Married', and 'Graduated' columns from the dataset in order to reduce the dimensionality and complexity. The output of this node consists of 6842 records and 8 features for furtherance.

1.5 Data Manipulation & Visualization:

In this section, we analyze the data using Statistical and Visualization nodes.

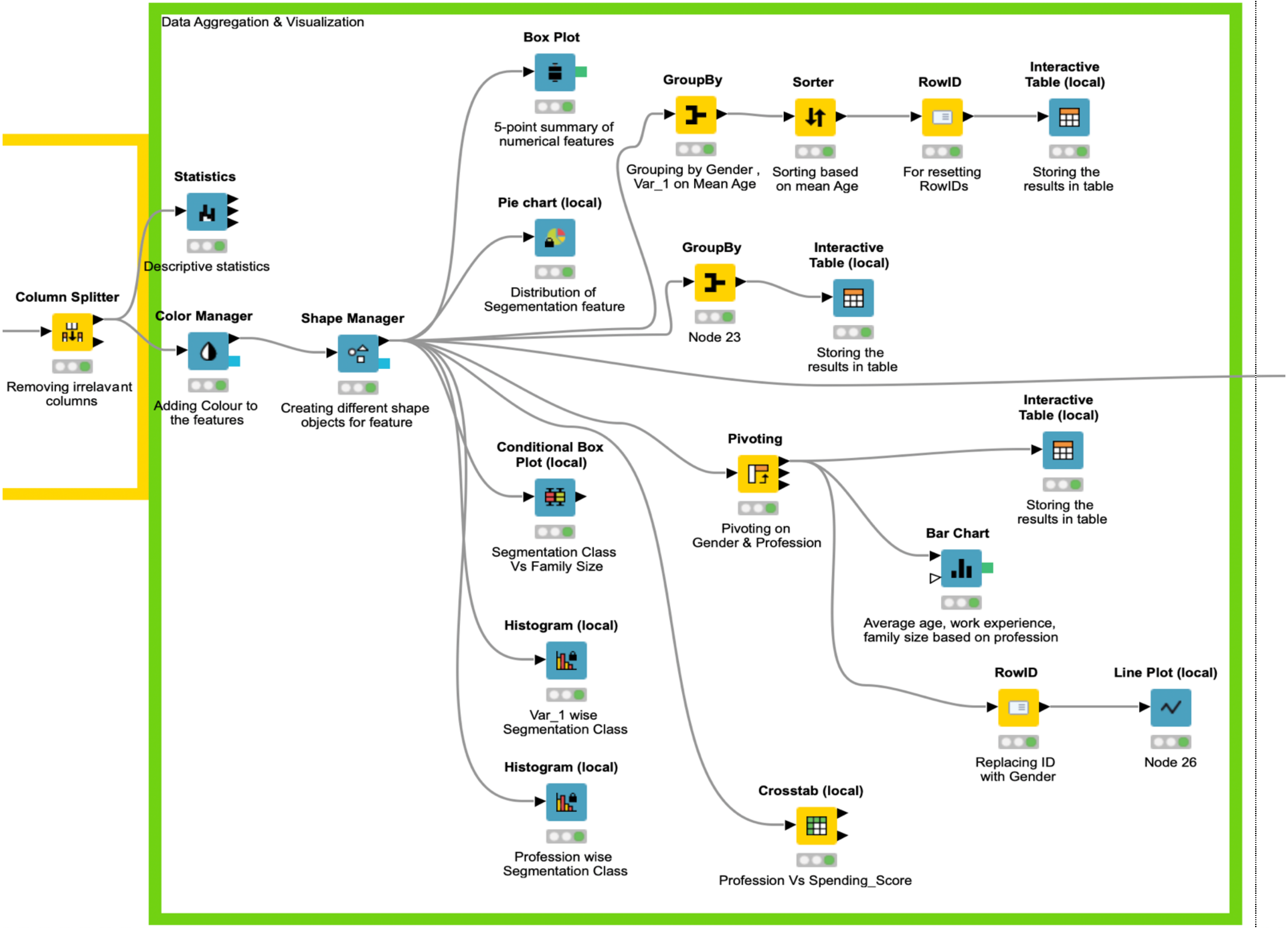

Data Aggregation & Visualization
Column Splitter
Removing irrelavant columns
Statistics
Descriptive statistics
Color Manager
Adding Colour to the features
Shape Manager
Creating different shape objects for feature
Box Plot
5-point summary of numerical features
Pie chart (local)
Distribution of Segementation feature
GroupBy
Grouping by Gender , Var_1 on Mean Age
Sorter
Sorting based on mean Age
RowID
For resetting RowIDs
Interactive Table (local)
Storing the results in table
GroupBy
Node 23
Interactive Table (local)
Storing the results in table
Conditional Box Plot (local)
Segmentation Class Vs Family Size
Pivoting
Pivoting on Gender & Profession
Interactive Table (local)
Storing the results in table
Bar Chart
Average age, work experience, family size based on profession
Histogram (local)
Var_1 wise Segmentation Class
RowID
Replacing ID with Gender
Line Plot (local)
Node 26
Histogram (local)
Profession wise Segmentation Class
Crosstab (local)
Profession Vs Spending_Score

Figure 3 – Workflow of Data Manipulation and Visualization

- **Statistics:**
  The Output table from the 'Column Splitter' node is fed to the 'Statistics' Node to check the to compute basic statistical measures and characteristics of a dataset. For numeric data, it will provide 'Min', 'Mean', 'Median', 'Max', 'Std. Dev.', 'Skewness', 'Kurtosis', 'No. Missing', Distribution of data in 'Histogram'. For Nominal data, it will provide 'No. Missing', Distribution in 'Histogram'.

- **'Color Manager' & 'Shape Manager' Nodes:**
  The output of the 'Column Splitter' node is fed to the 'Color Manager' and 'Shape Manager' nodes to add colours and shapes to the records as per the Segmentation feature to distinguish between various segments (A, B, C, D). Output from these nodes is fed to various plotting nodes for visual representation of data.

- **Box Plot:**
  A 'Box Plot' node is used for a five-number summary of the numeric columns in the dataset. The A summary consists of five values: the most extreme values in the data set (the

maximum and minimum values), the lower and upper quartiles, and the median. The Box plot of the features 'Age', 'Work_Experience' and 'Family_Size' are as follows.

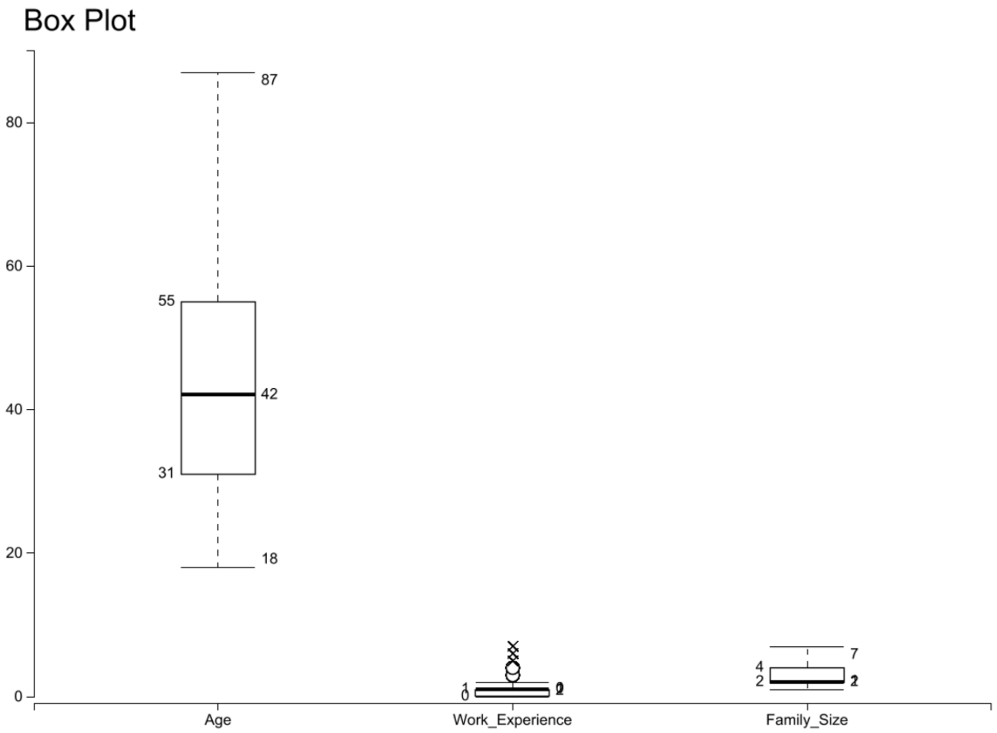


Figure 4 – Box plots on Age, Work_Experience, Family_Size

- **Pie Chart:**
  The "Pie Chart" node is used to visualize and explore the distribution of categorical data by creating a pie chart. In our study, we created a 'Pie Chart' on the Segmentation feature using the count of each variable. From the chart, we can see an overview of the distribution of each categorical data element is the same. Hence, our data is balanced.

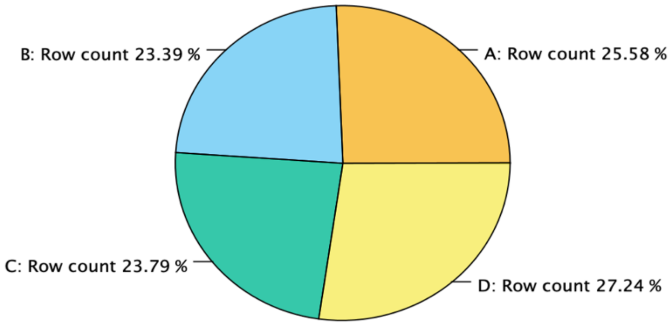


Figure 5 – Pie chart on Distribution of Segmentation column classes

- **Conditional Box Plot:**
  The ‘Conditional Box Plot’ node is used to create box plots based on the conditional grouping of data. In this study, we used this to plot a segmentation feature class-wise five-number summary against the ‘Family_Size’ feature.

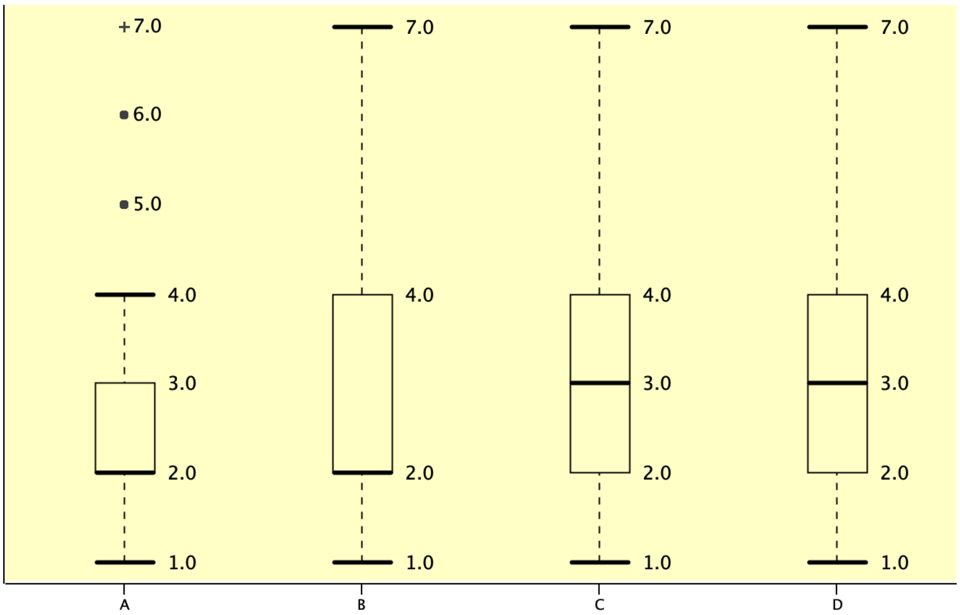


Figure 6- Conditional Box plot on Segmentation Class w.r.t Family_Size

- **Histogram:**
  The "Histogram" node is used to create histograms, which are graphical representations of the distribution of a dataset. Histograms display the frequencies of values in different intervals, providing insights into the data's underlying patterns and characteristics. We used this chart to plot Var_1 distribution with respect to the Segmentation feature class and Profession-wise Segmentation feature class distribution based on the count of rows.

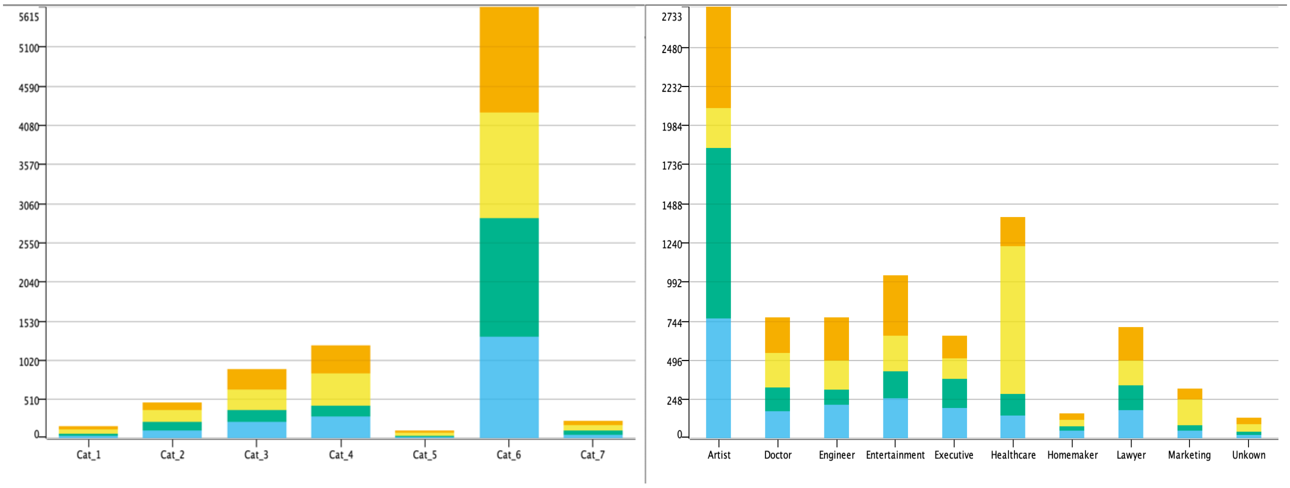
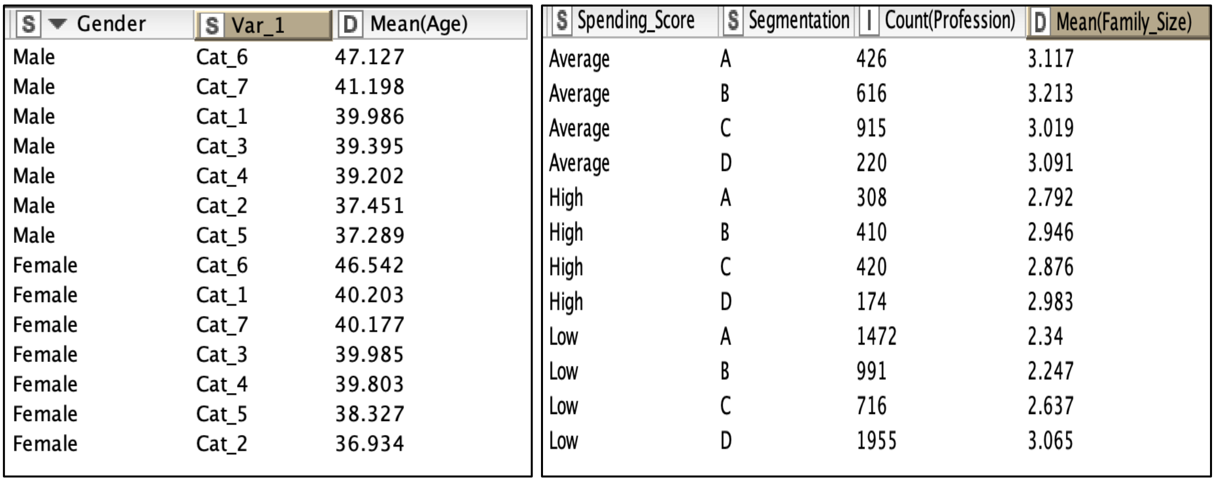


Figure 7 – Histogram on Var_1 and Profession features w.r.t Segmentation class

- **‘GroupBy’ & ‘Sorter’ Nodes:**
  When organizing rows in a dataset according to one or more columns, the "GroupBy" node is utilized to process the grouped data using computations or aggregates. This node is very helpful for combining data activities such as summarizing and computing statistics for each group. We used the ‘GroupBy’ node to find the mean age of each category according to Gender and arranged the values in Descending order of Age using the ‘Sorter’ node. We similarly grouped the Spending Score, and Segmentation on count of values and average family size.

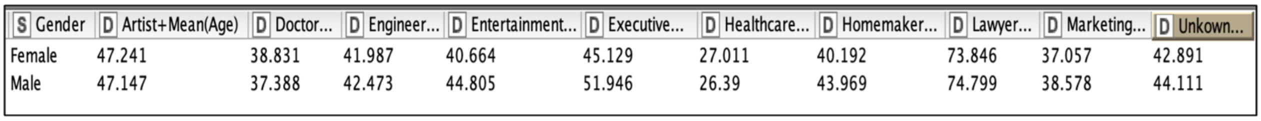


S ▼ Gender	S Var_1	D Mean(Age)
Male	Cat_6	47.127
Male	Cat_7	41.198
Male	Cat_1	39.986
Male	Cat_3	39.395
Male	Cat_4	39.202
Male	Cat_2	37.451
Male	Cat_5	37.289
Female	Cat_6	46.542
Female	Cat_1	40.203
Female	Cat_7	40.177
Female	Cat_3	39.985
Female	Cat_4	39.803
Female	Cat_5	38.327
Female	Cat_2	36.934

S Spending_Score	S Segmentation	I Count(Profession)	D Mean(Family_Size)
Average	A	426	3.117
Average	B	616	3.213
Average	C	915	3.019
Average	D	220	3.091
High	A	308	2.792
High	B	410	2.946
High	C	420	2.876
High	D	174	2.983
Low	A	1472	2.34
Low	B	991	2.247
Low	C	716	2.637
Low	D	1955	3.065

Table 2 – Groupby Tables on Gender vs Var_1 and Spending_Score vs Segmentation

- **Pivoting:**
  The "Pivoting" node is frequently used to restructure data and is helpful in a variety of situations, including gathering data and getting it ready for analysis. It alters a table's layout in order to change data. In this study, we pivoted the Average age data on Gender and Profession. The result of the Pivoting table is shown below. Plotting Box plot and line plot on the result table.



S Gender	D Artist+Mean(Age)	D Doctor...	D Engineer...	D Entertainment...	D Executive...	D Healthcare...	D Homemaker...	D Lawyer...	D Marketing...	D Unkown...
Female	47.241	38.831	41.987	40.664	45.129	27.011	40.192	73.846	37.057	42.891
Male	47.147	37.388	42.473	44.805	51.946	26.39	43.969	74.799	38.578	44.111

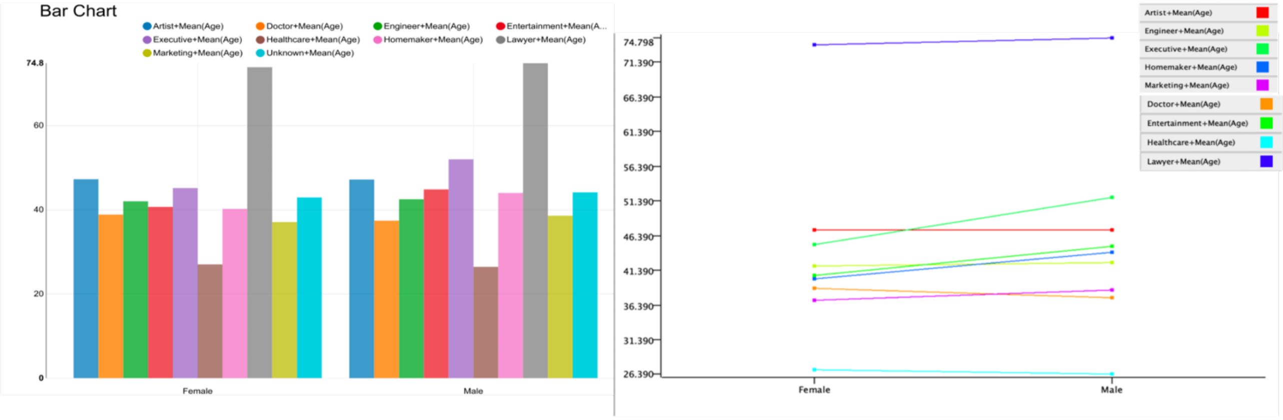
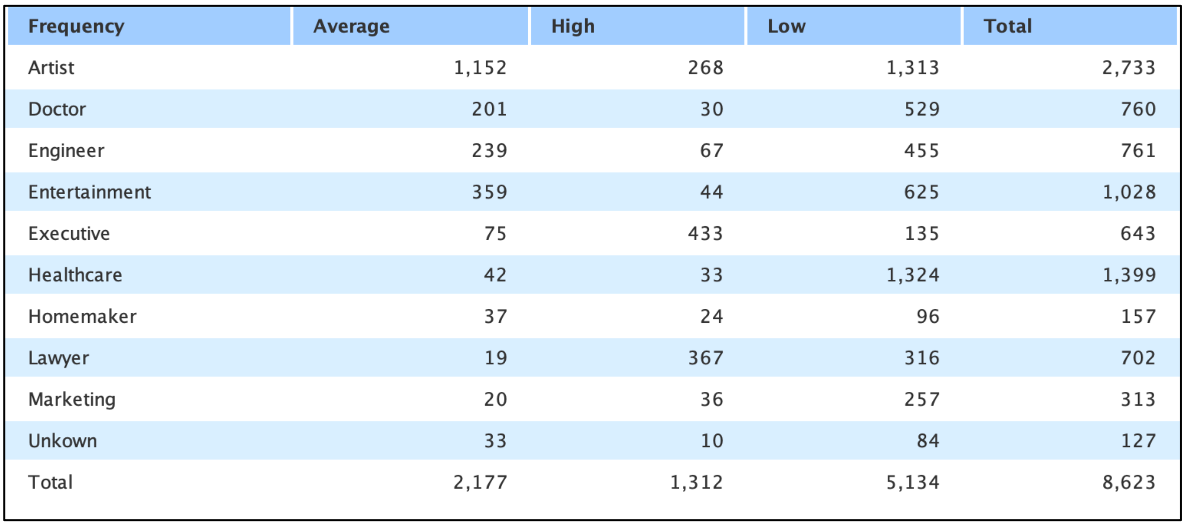


Figure 8 – Histogram and Line Plot on Gender w.r.t avg Age and Profession

- **CrossTab:**
  Cross-tabulations and contingency tables are made with the "Crosstab" node. A contingency table is a table that illustrates the connections between variables as well as the frequency distribution of those variables. In particular, the "Crosstab" node helps examine the relationship between two category variables. Here, we used to create a contingency table between the 'Profession' and 'Spending_Score' features.



Frequency	Average	High	Low	Total
Artist	1,152	268	1,313	2,733
Doctor	201	30	529	760
Engineer	239	67	455	761
Entertainment	359	44	625	1,028
Executive	75	433	135	643
Healthcare	42	33	1,324	1,399
Homemaker	37	24	96	157
Lawyer	19	367	316	702
Marketing	20	36	257	313
Unkown	33	10	84	127
Total	2,177	1,312	5,134	8,623

Table 3- CrossTab on Profession and Spending_Scores

1.6 Feature Encoding and Scaling:

The output table from the 'Shape Manager' is input for the Feature Encoding and Scaling section. The workflow of the Feature Encoding and Scaling section is as follows.

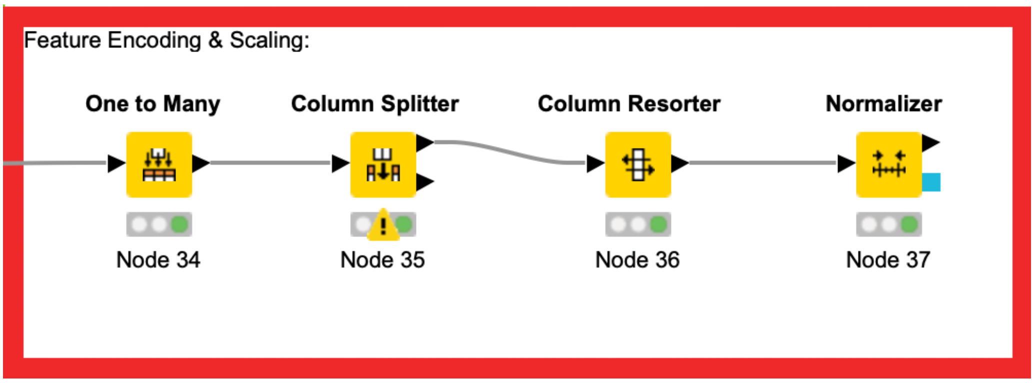


Figure 9- Feature Encoding and Scaling workflow

We use the following nodes in this section.

- **One to Many:**
  This node is used for applying the one-hot encoding on categorical data columns. One-hot encoding creates binary columns for each category, representing the presence or absence of that category. The output table of this node consists of 6842 records and 30 columns.

- **Column Splitter:**
  This node is used to remove the Original nominal columns ('Gender', 'Profession', 'Spending_Score', and 'Var_1') which are One-hot encoded in the previous node. The output of this node contains 6842 records and 26 columns.

- **Column Resorter:**
  This node is used to re-arrange the columns with the target column at the end i.e., the Segmentation column at the last.

- **Normalizer:**
  Scaling and normalising numerical characteristics in a dataset is accomplished with the "Normalizer" node. In order to guarantee that all features have similar scales and avoid particular characteristics from predominating over others in models that are sensitive to the scale of the input features, normalisation is a typical pre-processing step in machine learning. The output of this node contains 6842 records and 26 columns.

- **CSV Writer:**
  By using this node, we can save the feature encoded and scaling data into a CSV file relative to the current working area directory.

- **Correlation:**
  This node computes and examines the correlation between a dataset's numerical variables. The statistical link between two variables is measured by correlation, which also shows the direction and intensity of the linear relationship between them.

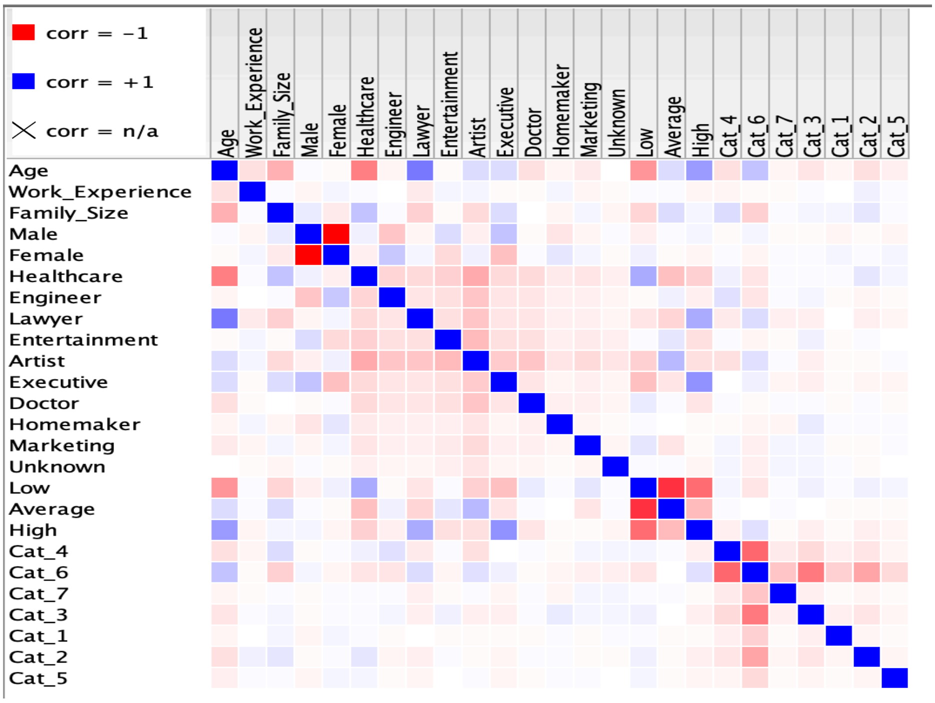


Figure 10 – Correlation Diagram of Features

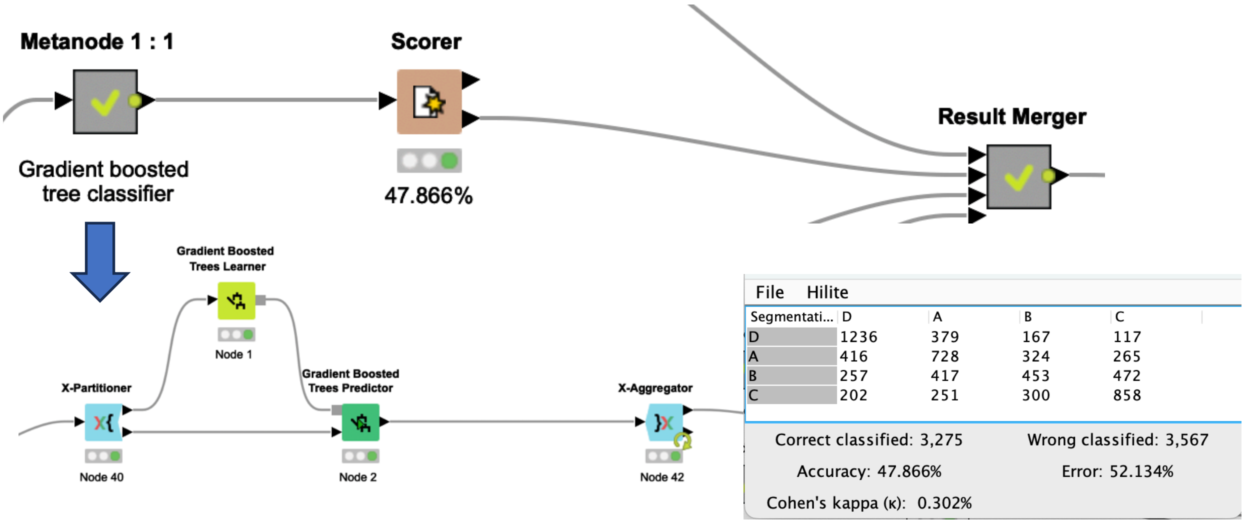
2. Classification Model:

A machine learning algorithm is trained to predict the class or category of a target variable based on input information in order to create a classification model. KNIME offers an intuitive platform for creating, optimizing, and testing classification models. The two types of classification problems are Binary classification and Multilabel classification. We are using the Multilabel classifier for our dataset since we have to predict four classes (A, B, C, D). Some examples of Classification models are Decision trees, SVM, Neural networks, K-nearest neighbors and ensemble methods like the Random forest, Gradient boosting, Ensemble tree, XGBoost tree etc.,[1,2] The performance of these models is increased by using Cross-validation, Parameter optimization and Feature selection methods. The evaluation metrics used for these models are Accuracy, Precision, Recall, F1-Score, Sensitivity, Specificity, ROC curve etc.,

Here in this study, we use the Gradient-boosted tree Classification model and Ensemble Tree Classification model as classification models.

2.1 Gradient Boosted Tree Classification Model with Cross Validation:

Gradient Boosted Trees (GBT) is an ensemble learning approach that combines the predictions of numerous weak learners (often decision trees) to produce a powerful predictive model. It is part of the boosting algorithm family, which iteratively improves model performance by focusing on mistakes produced by prior models in the ensemble. In gradient-boosted trees, the term "gradient" refers to the approach of minimizing the loss function via gradient descent optimization. It has the following advantages High Predictive Accuracy, Robustness to Overfitting, and Feature Importance. However, GBT has several drawbacks, such as possible hyperparameter sensitivity and lengthier training durations when compared to other methods.



Segmentati...	D	A	B	C
D	1236	379	167	117
A	416	728	324	265
B	257	417	453	472
C	202	251	300	858

Correct classified: 3,275 Wrong classified: 3,567

Accuracy: 47.866% Error: 52.134%

Cohen's kappa (κ): 0.302%

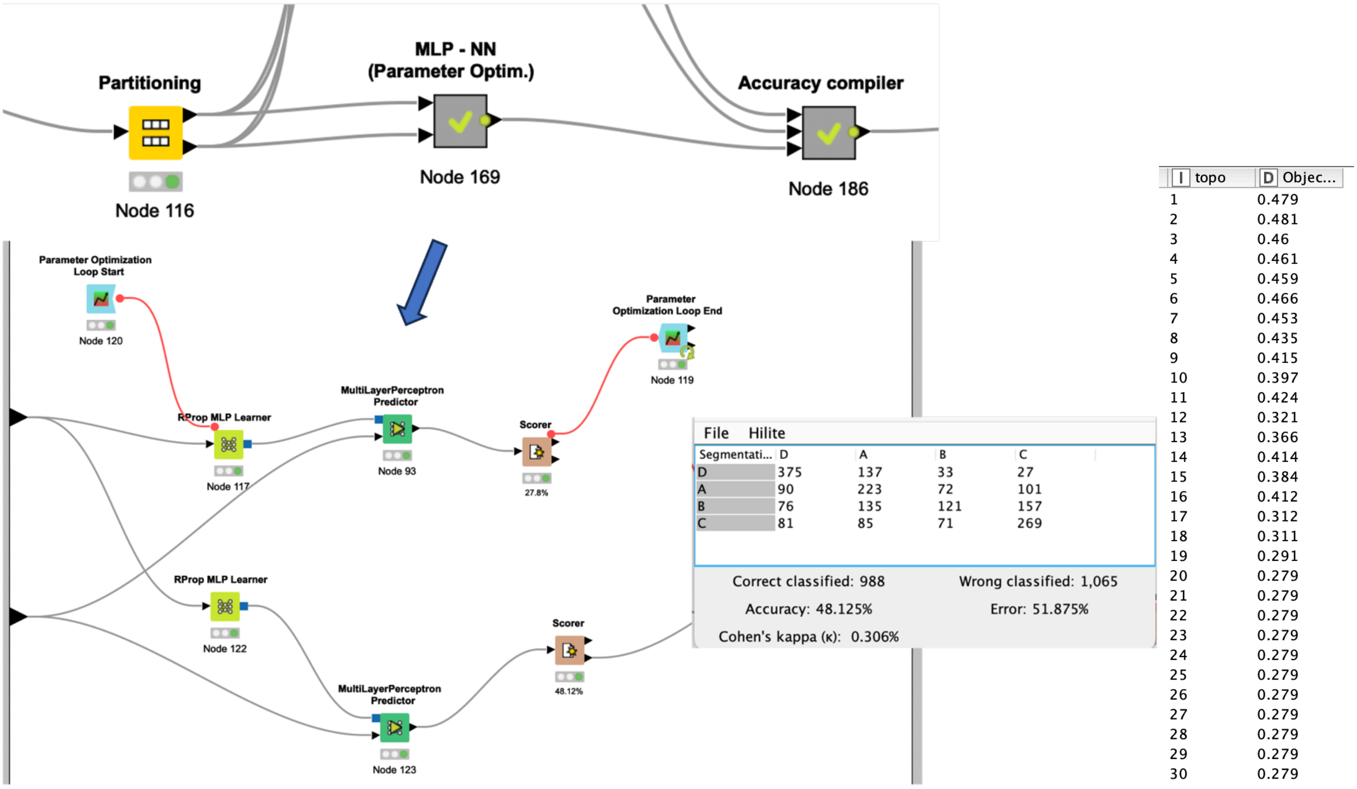
Figure 12- GTB Meta node with Cross-Validation and Accuracy Statistics

In KNIME, the implementation of a Gradient-boosted tree classifier involves using the available ‘Gradient Boosted Tree Learner’ node and ‘Gradient Boosted Trees Predictor’ node to create and train a model based on the Gradient Boosted Tree algorithm. The ‘Scorer’ node is used to show the performance of the algorithm.

The attained accuracy of 48.866% and the error rate of 52.134% in the application of Gradient-boosted trees (GBTs) to segmentation data prediction highlights a major challenge: the dataset's intrinsic restriction in both size and variety. Even though these measures might not seem like much at first, it's important to understand the unique benefits that GBTs offer, particularly for projects using little data. The GBTs' iterative learning process, which corrects faults with each subsequent tree, demonstrates their resilience in identifying significant patterns even from smaller datasets and helps the model adapt and enhance its prediction powers over time.

2.2 Multilayer Perceptron Predictor – Neural Network with Parameter Optimization:

A Multilayer Perceptron (MLP) is a sort of artificial neural network composed of numerous layers of nodes, each of which is linked to the nodes in the layers above and below it. It is a feedforward neural network, which means that information flows in only one direction—from the input layer to the output layer via the hidden layers. MLPs are commonly employed in machine learning applications such as classification and regression. It has the following advantages non-linearity and flexibility, feature learning, parallel processing, end-to-end learning, and high robustness against noisy data. Multilayer Perceptron predictors are versatile and can learn complicated correlations in data, but they require careful tuning of hyperparameters and can be difficult to understand.



Segmentati...	D	A	B	C
D	375	137	33	27
A	90	223	72	101
B	76	135	121	157
C	81	85	71	269

Correct classified: 988 Wrong classified: 1,065

Accuracy: 48.125% Error: 51.875%

Cohen's kappa (κ): 0.306%

topo	Objec...
1	0.479
2	0.481
3	0.46
4	0.461
5	0.459
6	0.466
7	0.453
8	0.435
9	0.415
10	0.397
11	0.424
12	0.321
13	0.366
14	0.414
15	0.384
16	0.412
17	0.312
18	0.311
19	0.291
20	0.279
21	0.279
22	0.279
23	0.279
24	0.279
25	0.279
26	0.279
27	0.279
28	0.279
29	0.279
30	0.279

Figure 13- MLP-NN with Parameter Optimization, Accuracy Statistics, Layer Params.

In KNIME, the implementation of an MLP classifier involves using the available 'RProp MLP Learner ' node and ' Multilayer Perceptron Predictor' node. We deployed the Parameter Optimization loop [3] on hidden layers to increase the performance of the Model from 27.8 % to 48.125%, The model with 2 hidden layers is the best performance model. The 'Scorer' node is used to show the performance of the algorithm.

3. Model Evaluation:

The Multilayer Perceptron Neural Network (MLP-NN) outperforms the Gradient-boosted trees (GBT) model with a higher test data validation accuracy of 50.98% compared to 45.06% on unseen data. MLP-NN's strength lies in its ability to capture complex non-linear relationships in the data. However, a critical drawback is the model's sensitivity to null values, preventing it from working on test data with missing values.

GBT, on the other hand, is generally robust and effective, particularly in handling null values due to its tree-based structure. It may require hyperparameter tuning for optimal performance. The choice between the two models depends on priorities: GBT for robustness and ease of handling missing values, or MLP-NN for its capacity to capture intricate patterns, provided null values are addressed through imputation or preprocessing. Further exploration of hyperparameters and architectures for both models is recommended to enhance their respective performances.

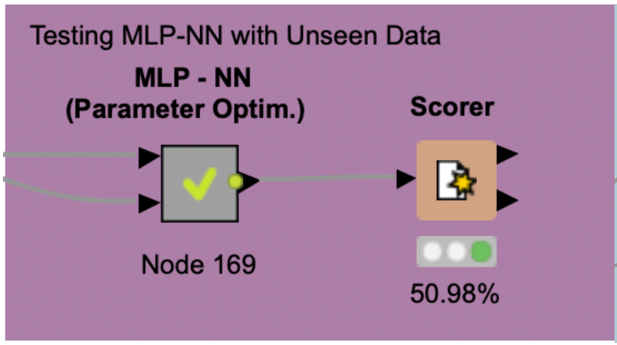


Figure 14 _ MLP-NN Testing Flow

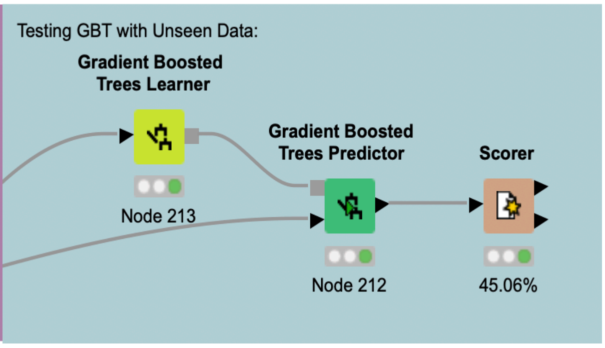


Figure 15_GBT Testing Flow

Model	Training Accuracy	Testing Accuracy
MLP-NN with Parameter Optimization	48.12%	50.98%
GBT with Cross Validation	47.86%	45.06%

Table 4 _ MLP-NN & GTP Training & Testing Accuracy scores

3.1 Evaluation Methods:

In the context of multilabel classification problems, accuracy serves as a primary evaluation metric due to its comprehensive and balanced assessment of the model's performance. Unlike precision, recall, sensitivity, and F1-score, accuracy considers both true positives and true negatives across all classes, making it well-suited for scenarios where instances may belong to multiple classes simultaneously. Its simplicity and ease of interpretation make accuracy a preferred metric for communication with both technical and non-technical stakeholders. Furthermore, accuracy's uniform treatment of all classes and reduced sensitivity to class imbalances contribute to its effectiveness in capturing the model's overall correctness. While precision, recall, sensitivity, and F1-score offer valuable insights into specific aspects of the model's performance, accuracy remains a robust and widely applicable metric for assessing the success of multilabel classification models.

3.2 Result :

To solve this classification Problem, 35 models were developed with the combination of Cross-validation, Parameter Optimization, Feature Optimization [3] and Dimensionality reduction (PCA and LDA) techniques. The top two models were discussed in this study, these two models were even performing better on Unseen test data. All the below-mentioned 35 model deployments were mentioned in the KNIME file.

Model Description	Accuracy	Cohen's kappa
Multilayer NN (Parameter Optimization)	48.12%	30.62%
Gradient Boosted Tree (X-Cross)	47.87%	30.23%
Gradient Boosted Tree (Feature + Parameter) Optimization + LDA	47.74%	30.24%
Multilayer NN (Feature + Parameter) Optimization + LDA	47.44%	29.85%
Gradient Boosted Tree (Feature Val + X-Cross)	47.41%	29.62%
Random Forest (Feature Val + X-Cross)	47.38%	29.54%
Random Forest (X-Cross)	47.35%	29.49%
Multilayer NN (Feature Val + X-Cross)	47.31%	29.51%
Multilayer NN (Feature + Parameter) Optimization + PCA	47.10%	29.36%
Gradient Boosted Tree (Parameter Optimization)	47.10%	29.29%
Tree Ensemble (Feature Val + X-Cross)	46.86%	28.83%
Tree Ensemble (X-Cross)	46.81%	28.77%
Multilayer NN (Feature + Parameter Optimization)	46.71%	28.70%
Gradient Boosted Tree (Feature + Parameter) Optimization + PCA	46.71%	28.85%
Gradient Boosted Tree (X-Cross) +PCA	46.48%	28.36%
Decision Tree (Parameter Optimization)	46.42%	28.46%
Multilayer NN (X-Cross)	46.29%	28.21%
XG Boost (Feature Val + X-Cross)	46.27%	28.08%
Multilayer NN (X-Cross) +PCA	46.08%	27.84%
Decision Tree (Feature + Parameter Optimization)	45.45%	27.06%

Gradient Boosted Tree (Feature + Parameter Optimization)	45.40%	26.96%
Decision Tree (Feature + Parameter) Optimization + LDA	45.40%	27.11%
Decision Tree (Feature + Parameter) Optimization + PCA	45.01%	26.46%
XG Boost (X-Cross)	44.78%	26.14%
K Nearest Neighbor (Feature Val + X-Cross)	43.26%	23.93%
K Nearest Neighbor (X-Cross) +PCA	43.23%	23.90%
XG Boost (X-Cross) +PCA	43.06%	23.85%
Decision Tree (Feature Val + X-Cross)	42.43%	23.05%
SVM (X-Cross)	42.37%	22.59%
K Nearest Neighbor (X-Cross)	42.25%	22.56%
Tree Ensemble (X-Cross) +PCA	40.91%	20.85%
Random Forest (X-Cross) +PCA	40.87%	20.79%
Decision Tree (X-Cross) + PCA	40.54%	20.51%
SVM (Feature Val + X-Cross)	40.11%	19.58%
Decision Tree (X-Cross)	39.67%	19.33%
SVM (X-Cross) +PCA	33.32%	10.56%

3.3 Conclusion:

Finally, the purpose of this research was to address a car company's development into new areas by capitalising on the success of its existing customer segmentation approach. The data was thoroughly understood and pre-processed, which included addressing missing values, deleting duplicates and outliers, and rejecting irrelevant features. To acquire insights into the distribution and properties of the data, visualisation techniques such as box plots, pie charts, and conditional box plots were used. The data was then processed for classification models by feature encoding and scaling. The Gradient Boosted Tree (GBT) and the Multilayer Perceptron Neural Network (MLP-NN) models were created and tested. The MLP-NN model outperformed the GBT model, with greater training and test data validation accuracy of 48.12% and 50.98%, compared to 47.86% and 45.06%, respectively. However, the MLP-NN is not generalised and cannot perform on null values, making predictions on test data with missing values difficult.

Because of its complete assessment across all classes in the multilabel classification problem, the model evaluation method used 'Accuracy' as the major metric. The findings showed that MLP-NN obtained better accuracy, highlighting its ability to capture complex relationships in data. 35 models were created in this study using various combinations of cross-validation, parameter optimisation, feature optimisation, and dimensionality reduction strategies. The top-performing models were discussed, and the MLP-NN with parameter optimisation was found to be the most accurate, with a training accuracy of 48.12% and a testing accuracy of 50.98%.

In conclusion, this study not only addressed the original issue statement, but also offered a deep examination of the data, effective pre-processing procedures, visualisation approaches, and the building and assessment of classification models. The findings provide important insights into the automaker's global development strategy, which will guide future decision-making processes.

References:

[1] C. Chauhan and S. Sehgal, "Sentiment Classification for Mobile Reviews using KNIME," *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, Greater Noida, India, 2018, pp. 548-553, doi: 10.1109/GUCON.2018.8674946.

[2] H. Çelik and A. Çinar, "An Application on Ensemble Learning Using KNIME," *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, Sakheer, Bahrain, 2021, pp. 400-403, doi: 10.1109/ICDABI53623.2021.9655815.

[3] https://hub.knime.com/knime/extensions/org.knime.features.optimization/latest/org.knime.optimization.internal.node.parameter.loopstart.LoopStartParOptNodeFactory

[4] https://hub.knime.com/knime/extensions/org.knime.features.ensembles/latest/org.knime.base.node.mine.treeensemble2.node.gradientboosting.learner.classification.GradientBoostingClassificationLearnerNodeFactory2

[5] https://hub.knime.com/knime/extensions/org.knime.features.js.views.labs/latest/org.knime.base.node.stats.dataexplorer.DataExplorerNodeFactory