Module Code: CSMDM21

Assignment report Title: Data-Driven Customer Segmentation: Analysis and Predictive Decision-Making in Market Expansion

Date of submission: 24th Nov, 2023

Actual hrs spent for the assignment: 60

1. INTRODUCTION

The pursuit of market expansion is both a challenge and a strategic imperative in the dynamic automotive industry. Our main subject is an aspirational automaker ready to expand into new markets using its current line of products. Motivated by previous achievements, the sales team of the company has adopted a segmentation strategy in the current market, categorizing customers into four groups (A, B, C, and D). Because of the remarkable outcomes of outreach and communication that are specifically tailored to each segment, it has been decided to repeat this strategy in the new markets.

Our job is to help the manager develop and test the classification models that will be needed to forecast the new market's customer segmentation. Our goal is to build predictive models that are aligned with the established segmentation framework by leveraging key variables such as gender, marital status, age, education, profession, work experience, spending score, family size, and an anonymized category (Var\_1).

This report summarizes our analytical journey, with the dataset from the Analytics Vidhya hackathon serving as the foundation. We intend to use machine learning techniques to uncover insights that will guide the company's targeted approach, ensuring a smooth transition and success in new markets.

A diagram of a process

Description automatically generated

Figure 1- Flowchart illustrating the sequence of actions taken to finalize the classifier.

We commence our analysis by examining the data through Excel, providing a fundamental understanding of its structure. Subsequently, leveraging KNIME [1], we employ diverse data transformation techniques to address missing values, clean the data, and perform other necessary adjustments. Following these transformations, we engage in data visualization to gain deeper insights, incorporating correlation analysis and feature extraction.

Once the feature set is finalized, we proceed to identify the optimal model by testing various features and tuning hyperparameters. Notably, the decision tree and an ensemble model combining DBSCAN [4] and k-nearest neighbors (knn) exhibit the most promising outcomes. The data is then fitted using the training dataset, and testing validates the superior performance of these models. After this, we evaluate the models using unseen data, employing the hold-out technique for validation.

Task1 workflow in KNIME explains about the data transformation, cleaning, insights. After saving the csv files as train\_test.csv and validation.csv. We use this as input for Task2\_3 workflow for classification and evaluation.

Interestingly, our ensemble model reveals shortcomings, indicating a lack of generalization. In contrast, the decision tree consistently demonstrates better results, even when applied to previously unseen data. The summarized flowchart in Figure 1 provides a visual representation of the entire process and the sequential steps taken in our analysis.

1. DATA UNDERSTANDING

A pie chart with numbers and a few rows of red blue and green

Description automatically generated with medium confidence

Figure 2- Pie chart demonstrating uniform distribution of segmentation types.

The basic understanding of data is done on excel. The total size of data is 10695 rows with 10 columns as data columns to predict segmentation column. The data has duplicate entries which can be dropped which is done by “duplicate row filter” in KNIME. This filter drops all the duplicate values where everything is same. While dropping values there is one interesting thing to note, that when we do not consider column “ID” for dropping duplicate values we are dropping around 600 entries, while with ID there are around 10650 entries. Also, while having a closer look at the data and checking for various columns, the column “ID” is the unique id which is used for numbering each row and does not hold relevance for finding dependency on segmentation column. We drop this column by using “column filter” in KNIME.

The dataset is categorized into four distinct segmentation types: A, B, C, and D. Looking at the distribution in the "Segmentation" column, it is noticeable that the data exhibits a symmetrical distribution without significant imbalances. This observation is visually depicted in the provided “Pie chart(local)” in KNIME (Figure 2), illustrating an even distribution of the total dataset into four segments, with each segment containing a relatively uniform number of rows ranging between 1900 and 2500.

Furthermore, it is imperative to retain a subset of the data for hold-out validation. We adopt a strategic partitioning approach, utilizing the "Partitioning" method from KNIME to achieve an 80-20 split. In this strategy, 80% of the data undergoes further analysis, while the remaining 20% is set aside as unseen data for validation. The strategic partitioning methodology ensures that both sets are replicas of each other, promoting an unbiased representation in both the training and validation datasets.

1. TRANSFORMATION AND CLEANING

The dataset comprises numerous missing values and outliers, requiring systematic handling to enhance the model's performance. Therefore, we will address these issues on a column-by-column basis.

Column "Age":

A diagram of a graph

Description automatically generatedA diagram of a blue and black line

Description automatically generated with medium confidence

Figure 3- Boxplot showing before and after removing outliers from column “Age.”

Column "Var1":

This column contains 7 categories, numbered from 1 to 7. Unfortunately, we lack specific information about these categories. In the absence of additional details, the most pragmatic solution currently is to drop the missing values. Handling these missing values differently could be explored if supplementary information regarding the categorization becomes available.

Column "Work Experience":

A screen shot of a diagram

Description automatically generated

Figure 4- Scatterplot showing distribution of work experience over age.

By plotting a scatter plot on KNIME using “Scatter plot” node, we find that the average of work experience across various ages from 18 to 84 years using the available information, distinct clusters form in10-year spans, such as 16 to 24 years or 25 to 34 years (Figure 4). As a result, we opt to substitute missing values with the average work experience values corresponding to the relevant age group using Column Expression node [3]

Column "Ever\_Married":

Values are categorized as "Yes" or "No" based on the median value. If greater than the median, it is labeled "Yes"; otherwise, it is labeled "No." This is done using Column Expression node[3]

Column "Graduated":

Considering that graduation typically occurs around ages 21, 22, or 23, varying based on countries, circumstances, and courses, we opt for an average age of 22. Individuals older than 22 are categorized as "Graduated," while those younger than or equal to 22 are labeled as "Not Graduated." This categorization is applied to fill the null values in the dataset using Column Expression node [3]

Column "Profession":

For individuals younger than 22 who have not graduated and lack any work experience, we assume that the person is currently not working. Consequently, we designate the missing values in these conditions as "NA," if applicable. Moreover, if the age is below 22 and the work experience exceeds 5 years, we observe that these individuals are predominantly labeled as "Homemaker." Thus, we replace the missing values under similar conditions with "Homemaker." Any remaining null values are filled with "Unidentified." Additionally, considering the practicality, it is implausible to have more than 10 years of experience as a doctor, engineer, or executive if the age is less than 22. Consequently, we choose to drop these values. All this is handled by Column Expression node [3] in KNIME

Column "Family Size":

Like work experience, for the "Family Size" column, plotting the average family size across different ages reveals clustering within 10-year spans. Considering this observation, we decide to replace missing values in the "Family Size" column with the average family size values corresponding to the relevant age group using Column Expression node. This approach aims to enhance the dataset by imputing missing values in a manner reflective of the observed patterns in the data.

A chart with numbers and points

Description automatically generated with medium confidence

Figure 5- Scatterplot showing distribution of family size over age

Interrelated Columns (Work Experience, Graduation, Profession):

Given the relationship established in the previous explanations between work experience, graduation, and profession, we can infer that these variables are interrelated. Consequently, we decide to drop the rows where all three columns have missing values.

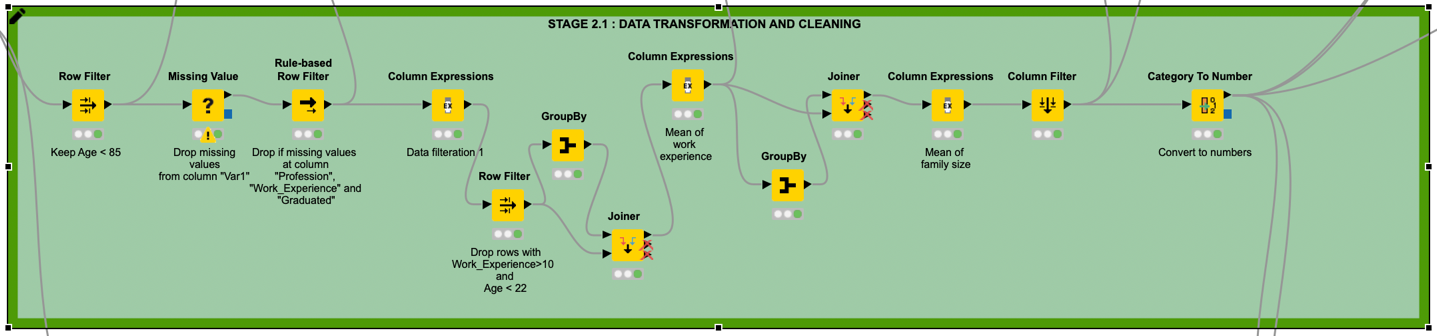


Figure 6- Data Transformation and Cleaning in KNIME

These steps are taken to ensure that our dataset is reliable and trustworthy. It's as if we're cleaning up and repairing any flaws in the data so that when we look at it, we can be confident that it's reliable and provides us with the right information for our analysis. The image below shows the overall steps taken to make the data work efficiently for our prediction (Figure 6)

1. DATA VISUALISATION

Data visualization simplifies complex information, making patterns and trends easily understandable. Various visualization tools, including line charts and pie graphs, play specific roles, assisting both experienced professionals and beginners in comprehending intricate datasets. Beyond mere appearance, data visualization provides clarity and valuable insights, helping decision-makers identify opportunities and tackle challenges. In our model, we utilized diverse visualizations such as Pie charts(local), Scatter Plot, Bar Chart, Sunburst Chart[ref], Statistics etc. to illustrate different trends within the data (Figure 7)

A screen shot of a computer

Description automatically generated

Figure 7- Data Visualization in KNIME

Following the completion of data cleaning and transformation, it is observed that there are no missing values in the final dataset, comprising approximately 8700 entries for both training and testing. For a comprehensive overview of basic statistics, including minimum, maximum, and average values, refer to the Statistics outlined in Table 1 from KNIME

A screenshot of a data

Description automatically generated

Table 1- Statistical data of features for dataset

Additionally, an exploration of the data distribution across segments A, B, C, and D reveals the number of individuals categorized by their spending levels—low, average, and high. This analysis provides valuable insights into the overall distribution of spending patterns within each segment. (Figure 8)

The analysis indicates notable disparities in spending patterns among different segments. In segments A and B, the difference between the number of individuals with low spending scores compared to those with high and average scores is substantial. This gap decreases in segment B. In contrast, for segment C, there are more individuals with average spending scores than those with high or low scores. The number of people with high spending scores appears relatively consistent across all segments. While segment C exhibits the highest number of individuals with average spending, segment D has a considerably smaller population in the average spending score category.

A graph with different colored bars

Description automatically generated

Figure 8- Spending score count for different types of segmentation

Similarly, we can analyze each parameter in comparison with segmentation using the Sunburst model in KNIME. By interacting with the model and clicking, we obtain the percentage of data falling under each main category and its subcategories. An example of this analysis is depicted in the image below (Figure 9). This insight offers a concentrated view of a specific demographic within the dataset, providing information on the probability of a particular combination of attributes and behavior.

A pie chart with different colored circles

Description automatically generated

Figure 9- The image illustrates that the probability of a male individual, specifically an artist, with an average spending pattern falling under category 6 is 5.15%.

1. FEATURE EXTRACTION AND FINDING CORRELATION

To enhance visualization and maintain consistency across features, we've normalized the data using the Normalizer node in KNIME. This process ensures that all features fall within the range of 0 to 1, facilitating the creation of clearer and more standardized graphs for analysis. Normalizing the data allows for a more accurate comparison and interpretation of patterns across different features. Upon examination from bar chart (Figure 10), it appears that work experience does not exhibit significant differences across different segments, suggesting that this column may be less influential in the analysis. The normalized age displays variations ranging from 0.2 to 0.4, indicating the most diverse data across segmentation. The normalized age seems to be a more impactful variable with variations reflective of the different segments. Family size demonstrates some variations across segmentation, though not as pronounced as age. Further testing through feature extraction can help determine the significance of family size in the model.

A graph of a bar chart

Description automatically generated with medium confidence

Figure 10- Distribution of Age, Work Experience and Family size across Segmentation

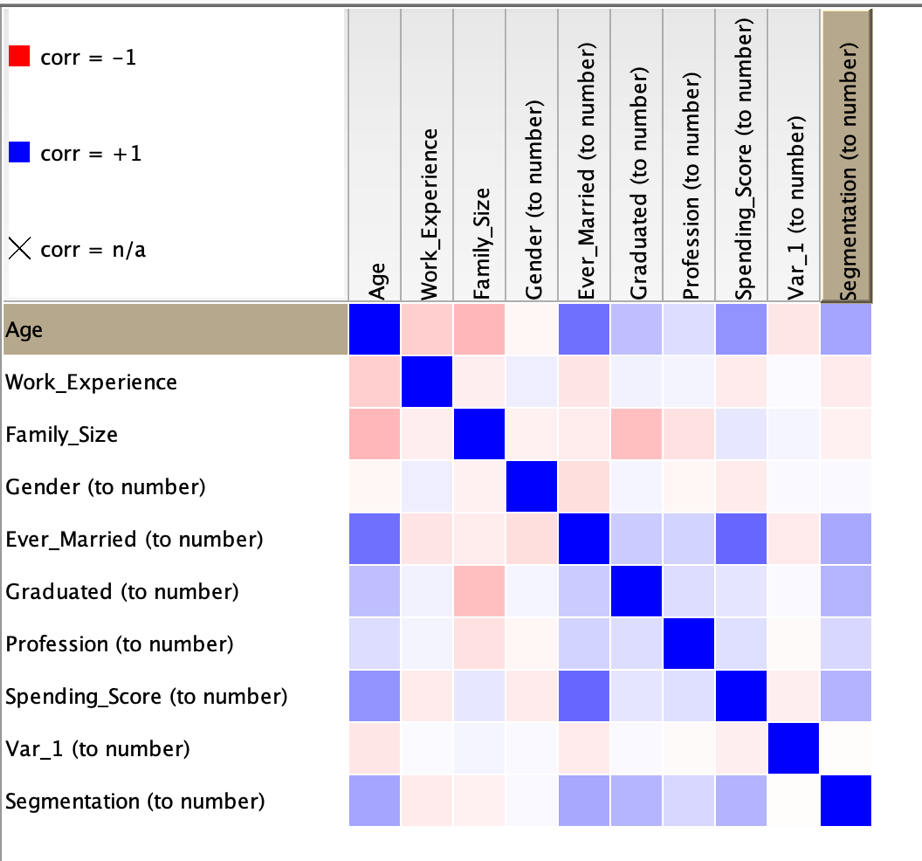


Figure 11- Linear correlation between different features in dataset.

Furthermore, as depicted in Figure 11, the correlation analysis done using Linear Correlation node [5] from KNIME reveals that the variable "Var\_1" exhibits almost no correlation with other variables. Consequently, we can consider excluding this column from our classification model, as it appears to contribute minimally to the overall correlation structure.

1. TRY VARIOUS CLASSIFIER AND HYPERPARAMETER TUNING TO GET BEST CLASSIFICATION MODEL

Classification is a fundamental and widely used technique in machine learning, and it finds application in various domains, including business, healthcare, finance, and more. The primary purpose of classification is to categorize or label input data into predefined classes or categories based on their features or attributes. Classification is a versatile tool in machine learning that allows us to automate decision-making processes, recognize patterns, and make predictions based on labeled data. Its widespread use is a testament to its effectiveness in solving a diverse range of problems across different industries.

After tackling the data transformation, cleaning, and extracting meaningful features, the next challenge is to find the ideal classifier model for predicting data outcomes. We started with the basics, testing linear classifiers to establish a foundation. As we delved deeper, more sophisticated models like k-nearest neighbors (KNN), support vector machine (SVM) with radial and polynomial features, decision trees, random forest, and ensemble tree classifiers were thrown into the mix. Interestingly, decision trees, random forest, and KNN stood out, each showing promising accuracies around 45%, 45%, and 40%, respectively.

We also examined the linear regression model to distinguish segmentation type "A" from others and "B" from others. Utilizing various models and refining hyperparameters, each individual model exhibited an accuracy surpassing 55%. However, when combining these models, the collective accuracy regressed to 40%, falling short of the performance achieved by the existing models. This underscores the importance of a thoughtful approach, considering both individual and ensemble models for optimal results. Additionally, the model appears to perform well in distinguishing D and C values rather than A and B. When running the same model to discern "D" from others, it achieves an accuracy of approximately 59%. This observation indicates a certain specificity in the model's proficiency for certain segmentation types.

We also explored ensemble methods, weaving multiple models together and using decision trees strategically to pinpoint the right segmentation type, even this did not improve the accuracy. We introduced k-means clustering to form data clusters. Leveraging this cluster information as a feature for the KNN model brought a welcomed boost in accuracy.

Refining our models involved scaling and normalizing values. Later, experimenting with the DBSCAN technique for the KNN model proved fruitful, pushing the accuracy needle to around 48%. Using hyperparameter tuning in KNIME with “Parameter Optimization Loop Start” node and “Parameter Optimization Loop End” node between k=2 and k=30 and running it for every one step, we got the best accuracy and least errors with train and test data at k=25. We tried this parameter on train and test and got accuracy around 47% (Figure 12) with K Nearest Neighbor node in KNIME.

A diagram of a diagram

Description automatically generated with medium confidence

Figure 12- Image showing ensemble method with hyperparameter tuning training and evaluation.

A diagram of a process flow

Description automatically generated

Figure 13- Decision tree method with hyperparameter tuning training and evaluation.

For the second model, I opted for the decision tree technique. While the decision tree itself is yielding satisfactory results, other models like random forest and ensemble methods are not exhibiting higher performance like mentioned in [2] Consequently, incorporation of normalization techniques in conjunction with the decision tree improved the performance. In this case, DBSCAN did not contribute to any significant improvement, so the basic parameters were retained. The highest accuracy achieved for this model stands at 48%. In KNIME, we have different Decision Tree Learner and Decision Tree Predictor as a Decision Tree model. (Figure 13)

This iterative and hands-on approach emphasizes the need for thoughtful experimentation in crafting a robust classification model tailored to the unique characteristics of the data.

1. RELIABLE ACCURACY USING HOLD OUT VALIDATION

The model is assessed on the validation set after it has been trained and tested. Accuracy, precision, recall, and other performance metrics are computed by comparing the test set's actual results with the model's predictions. For the same, we are employing the hold-out validation technique here.

The hold-out method's main objective is to evaluate the model's ability to generalize to previously undiscovered data. A model may have picked up on underlying patterns in the data and be able to predict new, unseen instances with accuracy if it performs well on both the test and validation sets. Overfitting can also be found with the aid of the hold-out method. It is possible that a model is overfitting the training data and not properly generalizing to new data if it performs remarkably well on the test set but poorly on the validation set. Depending on the test set performance, hyperparameter tuning or model modifications might be required in some circumstances. There may be several iterations of this process, involving training, evaluation, and improvement.

We can see the evaluation section in both decision tree and ensemble method (Figure 11 and Figure 12) For assessment purposes, we employ the hold-out method, partitioning the data and reserving 20% of the total dataset before engaging in data cleaning and transformation.

This reserved data subset is subsequently utilized to assess the performance of our models. For instance, when evaluating the KNN model, it demonstrates strong performance during the training-test phase. However, when applied to the remaining 20% of unseen data, the results indicate an accuracy of up to 35%. This suggests potential overfitting, indicating a lack of generalization to new, unseen instances. We can refer Table 2 which shows the overall values detected by the two algorithms and we can clearly depict that decision tree is more generalized model than the ensemble method used.

In contrast, the decision tree model undergoes evaluation with results that align closely with the training set outcomes. The validation technique yields an accuracy of 44%, indicating the model's reliability and consistency in maintaining performance beyond the training data.

A table with numbers and a few letters

Description automatically generated with medium confidence

Table 2- Comparison of TP, TN, FP, FN using two algorithms.

1. DISCUSSION

Compared to the KNN model, decision trees have several advantages in the given data and evaluation results. First, because of their transparent representation of the decision-making process via a tree structure, decision trees are naturally interpretable. For comprehending and elucidating the variables influencing predictions, interpretability can be extremely important. Second, because decision trees pick only the pertinent features, they are a good fit for datasets of different sizes and demonstrate computational efficiency, particularly in the training stage. Decision trees are also capable of efficiently handling irrelevant features by prioritizing the most informative ones. Interestingly, decision trees also show good handling of missing values, which strengthens them in situations where data completeness is an issue. The decision tree's consistency in performance over the training set and validation data is another indication of how well it generalizes to new situations. Together, these characteristics make decision trees a strong and understandable option for the dataset, providing benefits over the KNN model.

Decision trees have limitations even though they have many benefits. Their sensitivity to even minute changes in the data is one of their main limitations. Decision trees are less stable than some other machine learning models because small changes to the dataset or input features can result in a different tree structure. Multiple interacting features in complex relationships may be difficult for decision trees to capture. Complicated and nonlinear true decision boundaries may lend themselves more to the use of ensemble methods or simpler models. Additionally, if classes predominate in the dataset, decision trees may produce biased predictions. This problem can be lessened by balancing the dataset or by employing strategies like ensemble methods.

We have experimented with various methods and combinations, but this represents the best achievable outcome. However, there are areas for potential improvement:

Data Quality: The dataset is limited and inadequately cleaned, preventing the model's performance from surpassing 50%. Cleaning procedures, including outlier removal, and handling missing values, have a substantial impact due to the dataset's small size.

Data Distribution: Addressing imbalances in the dataset through oversampling, under sampling, or synthetic data generation could enhance dataset size and reduce bias. However, attempts to modify the distribution, especially through resampling, contribute to issues like outliers, missing values, and inaccuracies, thereby decreasing overall accuracy.

Hyperparameter Tuning: Hyperparameter tuning yields optimal results on larger datasets. In the case of a small dataset, the tuning process tends to produce higher values without significantly affecting the results.

Insufficient Information for "Var\_1": The absence of information regarding "Var\_1" necessitates the removal of missing values without alternative handling methods. Additional insights into this column could offer better strategies for managing missing values.

Outlier Removal Challenges: Complete removal of outliers from all columns is unfeasible. The presence of outliers in certain columns may be integral to the dataset, and attempting universal removal may not be practical.

In summary, while the current approach represents the best achievable outcome, there are opportunities for enhancement, particularly in augmenting data quality, addressing imbalances cautiously, optimizing hyperparameters for smaller datasets, seeking additional information for certain columns, and refining outlier removal strategies.

1. CONCLUSION

The analysis began by comprehending the data, using Excel and KNIME to explore the dataset's structure, identify duplicates, and eliminate irrelevant columns. Symmetrically distributed segmentation types (A, B, C, D) were noted. Transformation and cleaning processes tackled missing values, outliers, and correlated columns to ensure data reliability. Data visualization played a crucial role, revealing spending patterns and parameter-segment relationships. Feature extraction normalized data, emphasizing age as a significant variable. Various classification models, including decision trees, random forests, and KNN, were explored, with decision trees showing promise. Ensemble methods, hyperparameter tuning, and hold-out validation were employed for model refinement. The discussion highlighted decision trees' advantages, such as interpretability and computational efficiency, while acknowledging limitations like sensitivity and potential bias. The conclusion acknowledged achievements but outlined areas for improvement, emphasizing data quality, cautious handling of imbalances, and refining outlier removal strategies. The overall aim of the analysis was to guide the company in informed decision-making for successful market expansion, emphasizing thoughtful experimentation and continuous model refinement.

REFERENCES

1. S. Omanovic, A. Midzic, Z. Avdagic, D. Pozderac and A. Toroman, "Missing Values Interpolation in PurpleAir Sensor Data based on a Correlation with Neighboring Locations using KNIME Analytics Platform," 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2023, pp. 291-295, doi: 10.23919/MIPRO57284.2023.10159808.
2. 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.
3. KNIME AG: “Column Expressions”. [Online]. Available: <https://hub.knime.com/-knime/extensions/org.knime.features.expressions/latest/org.knime.expressions.base.node.formulas.FormulasNodeFactory>
4. KNIME AG: “Clustering with DBSCAN”. [Online]. Available: <https://hub.knime.com/-knime/spaces/Academic%20Alliance/Guide%20to%20Intelligent%20Data%20Science/Example%20Workflows/Chapter7/03_DBSCAN~WgQVjVJugr9Nu24W/current-state>
5. KNIME AG: “Linear Correlation”. [Online]. Available: https://hub.knime.com/knime/-extensions/org.knime.features.base/latest/org.knime.base.node.preproc.correlation.compute2.CorrelationCompute2NodeFactory