**University of Central Lancashire**

**UCLan, Cyprus**

**School: School of Sciences**

**Assessment Title: "Optimizing New Product Marketing through Predictive Modeling and Data Mining Techniques in Retail Analytics"**

**Course Title: MSc. Data Analytics**

**Module Title: Knowledge Discovery**

**Module code: CO4762**

**Year of Study: 2024-2025**

**Name: Alamgir Hossain**



1. **Executive Summary:**

*Customer analysis is gathering and analyzing customer needs and desires. By gaining a deeper understanding of your customers, you can tailor your product or service offerings to meet their needs, increasing sales and customer loyalty. In addition, customer analysis can help you build stronger relationships with your clients and customers. You can establish trust and credibility by showing that you understand their needs and wants, leading to higher customer satisfaction and repeat business. Overall, customer analysis is crucial for any business looking to succeed in today's competitive market. By truly understanding your customers, you can gain a significant advantage over your competitors and position your business for long-term success. As a Data Analyst at a fictitious supermarket company, I help establish the marketing plan for a new product range. To accomplish this, I analyze the company's customer data and determine which customers will most likely buy these products. SuperApp is thrilled to introduce our new line of products, and we want to ensure that our valued customers have access to them. To help you get started, we're offering coupons for the new product line to all participants of our loyalty program. We're confident you'll love our latest products, so we want to ensure you have every incentive to try them out. To help us better understand your preferences, we've been collecting data on recent purchases of similar products. This information will help us customize our offerings to suit your needs better. In order to make our prediction, I developed thirteen different predictive models - Five non-parametric and Eight parametric. These models were created to identify the best individuals in a database to target for a new product range. Among the non-parametric models, I used Decision Tree algorithms, and among the parametric models, I used Regression, Neural Network, Support Vector Machine (SVM), and Clustering algorithms. To evaluate the accuracy of my models, I utilized various metrics. Ultimately, I selected the Neural Network (Extended NN) model as the champion model. In analyzing our ScoreData results, we observed that when our target equals 1, the EM\_PROFIT is displayed, and when the target equals 0, the EM\_PROFIT is 0. Additionally, we noticed that when IMP\_ProsperityClass is high, both the EM\_PROFIT and EM\_probability are high, and the majority of our predicted customers are female. This indicates that most of our product enthusiasts are female with an average age of 35. Based on my study, I recommend that we identify our most probable customers based on Target =1, EM\_ProsperityClass and EM\_PROFIT.*

1. **Data Preparation and Exploration.**

I used a supervised learning approach since the data included records of customers who recently purchased related products.

1. **The  Workflow:**
2. **Data importation:**

I've created a prediction model using a dataset that contains information about customer demographics, loyalty status, and purchase history. The dataset consists of over 22,000 observations and 12 variables aggregated at a customer level, offering a comprehensive view of attributes from a star schema. The variables in the dataset, along with their default roles and levels, are listed below. The attributes in the dataset describe customer demographics, loyalty, and purchase behavior. Additionally, there are three Interval Input, one Interval Id, two Input Ordinal and four Nominal Input variables. Furthermore, there are two target variables, one being Nominal and the other Binary.

In our analysis, we will categorize the variables to gain valuable insights:

- **Interval Input Variables:** We will use Age, Amount Spent, and Customer Retention to effectively predict the Target Variable.

- **Ordinal Input Variables:** Card Class and Prosperity Class will also play a significant role in our prediction model. Prosperity Class is measured on a scale from 1 to 100, where 100 represents the highest level of prosperity, while Card Class reflects loyalty status, including categories such as Bronze, Silver, Gold, and Platinum.

To enhance our understanding, we will first explore the correlation between Prosperity Class and the Target Variable.

- **Nominal Input Variables:** Additionally, we will investigate how District, Gender, TV Region, and Residential Type (Res Type) correlate with the Target Variable.

Lastly, we will use Customer ID as a Unique Identifier to ensure our analysis is precise and organized. This structured approach will help us uncover meaningful relationships within the data.

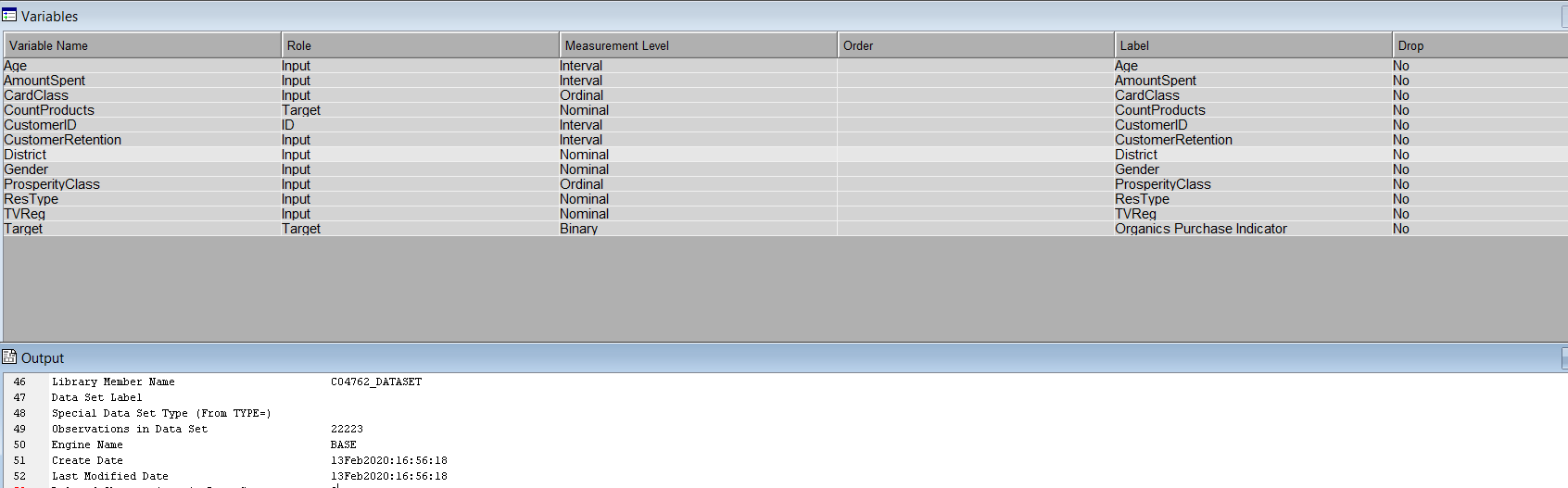


  Figure 1: Features of the Datasets

I added the co4762\_DATASET data source to the co4762\_diagram workflow and saved it as SAS files using the "save data" node. It will serve as the Input Data node for this exercise.

1. **Data Visualization**

Visualizing input data to identify potential patterns and check for missing values is essential before building a model. Weak data can result in an unreliable model. To better understand the customer data set, I used the "StatExplore" node to perform some descriptive analysis, such as calculating the mean, median, and normal distribution.

The output of the StatExplore node is presented in the browser display. From the variable summary results, we can see that there are two target variables: one is CountProducts with a frequency count of 4, and the other is Target with a frequency count of 2.

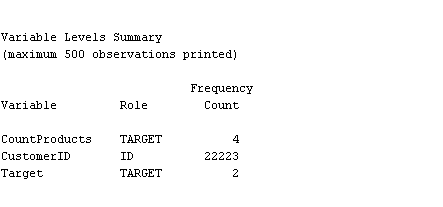


Figure 2: Variable levels summaries

Upon analyzing the class variable and interval summary statistics, we gained insights into the number of levels, missing values, mode, and mode percentage of the attributes. CardClass, District, and Target have no missing values. However, CountProducts has the highest count of missing values, with 16718 missing entries. It's worth noting that the majority of customers are from Lemesos, accounting for 38.85% of the customer base across the six districts. Additionally, Gold is the most common card class, representing 38.57% of the CardClass category among Gold, Silver, and Platinum. Female customers make up the majority, comprising 54.67% of the total customer base. Moreover, the percentage of the two target variables peaks at 75.23% when the target mode is 0.

Please take note of the following information: The recognized Maximum ProsperityClass is 100. When the ProsperityClass mode is 25, the percentage is 11.85, and when the mode is 21, the percentage is 11.65. ResType and TVReg denote the type of residential neighborhood and television region, respectively. Notably, when the TVReg mode is London, it accounts for the highest percentage.

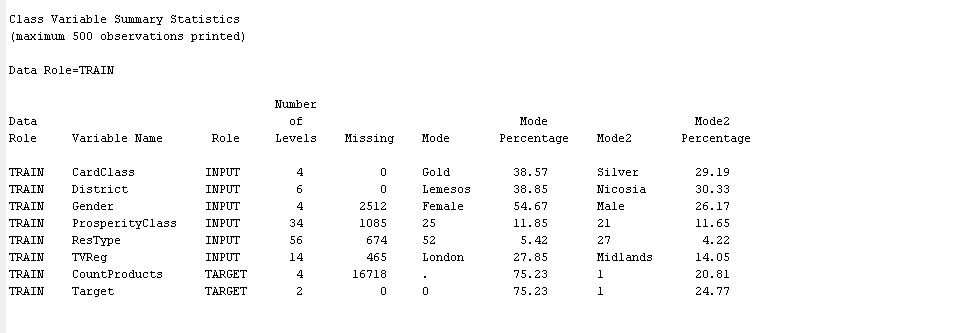


Figure 3: Class variables summaries

We can observe that AmountSpent counts the highest mean, median, and standard deviation from the interval variable summary statistics data. CustomerRetention and Age both have a few missing numbers.

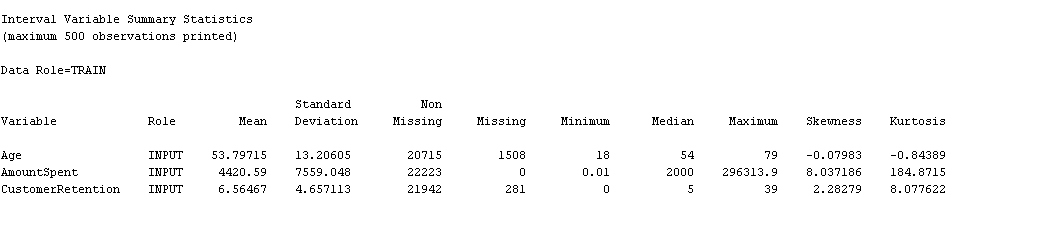


Figure 4: Interval variable summaries

**Second Target:**

SuperApp is excited to unveil our new line of products, and we’re committed to making them easily accessible for our valued customers. To achieve this, we will focus on estimating the number of products available as our second target. By analyzing our best-selling items, we can gain valuable insights into customer preferences and better understand which products resonate most with our audience. This approach will enable us to enhance our offerings and serve our customers even better.

I have employed a Chi-Square plot to effectively rank the top 10 variables based on their Chi-Squared statistics. Additionally, I developed a variable worth plot to systematically order the input variables according to their predictive value for the target variable. In our initial assessment, I identified 'CountProducts' as the target variable. The analysis of the Chi-Squared statistics, as shown in Figure 5 below, reveals insightful correlations between 'CountProducts' and other independent variables. Notably, our primary target demonstrates a strong correlation with 'CountProducts', while 'AmountSpent' displays a comparatively weaker correlation. With a p-value of less than 0.05, we can confidently reject the null hypothesis, indicating that 'ProsperityClass' has a significant correlation with the target variable.

Furthermore, it is important to recognize that our primary target is intertwined with various other variables, although it shows a lower correlation with 'AmountSpent'. Despite this, 'AmountSpent' maintains a substantial correlation with the primary target variable. This analysis not only highlights key relationships but also sets the stage for further exploration of these dynamics in our modeling efforts.

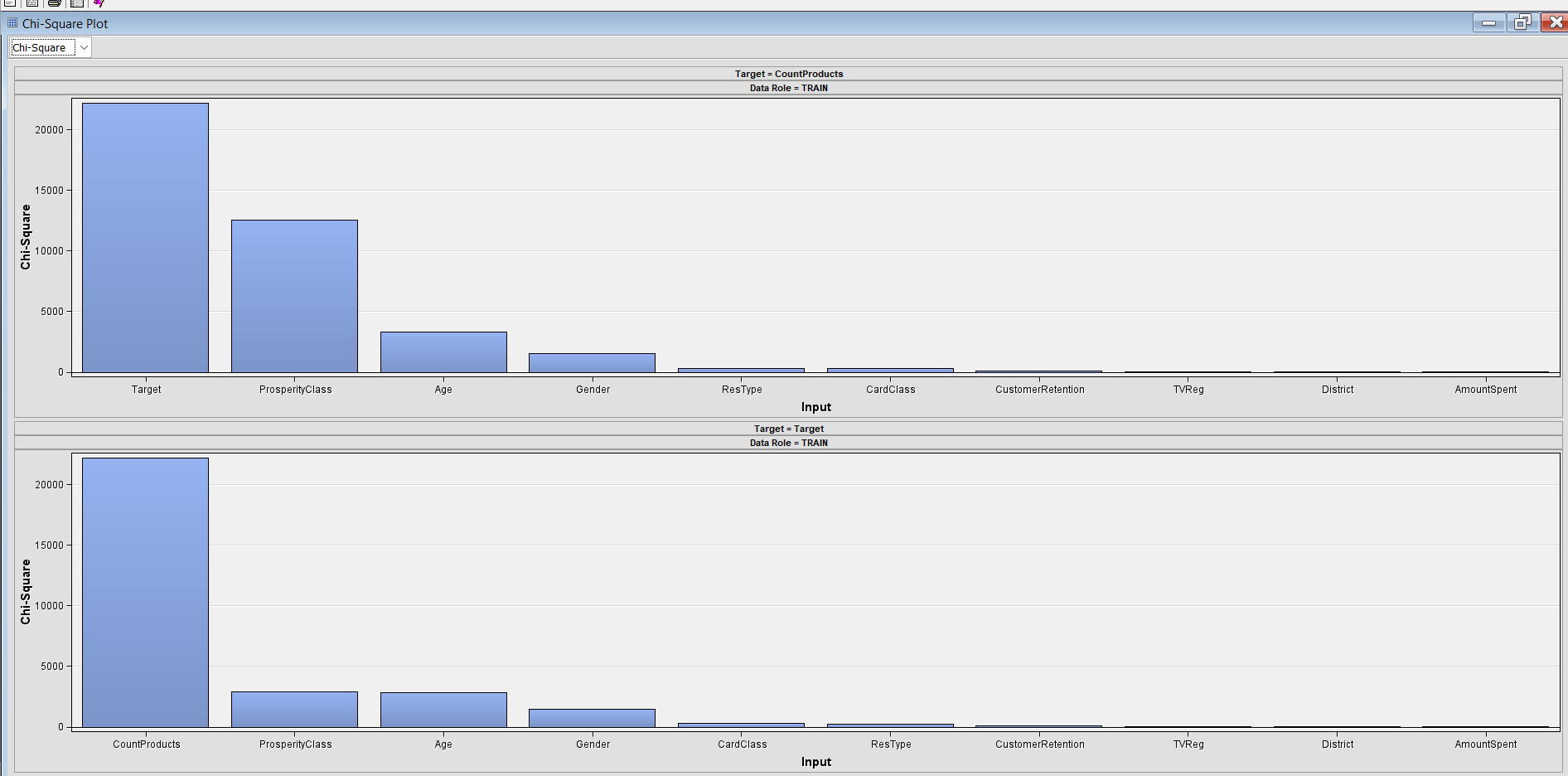


 Figure 5: Correlation between two Target and other attributes.

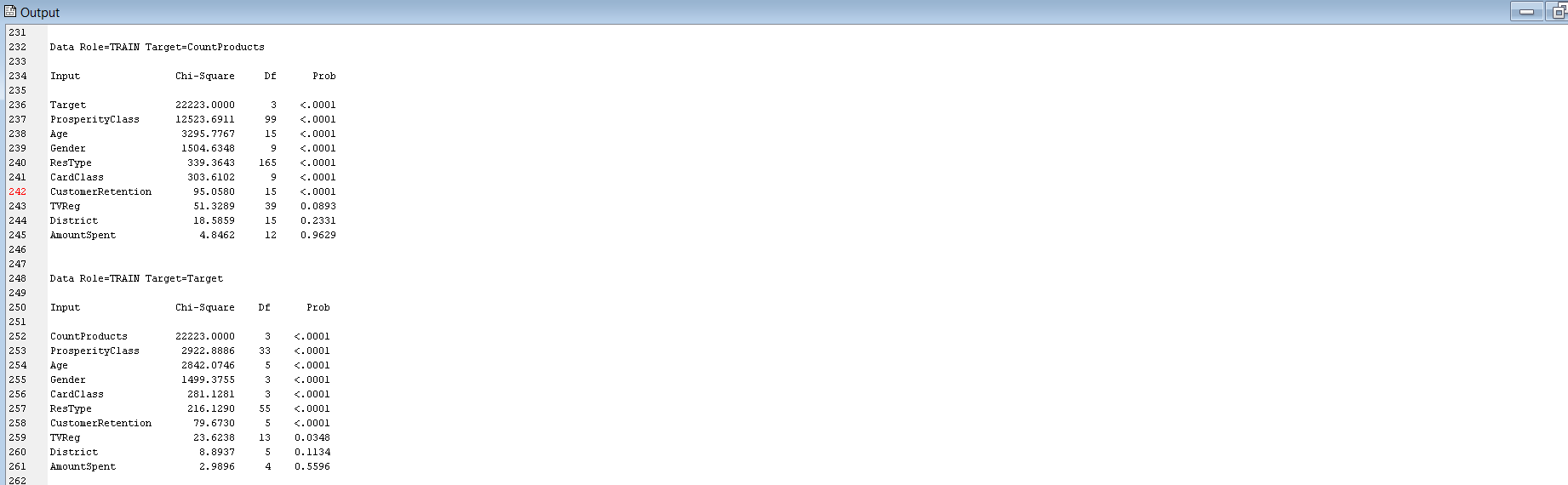


Figure 6: Chi-Square Statistics results

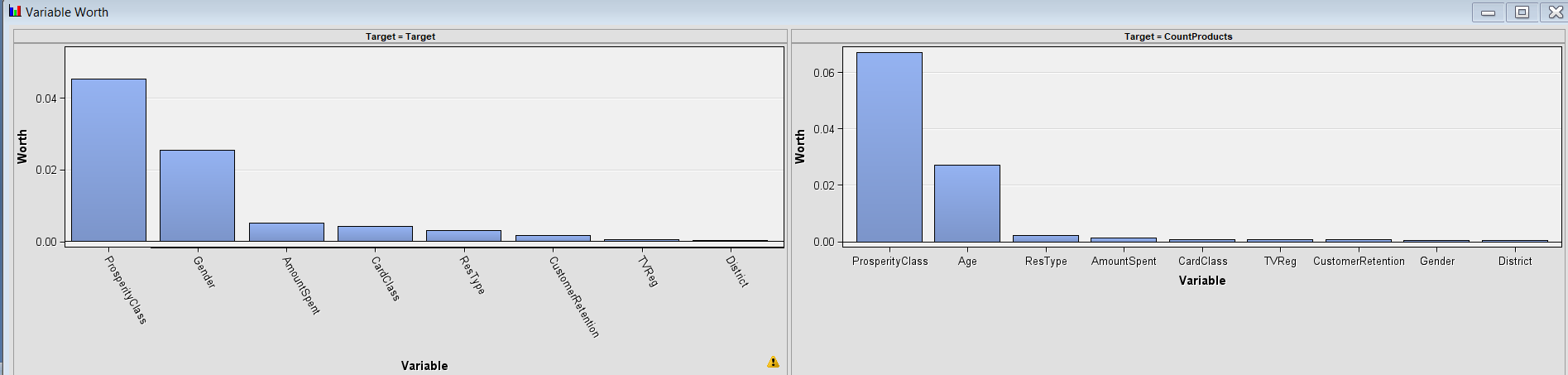


Figure 7: Variable worth between Targets and other variable

It's crucial to note that when CountProducts is rejected, the primary Target variable shows a high correlation with ProsperityClass, Age and Gender. This implies that ProsperityClass, Gender and the Age of Customers have a considerable influence on our Target.



Figure 8: Chi-Square statistics when second Target is rejected.

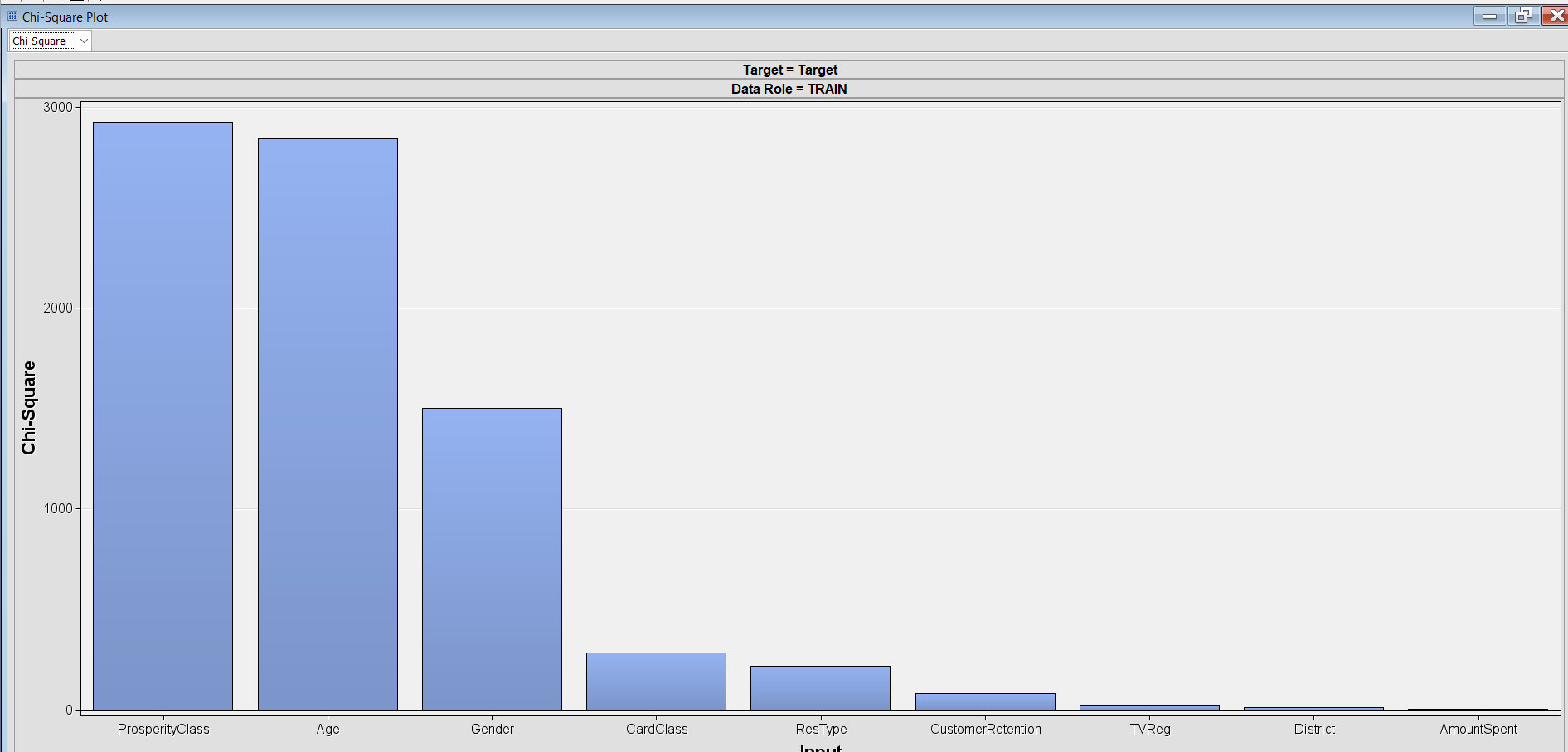


Figure 9: Correlation between Target and other variables when Second Target is rejected.

The strong correlation between the two target variables presents a compelling opportunity to utilize a multivariate approach or multitask learning. By training a single model to predict both targets simultaneously, we can significantly enhance model performance. This method allows the model to leverage shared information between the tasks, making it particularly effective. For instance, when both target variables relate to customer behavior, this joint learning approach can unlock valuable insights.

In a multivariate model, predictions for the two targets are generated in parallel, and our evaluation will focus on how accurately the model forecasts each target. We can employ a combined loss function or distinct evaluation metrics tailored to each target. By strategically analyzing our binary target variable, we can accurately identify which customers are most likely to buy the new line of products, thereby optimizing our marketing efforts and driving sales effectively.

1. **Data Partitioning:**

I need to partition the raw data into training and validation data sets to ensure the model's quality during fitting. The training data is used for preliminary model fitting, while the validation data is used to test the model empirically without overfitting. An optional test data set is used for a final assessment of the model. I use the Enterprise Miner Data Partition tool to partition the raw data. I partition 50% of the data for training and 50% for validation, but this exercise doesn't require a test data partition.

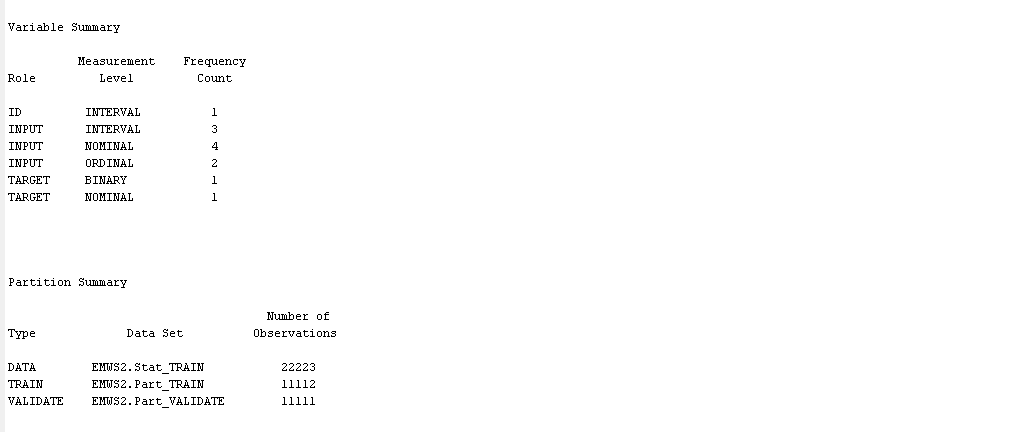


Figure 10: Data Partition Summaries

1. **Replacing Missing Values:**

I utilized a Replacement node to fill in the missing values in the input data. I did not replace any interval variables. The default setting of Standard Deviations from the Mean would enforce a range of values for each interval variable, which is not appropriate for this particular example. Therefore, I specified replacement values for the class variables in the data sets. The analysis indicates that the variable "AmountSpent" is included in the total replacement counts. It is important to note that Decision Trees are not affected by missing values. When a variable's value is missing, surrogate splitting rules can be used to choose other variable values. However, in SAS Enterprise Miner, Regression and Neural network models ignore observations containing missing values. This results in a smaller training data set, which can weaken the models' predictive power. To address this issue, missing values can be imputed before training the model.

1. **Data Imputation:**

In data mining, we often deal with missing values in datasets. To handle them, we can use the Impute node. It is a part of the SEMMA SAS data mining methodology used during the modification phase. The Impute node exports new values by creating new variables that contain replacements for the missing values. It does not overwrite the original data set but creates new variables with the imputed values. In the Impute node property, the input method is chosen as a Tree Surrogate for the Class variable and median for the Interval variable. The default input method determines the default statistic for imputing missing values. In our example, missing interval variable values are substituted with the median of the non-missing values, which is less affected by extreme values compared to the mean or midrange. This approach is particularly beneficial for imputing missing values from skewed distributions. For missing class variables, predicted values from a decision tree are employed for imputation. SAS Enterprise Miner constructs a decision tree for each class, potentially using surrogate splitting rules, with that variable as the target and the other input variables as predictors.

The data exported from the Impute node includes a new variable for each variable with missing values. The original variable remains unchanged and a new variable with "IMP\_" prefixed to the original variable's name is created. The original version of each variable also exists in the exported data with the role "Rejected". For example, variables SES and URBANICITY have values that are replaced and imputed. So, in addition to the original version, there are new versions prefaced with "IMP\_REP\_" in the exported data. This table is a subset of the SAS output from the Impute node run.

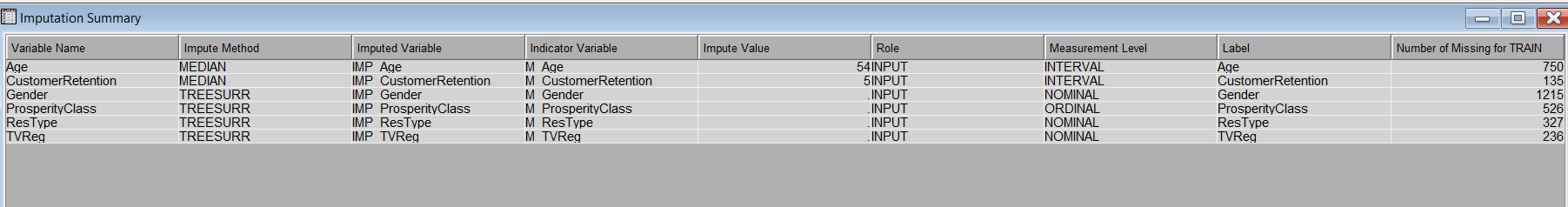


Figure 11: Imputation Summaries

1. **Simplifying Flow Diagram:**

I have used the Control Point node to simplify a process flow diagram by reducing the number of manual connections between multiple data sources and multiple model flows. Making many manual connections takes time and results in difficult-to-read process flow diagrams.

**3. Predictive Modeling:**

In my research, I developed thirteen predictive models, Five non-parametric and Eight parametric. These models were created to determine the best individuals in a database to target for a new product range. I used Decision Tree algorithms as non-parametric models and Regression, Neural Network, Support Vector Machine (SVM), and Clustering algorithms as parametric models.

Decision tree methodology is a popular data mining method that creates a classification system based on multiple covariates or develops a prediction algorithm for a target variable. This method classifies a population into branch-like segments, constructing an inverted tree with a root node, internal nodes, and leaf nodes. The non-parametric algorithm can efficiently handle large, complicated datasets without imposing a complex parametric structure.

When the sample size is large enough, the study data can be divided into training and validation datasets. I used the training dataset to build a decision tree model and the validation dataset to determine the appropriate tree size needed to achieve the optimal final model.

Regression analysis is a predictive modeling technique that estimates the relationship between two or more variables. This method focuses on the relationship between a dependent (target) variable and independent variable(s) (predictors). Here, the dependent variable is assumed to result from the independent variable(s). The value of predictors is used to estimate or predict the likely value of the target variable.

Another parametric model I used in my research was the Neural Network. This model imitates brain functions to identify patterns in historical or new, incomplete data sets, allowing for intelligent future predictions and informed decision-making to meet goals.

Support Vector Machine (SVM) is a machine learning technique that separates the attribute space with a hyperplane, maximizing the margin between the instances of different classes or class values. This method often yields superior predictive performance results.

Cluster analysis is a method of grouping objects based on their similarity, such that objects within the same cluster are more similar to each other than to objects in another cluster. Clusters can be determined through various criteria such as smallest distances, density of data points, graphs, or statistical distributions. Data scientists and other professionals use clustering to extract essential insights from data by observing the groups (or clusters) the data points belong to when a clustering algorithm is applied to the data. Unsupervised learning is a type of machine learning that searches for patterns in a dataset without pre-existing labels and minimal human intervention. Clustering can also be used for anomaly detection to identify data points that are not part of any cluster or outliers.

Clustering is typically used to identify groups of similar objects in datasets containing two or more variable quantities. This data can be collected from a variety of sources such as marketing, biomedical, or geospatial databases. There are two main types of clustering algorithms, K-Means and hierarchical clustering. K-Means identifies clusters by minimizing the mean distance between geometric points, while hierarchical clustering groups data into a multilevel hierarchy tree of related graphs, starting from the finest level (original) and proceeding to the coarsest level.

To select the best model, I compared the statistical performance of the thirteen competing models using an evaluation criterion. Then, I used the performance measure with the validation data to choose the champion model.

1. **Decision Tree:**

The decision tree models have several advantages. They are easy to understand conceptually and can effectively handle nonlinear relationships between input variables and target variables. Furthermore, they are able to manage missing values without requiring imputation. As a result, I have decided to begin by modeling the data using decision trees, and I will later compare these models to others in the example.

In SAS Enterprise Miner, there are two approaches to building a decision tree: automatic and interactive. To begin, opt for the automatic training and pruning of a tree by selecting the Decision Tree node. In the Properties Panel, navigate to the Train properties and set the Maximum Depth splitting rule property to 10. This allows SAS Enterprise Miner to train a tree with up to ten generations of the root node, with the final tree having fewer generations due to pruning. Set the Leaf Size node property to 8 to establish the minimum number of training observations in any leaf, and the Number of Surrogate Rules node property to 4 to allow the use of up to four surrogate rules in each non-leaf node if the main splitting rule relies on an input with a missing value. The Assessment Measure subtree property is automatically set to Decision if you've defined a profit matrix in Create a Data Source. As a result, the Decision Tree node will create a tree that maximizes profit in the validation data. This is called Default Decision Tree.

The Node Rules window shows the IF-THEN logic used to assign observations to leaf nodes in the decision tree. There are six leaf nodes in this tree. Each leaf node provides the following information: node number, number of training observations in the node, percentage of training observations in the node with TARGET\_B=1 (predicted), adjusted for prior probabilities, and percentage of training observations in the node with TARGET\_B=0 (did not predict), adjusted for prior probabilities. The tree has been automatically pruned to an optimal size, so the node numbers in the final tree are not sequential and reflect the positions of the nodes in the full tree before pruning.

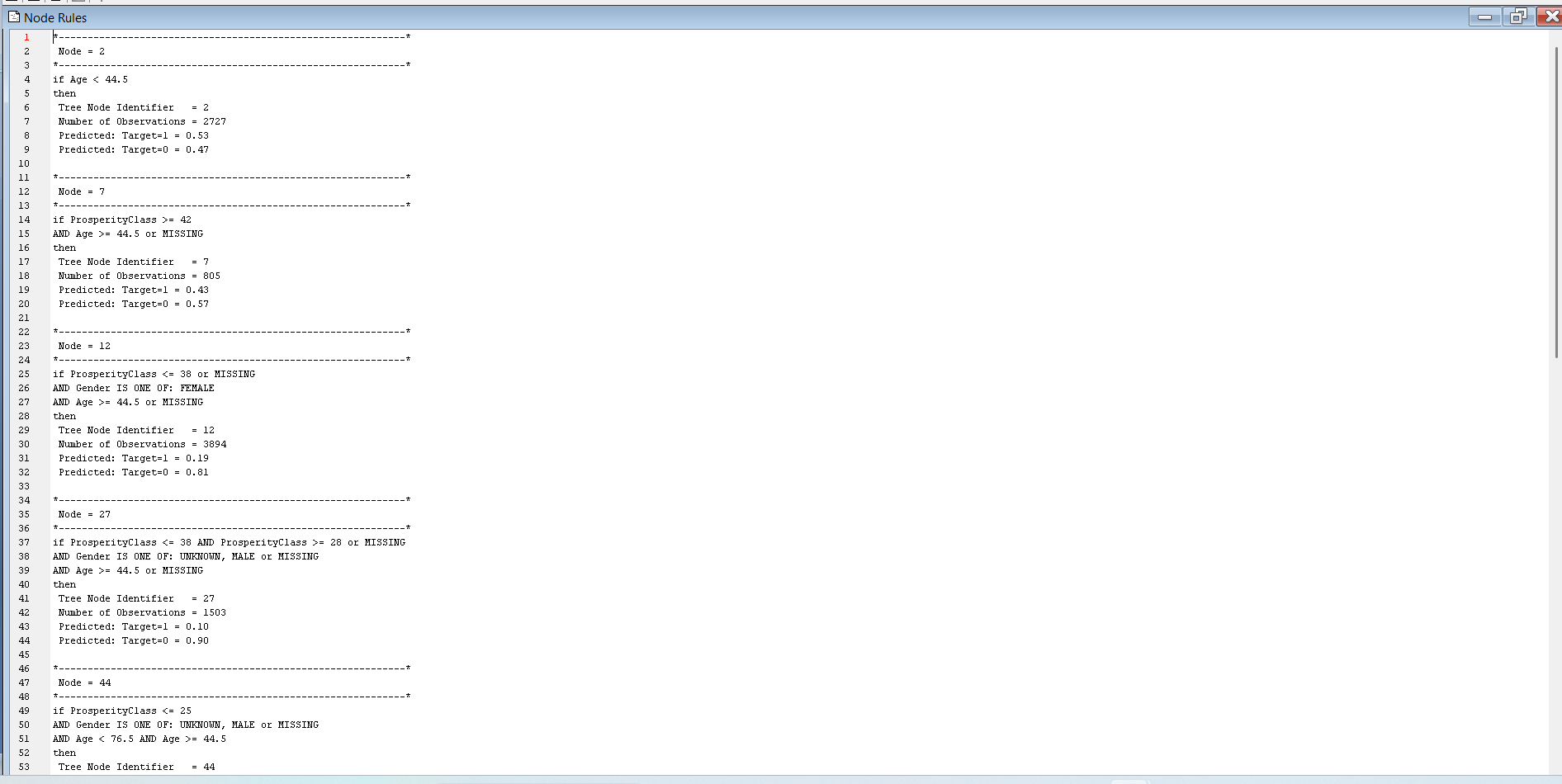


Figure 12: Node rules results of Default Tree

The subtree assessment plot reveals a positive trend, with a decrease in the misclassification rate for both training and validation datasets. This suggests that the Default tree model is well-suited for predicting the outcomes of the response variable. Additionally, the Cumulative Chart highlights an early declining trend, which could be a poor sign for the model's performance.



Figure 13: Misclassification rate of Default Tree

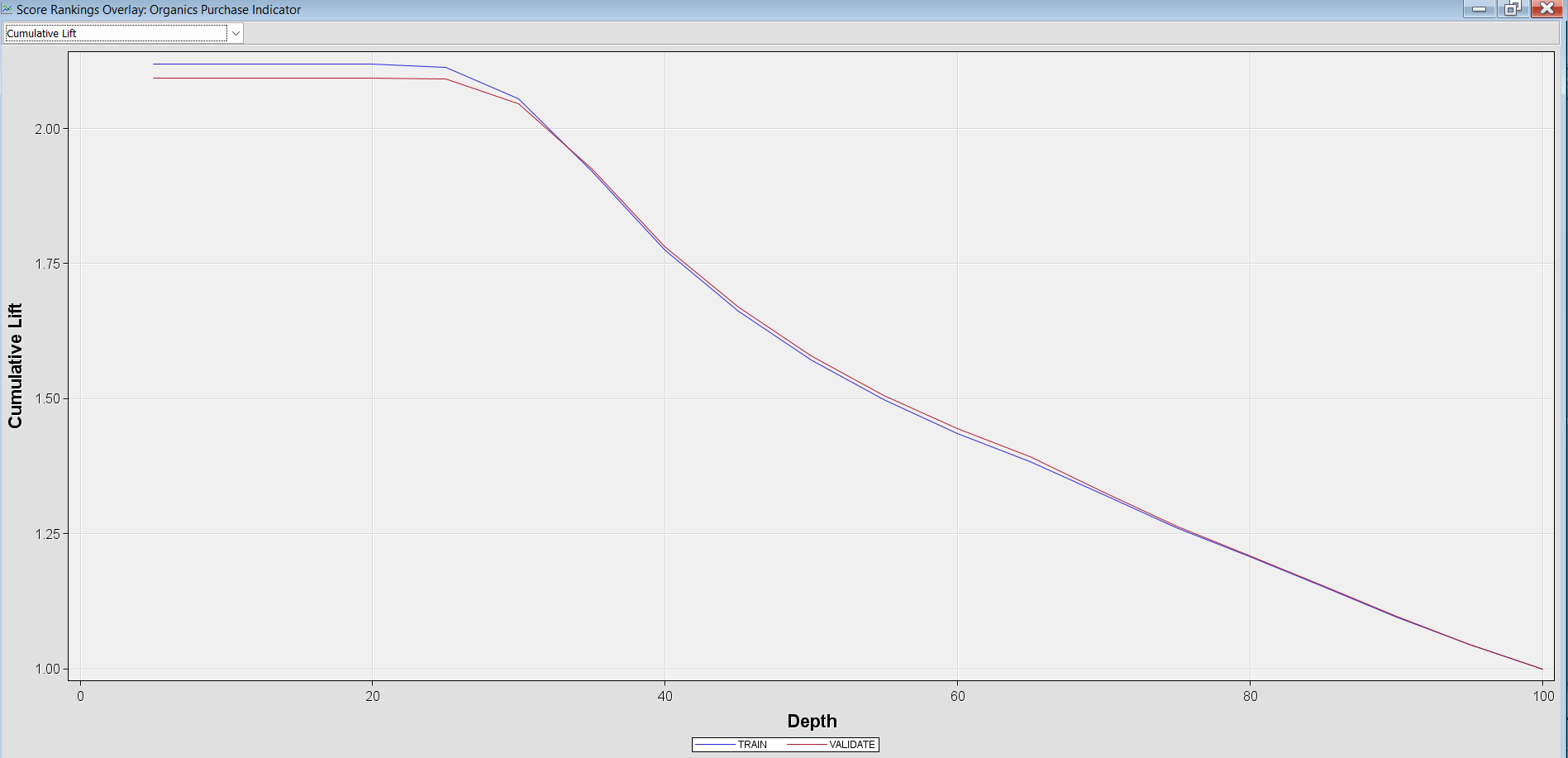


Figure 14: Cumulative Lift for Default Tree

I need a second decision tree that can be used as an interactive decision tree to perform interactive tree splits. The window for split nodes will list the candidate splitting rules sorted according to their log-worth values. The nodes can be split by choosing the highest log values.

The Split Node window opens and lists the candidate splitting rules ranked by logworth (-Log(p)). The Age rule has the highest logworth. The nodes are colored from dark to light, corresponding to low to high percentages of correctly classified observations. The average profit with Target = 1 is 3.24 for both training and validation. The total count for the training set is 11112. The validation accuracy achieved is 75.23%, indicating a solid level of performance. Additionally, the training and validation percentages for Target=1 are both at 0.25, providing a clear basis for further analysis and improvement.

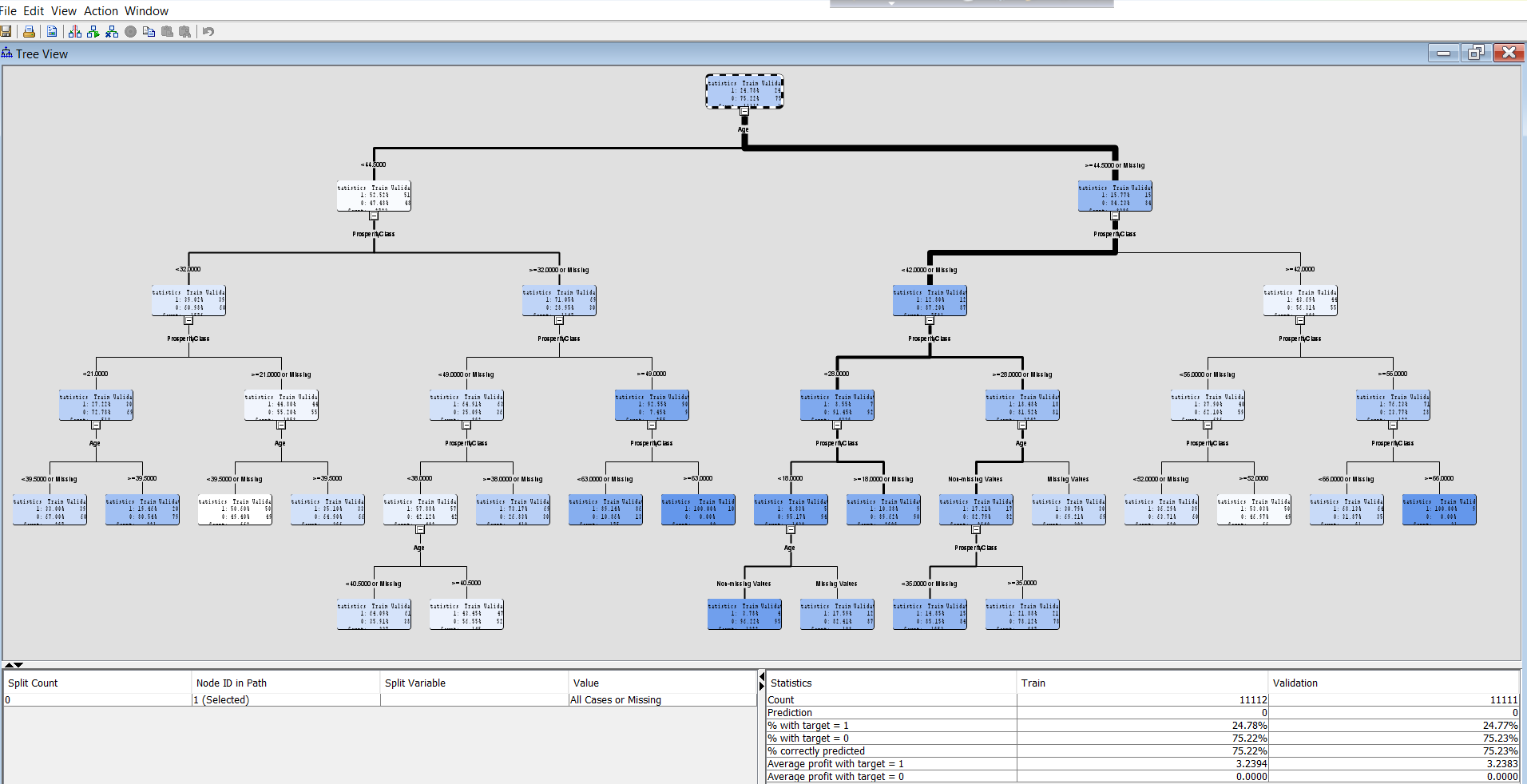


Figure 15: Tree view and statistics results of Interactive Decision Tree

The validation misclassification rate is an impressive 0.20 with 19 leaves, highlighting the model's effectiveness. Furthermore, the cumulative lift chart clearly demonstrates a substantial lift at depth 5, underscoring the model's strong predictive capabilities.

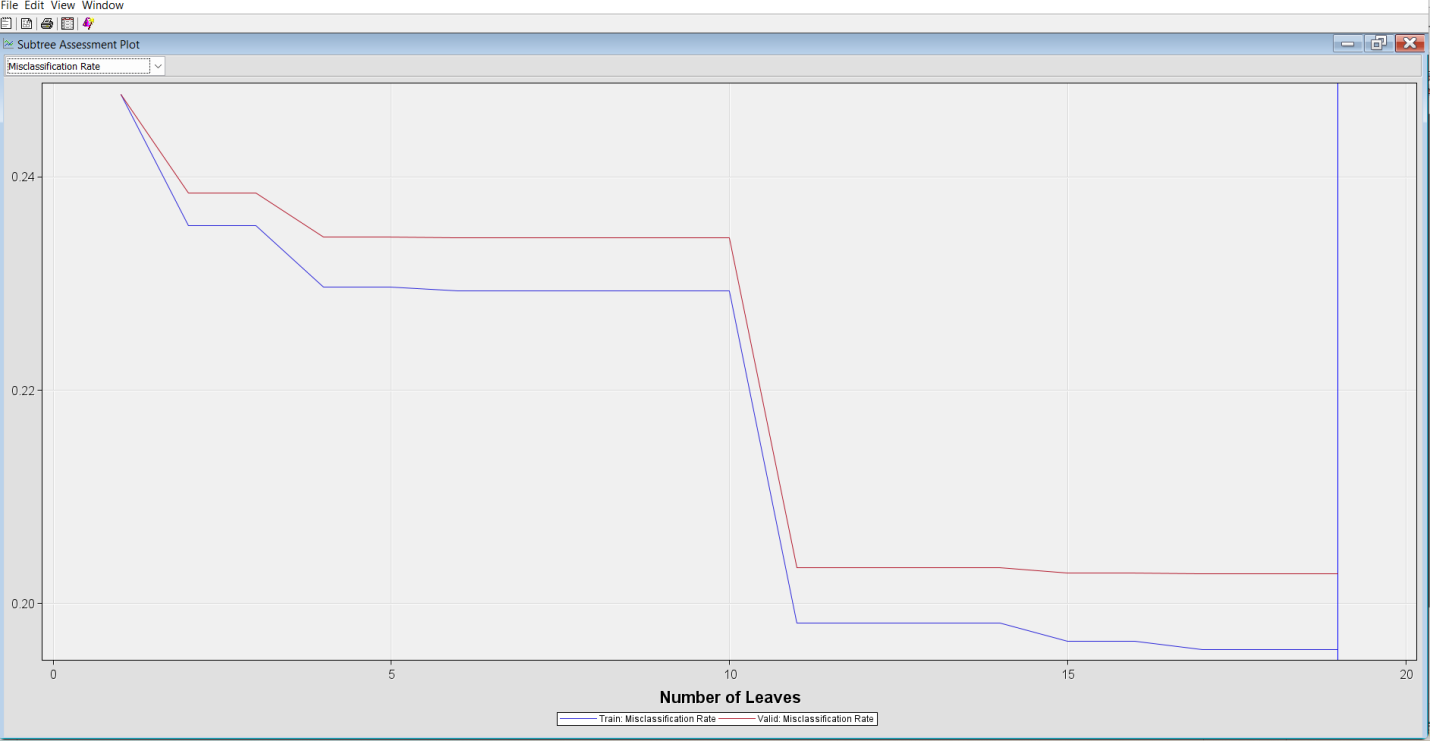


Figure 16: Misclassification rate of Interactive Decision Tree

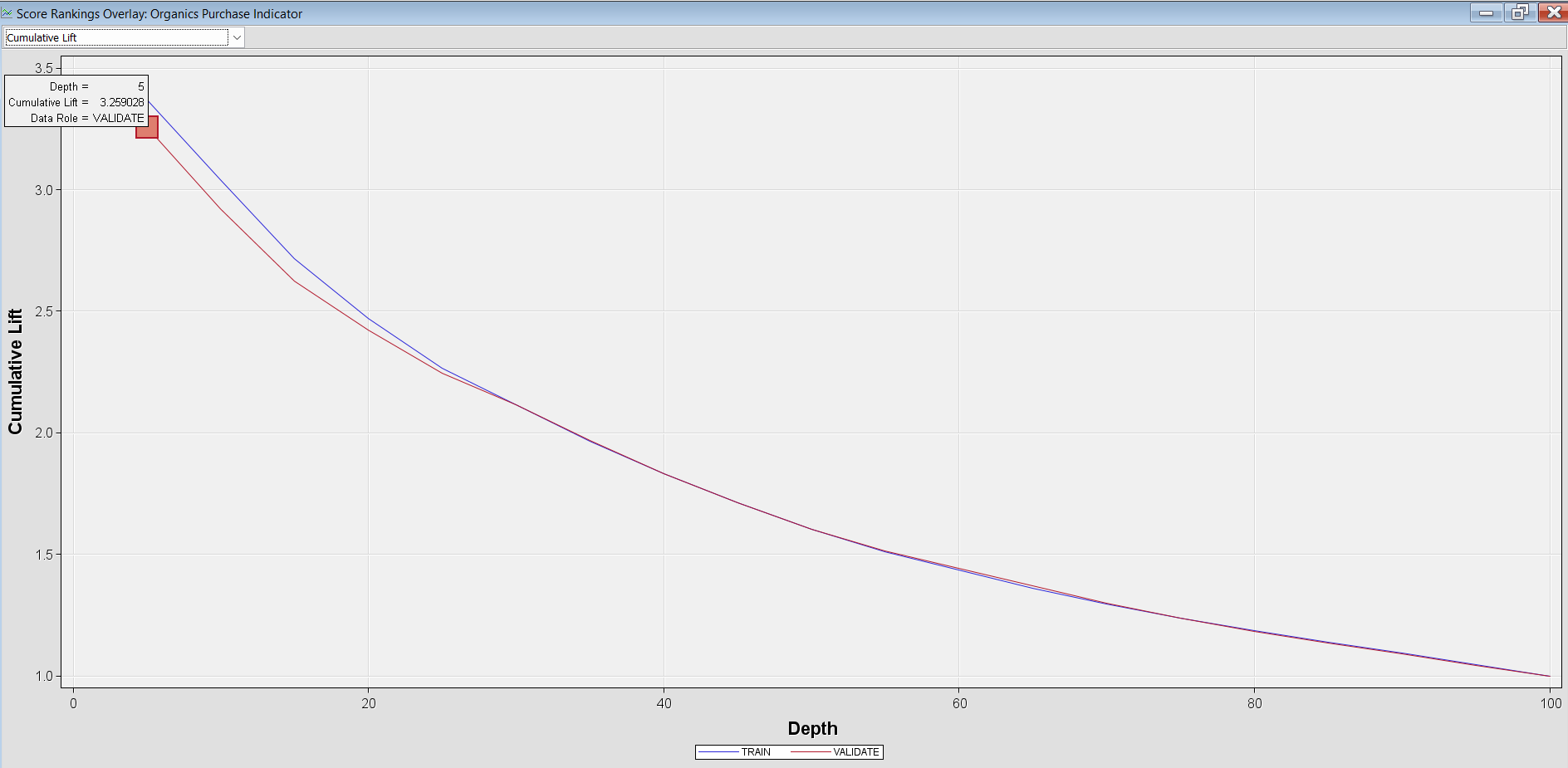


Figure 17: Cumulative lift chart of Interactive Decision Trees.

"I have created a new decision tree called the Maximal Tree, based on another decision tree. The Maximal Tree has a leaf size of 15, and we used the Largest subtree method to generate it. This method allows for an autonomous way to create the largest possible tree. By building the Maximal Tree, we can achieve optimal profits."

In SAS Enterprise Miner, a maximal tree is created using the training dataset during the decision tree modeling process. This tree is complex, aiming to fit the training data closely, potentially resulting in overfitting. It includes all possible splits and branches, leading to numerous leaves and nodes.

Key Aspects:

- Creation Process: The maximal tree is built by evaluating potential splits to determine the best variable and split point using algorithms like chi-square tests.

- Overfitting: A major concern with maximal trees is overfitting, which can reduce performance on unseen data. Pruning techniques are used to simplify the model and improve its generalization ability.

- Pruning: After constructing the maximal tree, pruning is used to find an optimal subtree that balances complexity and predictive accuracy by evaluating the tree against validation data.

The Misclassification Plot demonstrates results that closely align with the Interactive Decision Tree, suggesting consistency in our analysis. Importantly, we observe a high Cumulative Lift at depth 5, indicating a solid foundation for model performance. Additionally, the gradual decline in Lift provides a valuable insight into the model's steady effectiveness. These findings position us well for further enhancements and improvements in our modeling efforts.

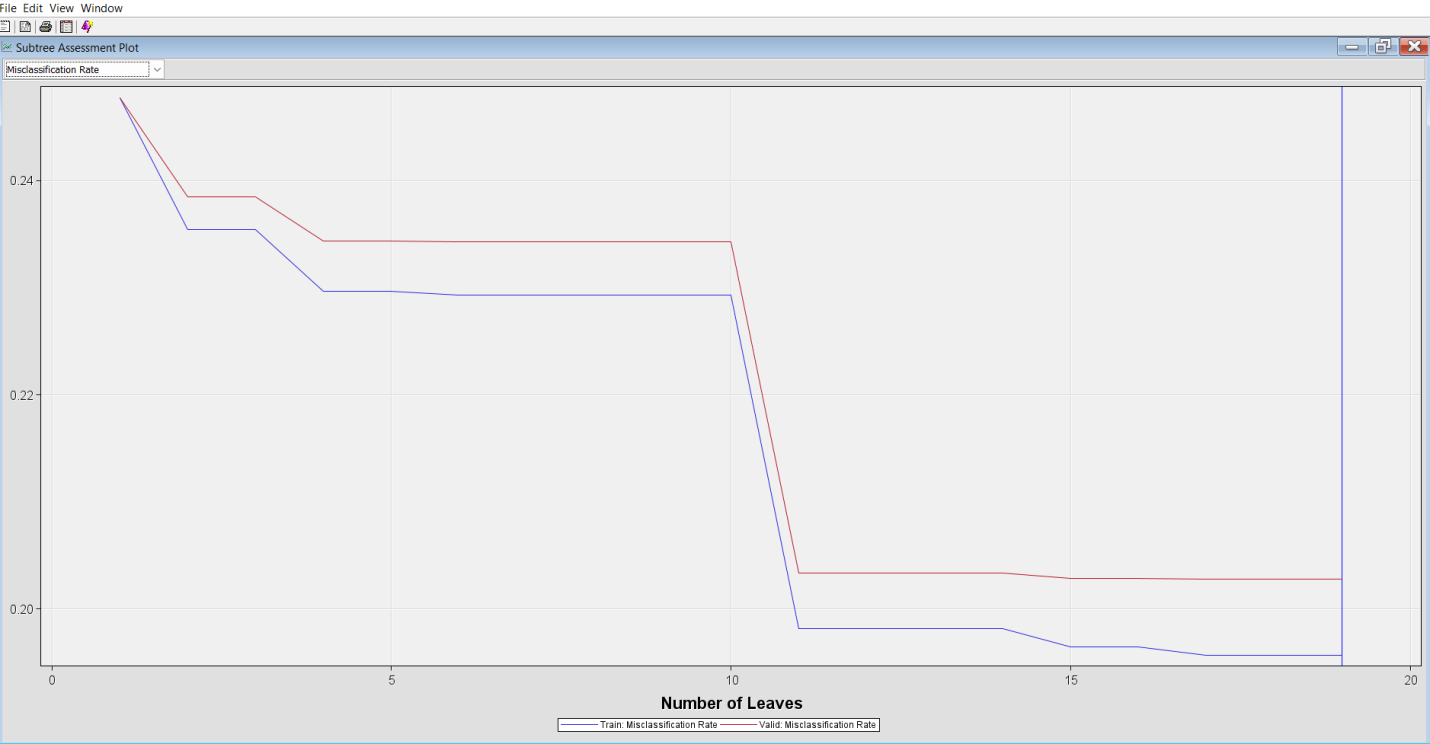


Figure 18: Misclassification rate of Maximal Tree

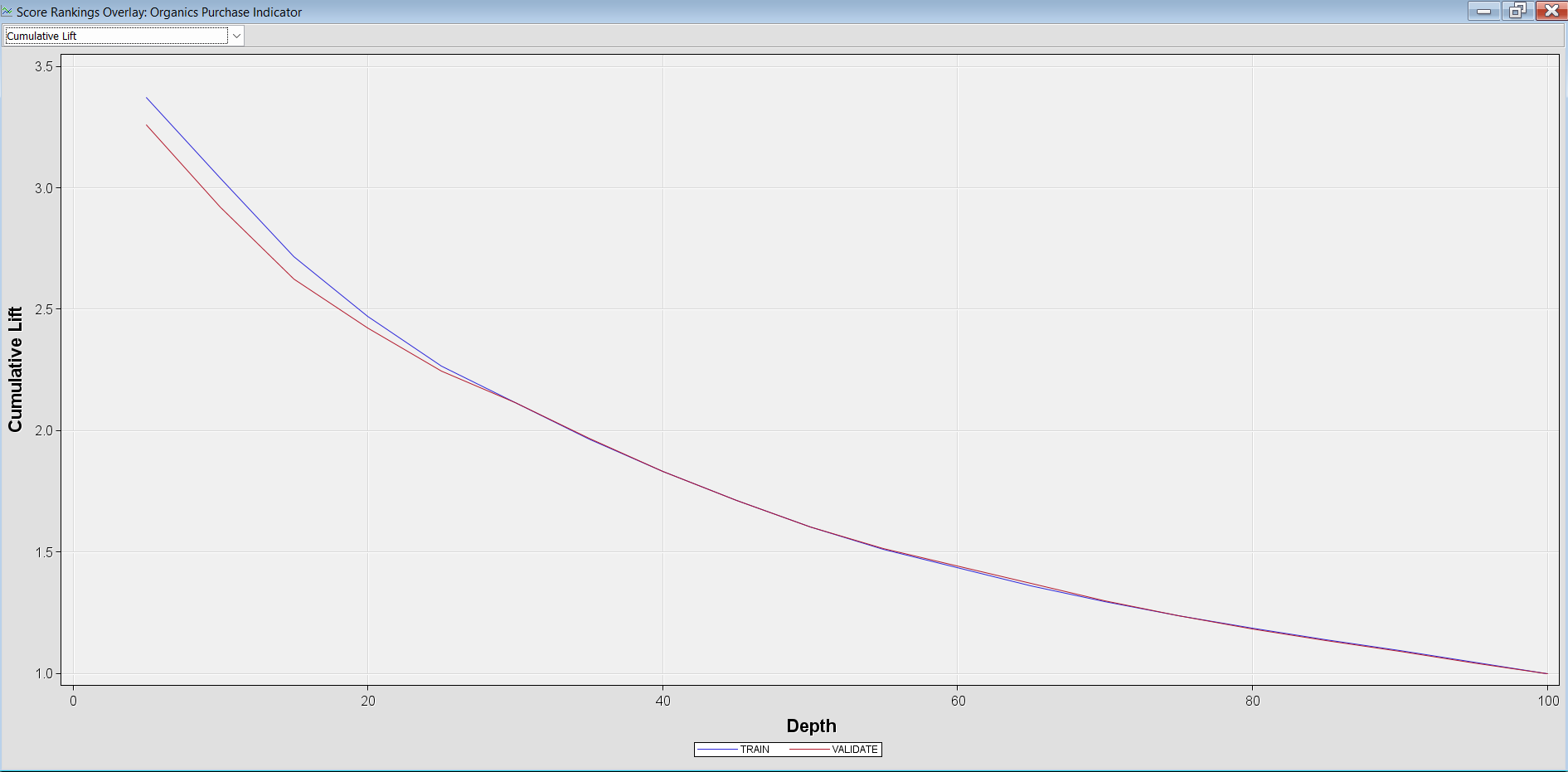


Figure 19: Cumulative Lift chart of Maximal Tree

By configuring the assessment measure property to Average Square Error and Misclassification, we can also build two Decision Tree, now known as a Probability Tree and Optimal Tree, respectively.

In SAS Enterprise Miner, a probability tree is a specialized type of decision tree designed to predict probabilities instead of simple categorical outcomes. It achieves this through a process known as pruning, which minimizes the Average Squared Error (ASE) to produce more accurate probability estimates for different outcomes. This makes it particularly valuable in situations where understanding the probability of various outcomes is essential.

Key Characteristics of Probability Trees:

Optimization: The probability tree is pruned to minimize the Average Squared Error, resulting in more accurate probability estimates for different branches of the tree.

Comparison with Other Trees: In the context of decision trees, there are typically three types:

1. Maximal Tree: This is the initial tree grown without any pruning, risking overfitting and poor generalization of unseen data.

2. Pruned Tree: This tree is crafted by removing branches from the maximal tree to enhance its performance on a validation dataset and optimize for a specific assessment measure.

In summary, probability trees in SAS Enterprise Miner serve as a powerful tool for generating precise probabilistic predictions, significantly enhancing the interpretability and utility of decision trees in data analysis.

The misclassification results and cumulative lift reveal a compelling trend that strongly aligns with the Maximal Tree, underscoring its significance in our analysis.

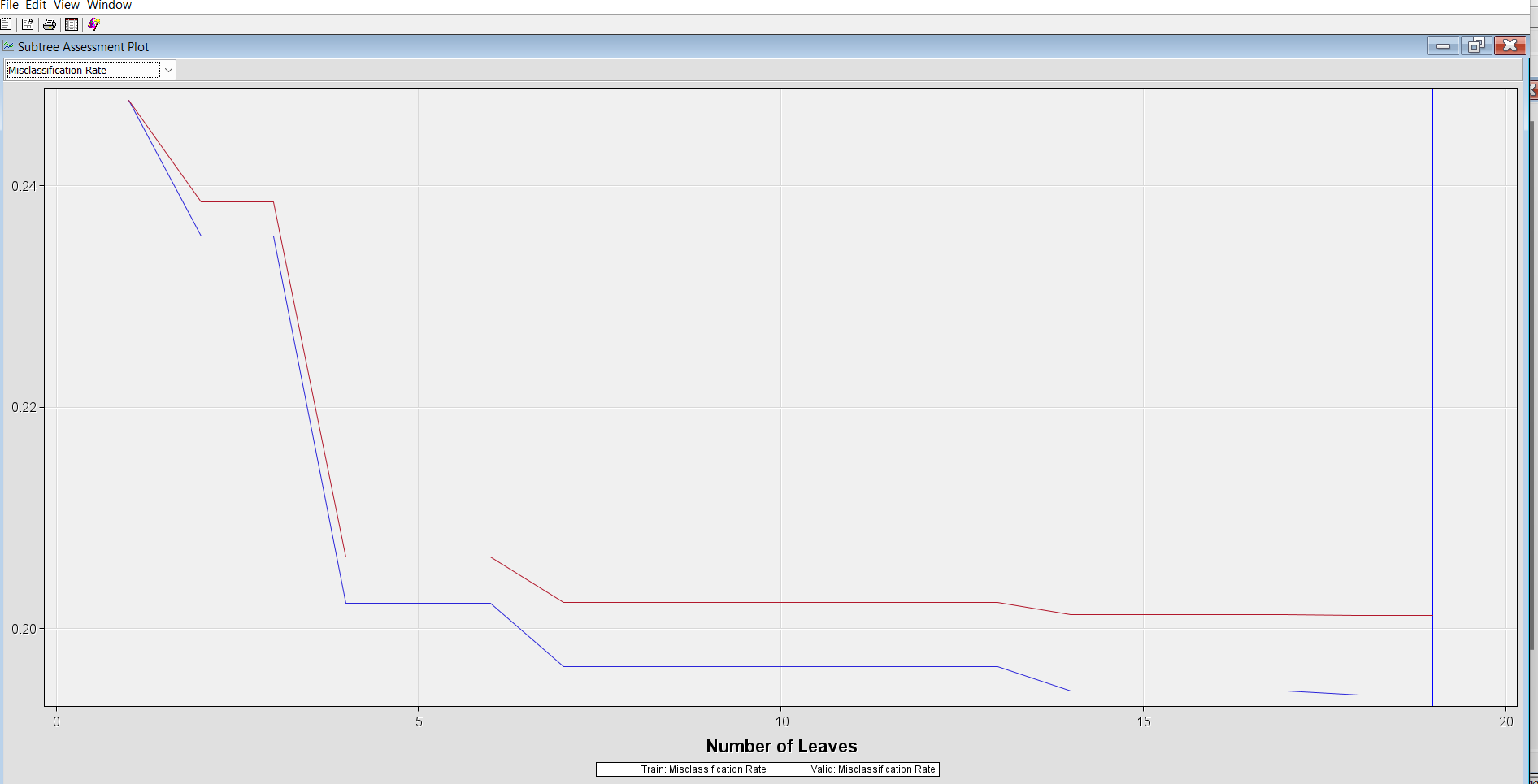


Figure 20: Misclassification rate of Probability Tree

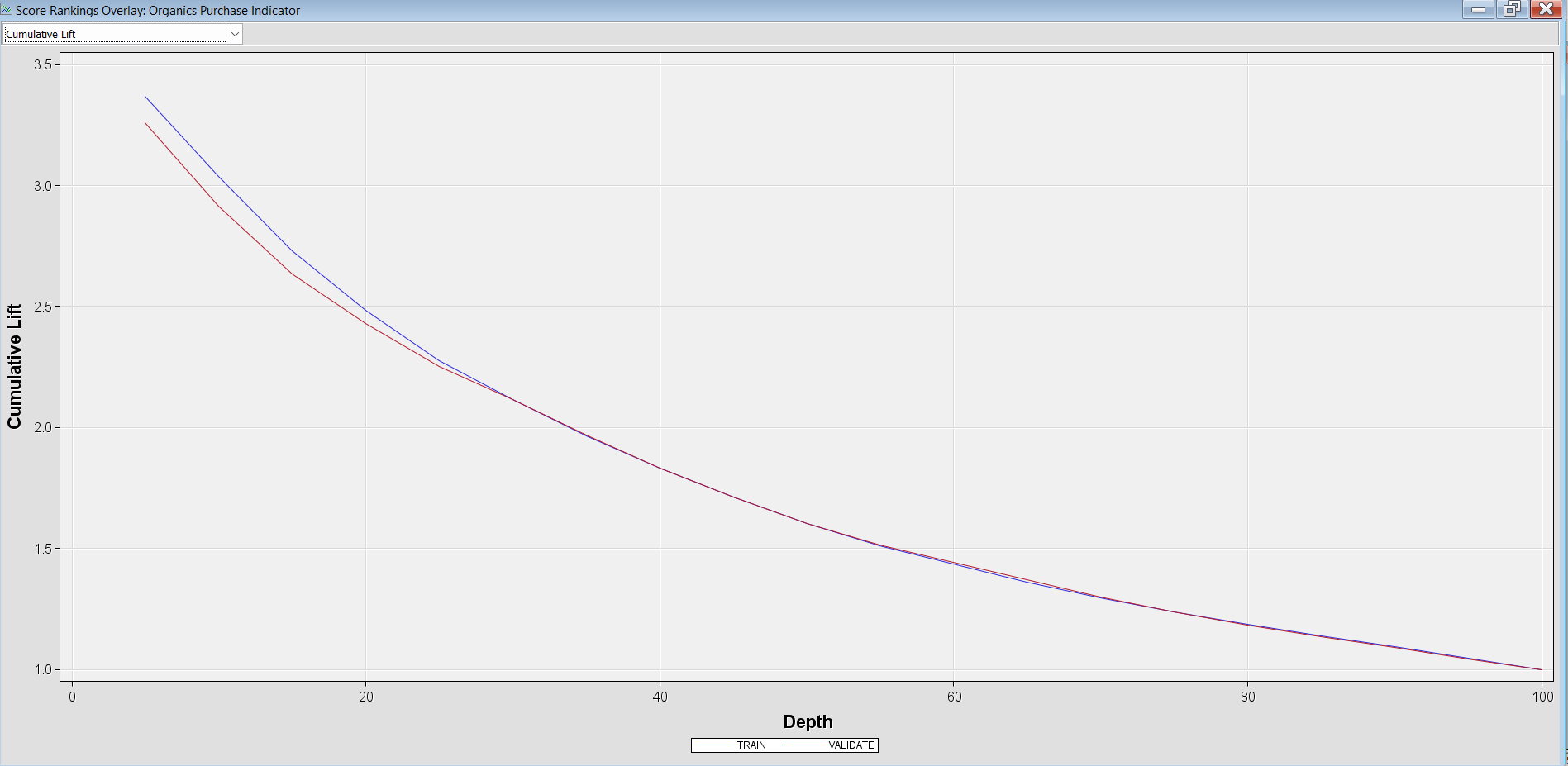


Figure 21: Cumulative lifts Chart for Probability Tree

The optimal tree in SAS Enterprise Miner is selected based on its performance on a validation set, avoiding overfitting and ensuring generalization to new data. Here are the key aspects:

- Training vs. Validation Sets: The training set creates a maximal tree, while the validation set prunes and improves its performance.

- Pruning Techniques: SAS Enterprise Miner simplifies the model while maintaining predictive power, offering options based on different assessment measures.

- Assessment Measures: Various assessment measures can be used, such as misclassification rate, average squared error, and lift.

- Modeling Flexibility: Decision trees in SAS Enterprise Miner handle both classification and regression tasks and can accommodate non-linear relationships.

In summary, the process involves training on a maximal tree, then validating and pruning it based on performance metrics to ensure an efficient and effective final model for making predictions.

We observe a notably reduced misclassification rate for the optimal tree count of 9 leaves, highlighting its effectiveness. Furthermore, the cumulative lift is impressively high at a depth of 5, underscoring the model's strong predictive power.

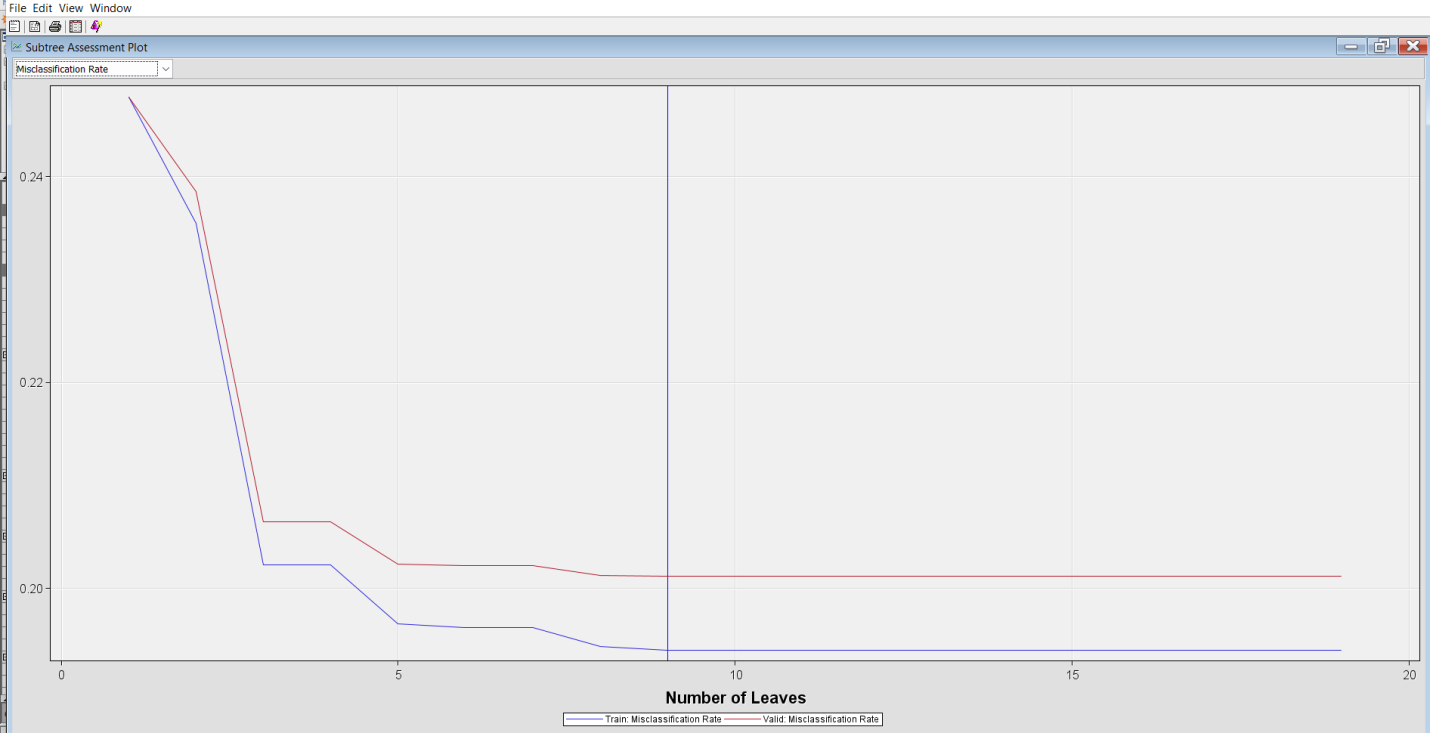


Figure 22: Misclassification rate of Optimal Tree

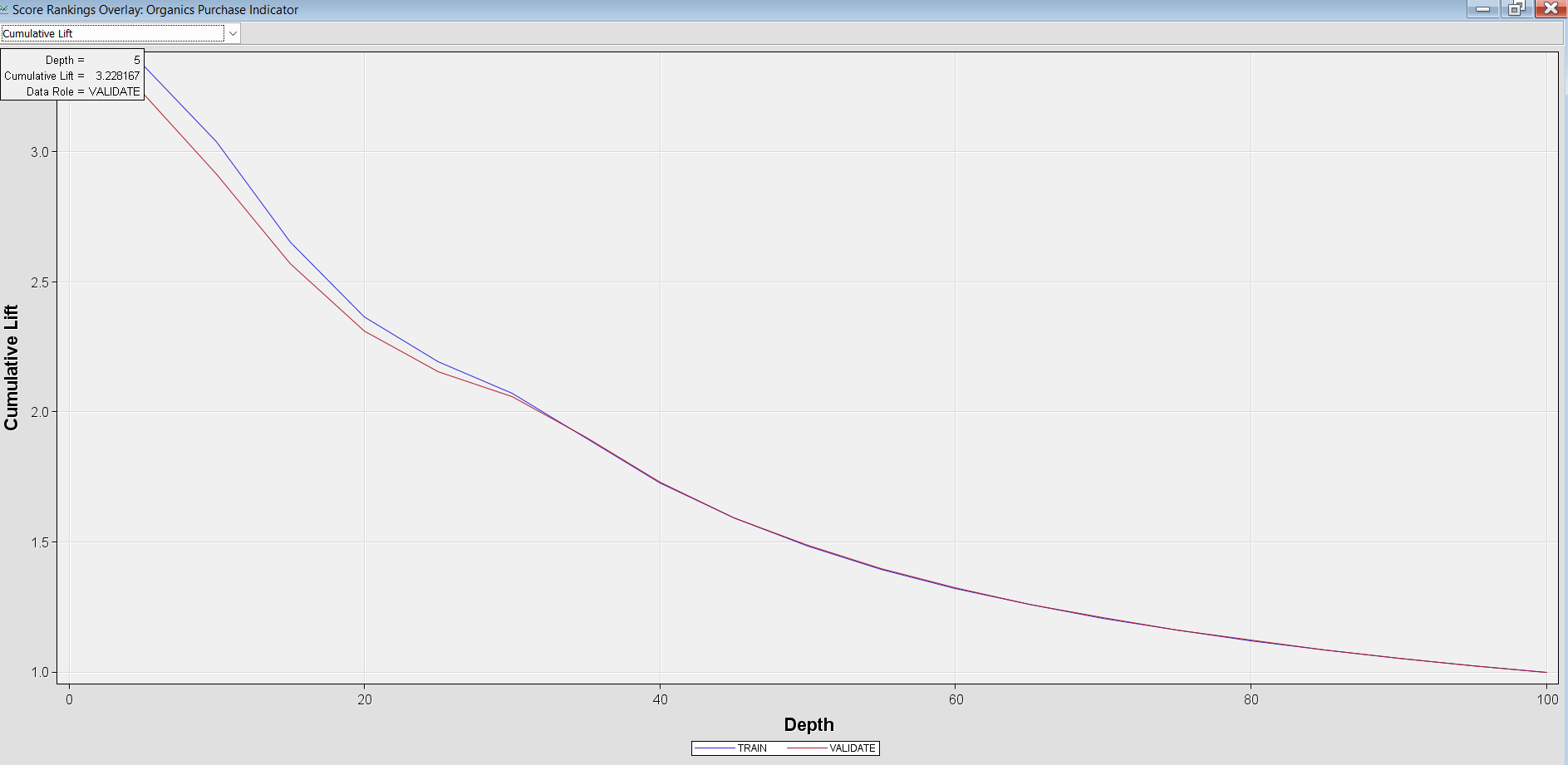


Figure 23: Cumulative lift for Optimal Tree

In our comprehensive analysis of the Decision Tree outputs, we observed that the Default Decision Tree and the Interactive Decision Tree exhibit remarkably similar leaf nodes. Furthermore, the Maximal Tree and the Optimal Tree present nearly identical leaf nodes as well. In contrast, the Probability Tree's leaf node stands out as distinctly different from the others.

Upon careful evaluation, we found that the Default Decision Tree generates the highest average profit for the target at 3.26720. However, it also has a higher valid misclassification rate of 0.23850. The Optimal Tree, while yielding a slightly lower profit of 3.24, shines with the lowest valid misclassification rate of 0.20115 for leaf 9, outperforming all others in this regard. The Probability Tree offers an average profit of 3.22953, coupled with a misclassification rate and average square error that are more favorable than those of the other trees. Additionally, its ROC curve is favorably positioned near the upper left corner, indicating strong predictive power.

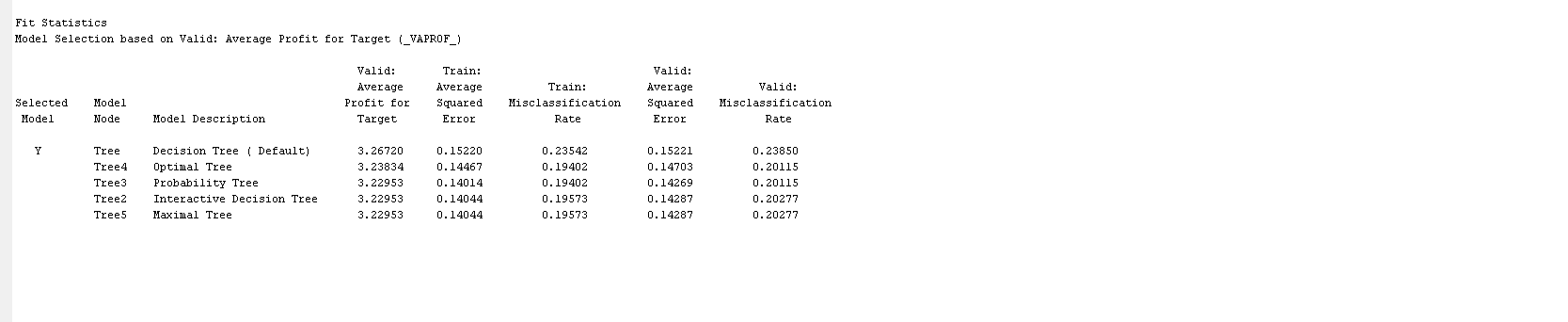
While the Probability Tree may have a marginally lower profit, its superior accuracy and reliability make it the most compelling choice for modeling. We strongly advocate for its selection in our analysis.

Figure 24: Fit statistics results of All Decision Trees

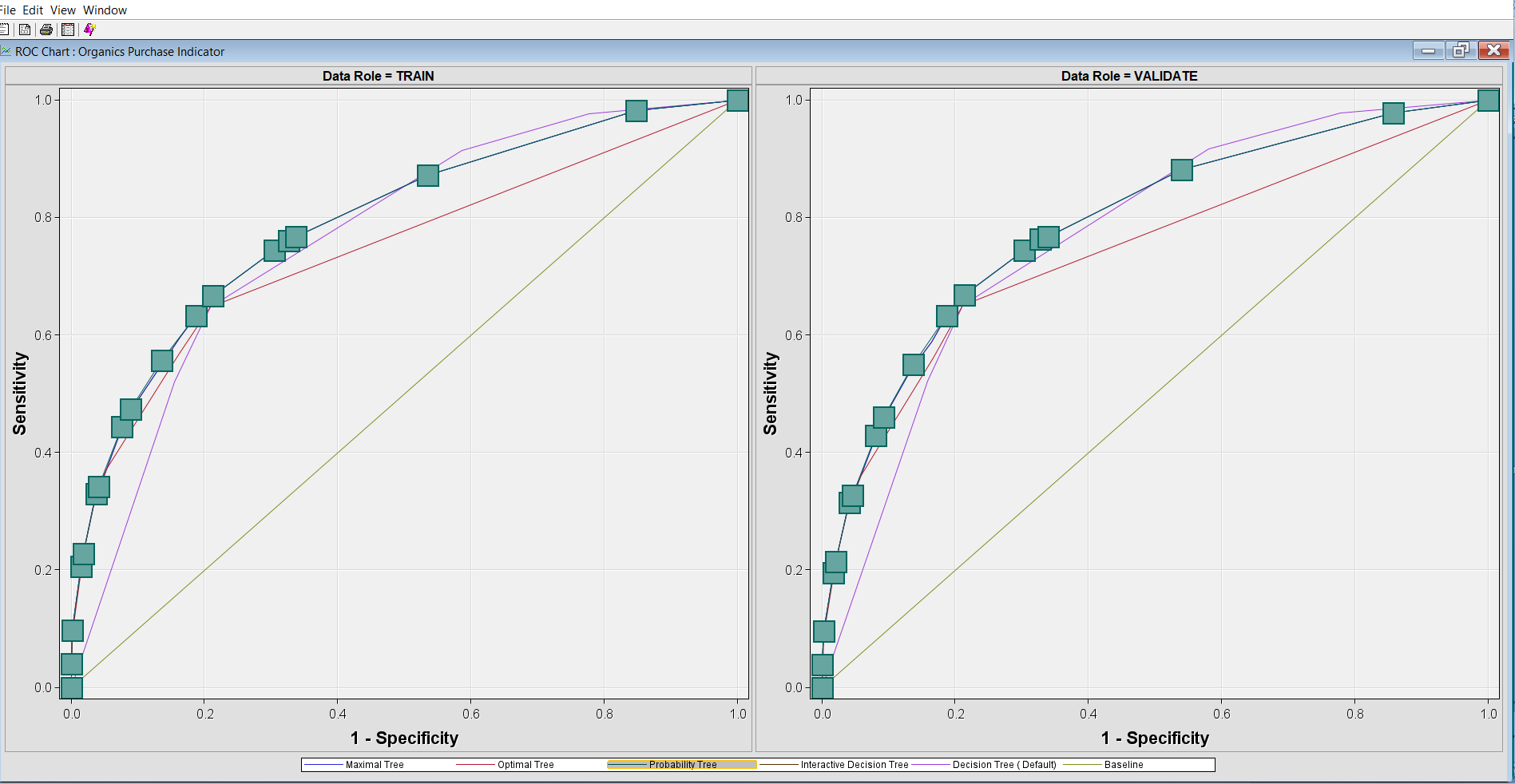


Figure 25: ROC curve of all Decision Trees model.

1. **Regression:**

The Regression tool in SAS Enterprise Miner supports linear regression for interval target variables and logistic regression for categorical target variables. It offers model selection methods like forward, backward, and stepwise selection. Before running regression models, it's essential to prepare the data by splitting it into training and validation sets. We can customize the stepwise process and compare regression models with other data mining models. Overall, regression in SAS Enterprise Miner is a powerful tool for analyzing and predicting outcomes.

1. **Data Transformation:**

It's recommended to preprocess the input data before feeding it into the Regression and Neural Network modeling nodes. Data transformation can help to stabilize the variance, eliminate nonlinearity, enhance additivity, and address non-normality issues. For many models, transforming the input data can result in better model fits. This transformation can involve one or more variables. The standard log transformation is commonly used to address skewness. When transforming the data variables, I focused on all Interval input variables and examined their distribution. I noticed that both AmountSpent and IMP\_CustomerRetention have skewed distributions. I applied the common log formula to normalize these two attributes. Below are the results showing the skewed variables.

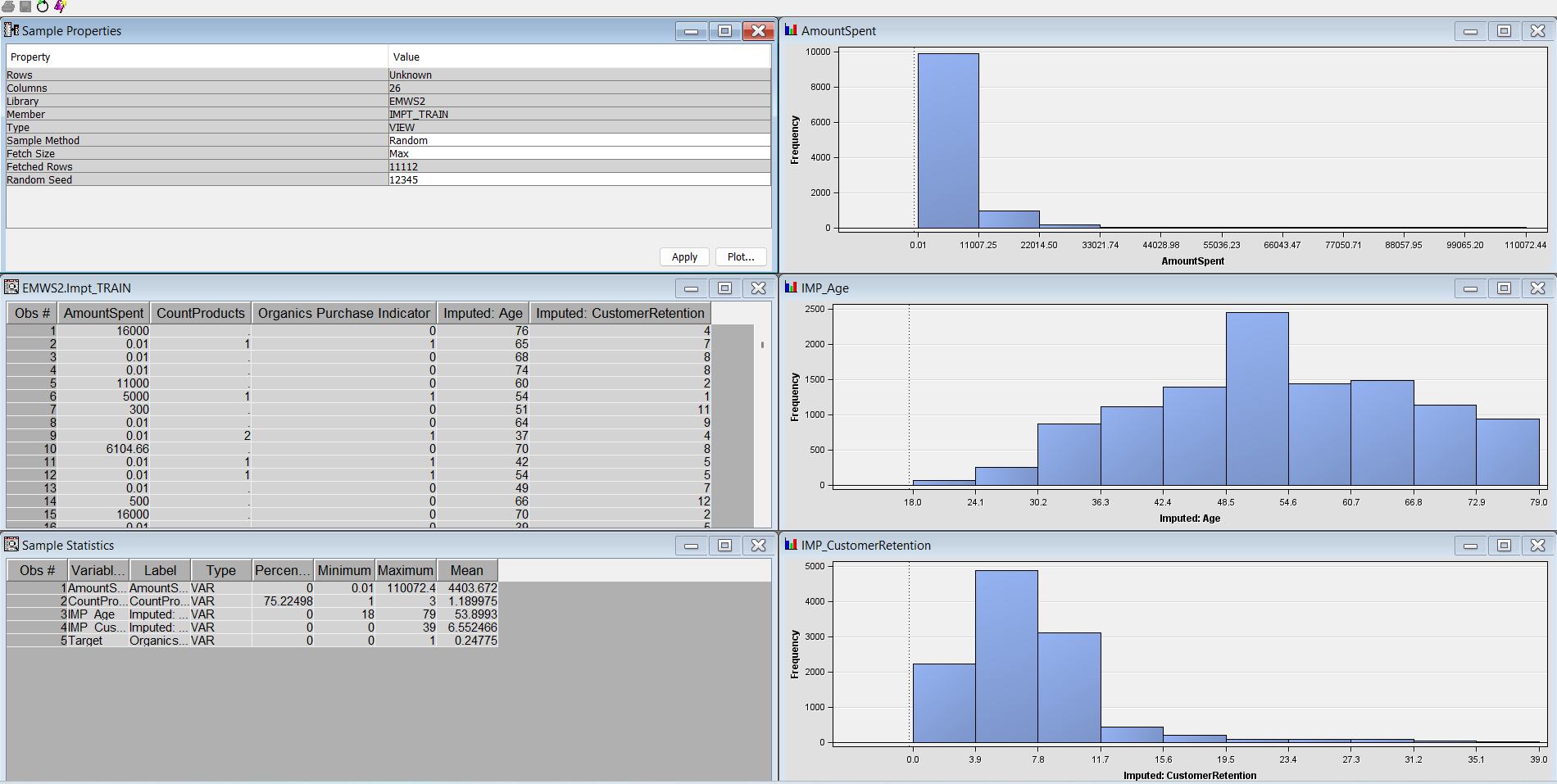


Figure 26: Sample properties

When exporting data from the Transform Variables node, a new variable is generated for each transformed variable, without overwriting the original. The new variable retains the original name but is prefixed with an identifier representing the transformation. For example, variables that have undergone a log transformation are prefixed with LOG\_, while those transformed using optimal binning are prefixed with OPT\_. Additionally, the original version of each variable is preserved in the exported data under the role of Rejected.

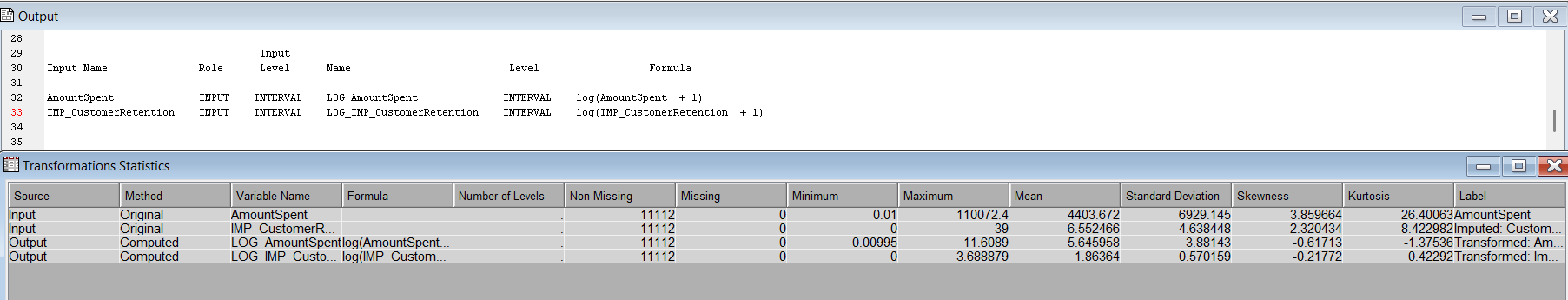


Figure 27: Transformation statistics

We can use a Regression node connected with the first control point to analyze the input variables. To highlight the corresponding observations from a histogram in the Explore window, we can select a bar in that histogram. This will highlight the corresponding observations in the EMWS1.Impt\_TRAIN data set window and other histograms. We can use the Stepwise Regression model to build a logistic regression model by selecting it from the Selection Model. The Regression node automatically performs logistic regression if the target variable is a class variable with a binary outcome. For continuous targets, the Regression node performs linear regression by default.

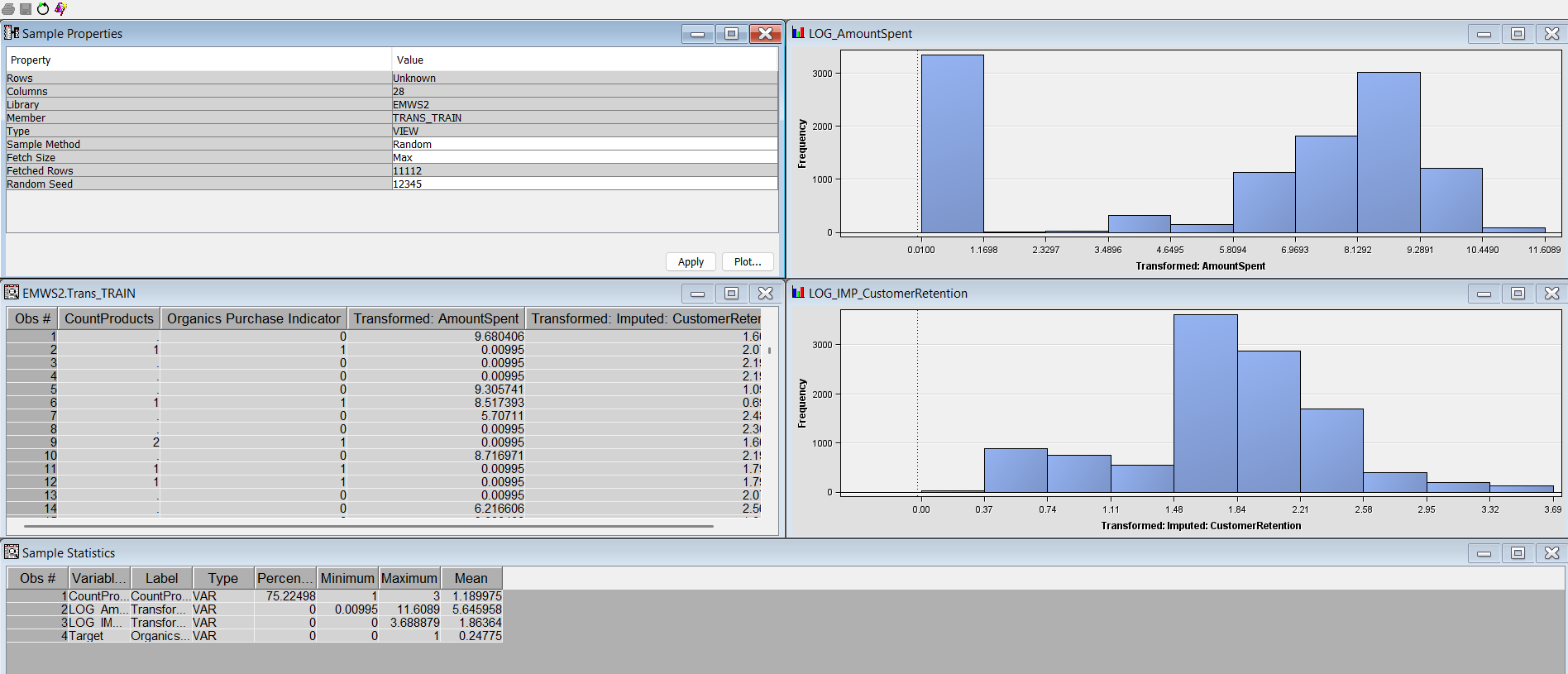


Figure 28: Emws2.Impt\_Train results of Default Regression.

The Output window in the Regression Results browser displays the stepwise variable selection process, including the fit statistics for each iteration. When analyzing the Cumulative Lift Curve, we can see that at a depth of 5, the cumulative lift is 3.55, and at a depth of 100, the cumulative lift is 1 for the training data. This demonstrates the percentage of positive responses relative to the percentage of customers contacted. The P-values for IMP\_Age, IMP\_Gender, IMP\_ProsperityClass, and M\_Gender, M\_ ProsperityClass are all below 0.05, highlighting their effectiveness as strong predictors.

The Fit statistics for the Regression model reveal compelling results: an impressive average profit of 3.25 during the validation phase, coupled with a remarkably low average squared error of 0.14. Furthermore, the RMSE stands at just 0.37, underscoring the model's accuracy and reliability. These figures highlight the model's effectiveness in delivering valuable insights.

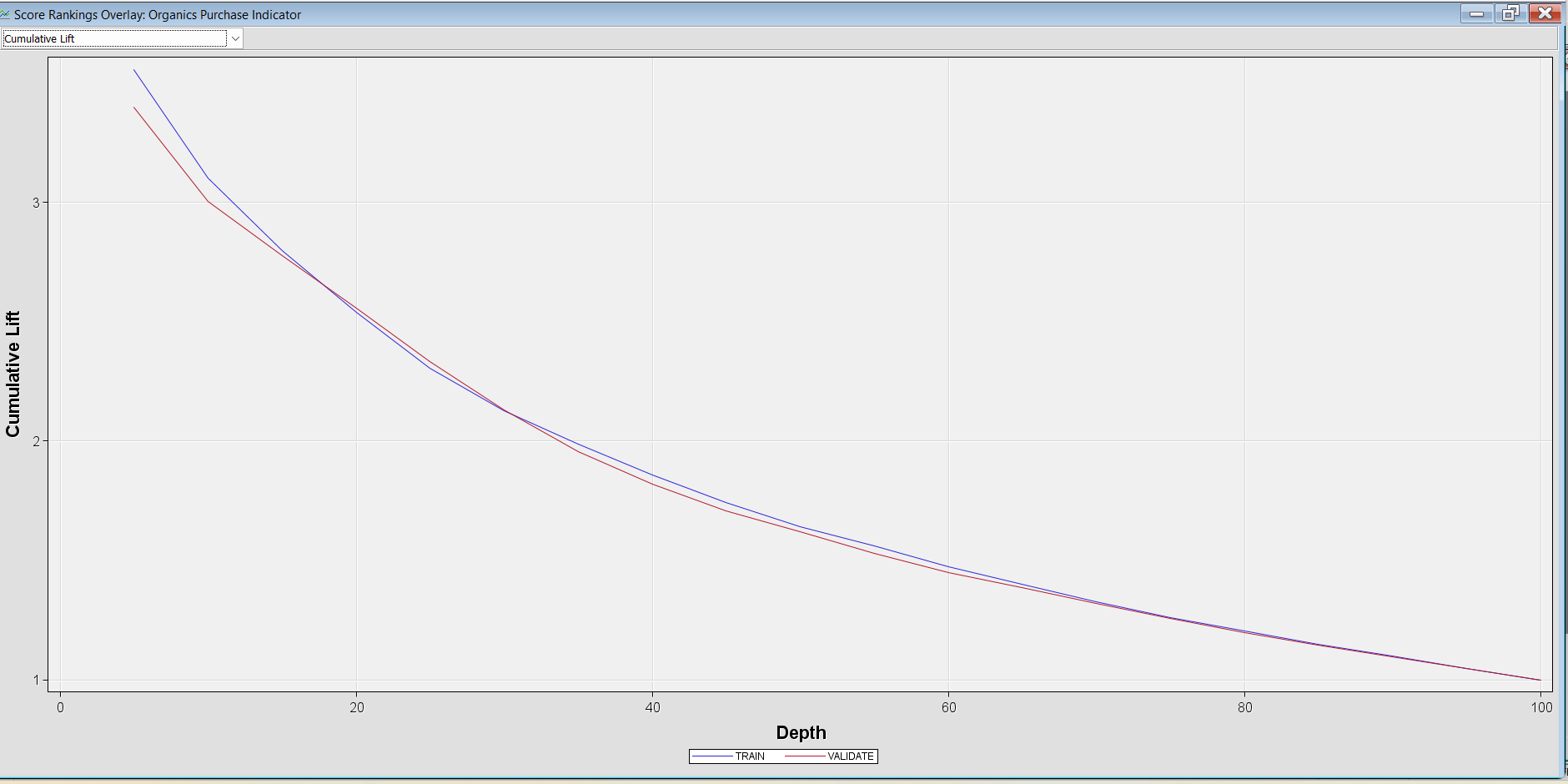


Figure 29: Cumulative lift of Default Regression



Figure 30: Fit statistics result of Regression (Default)

The Cumulative lift chart of the stepwise regression also demonstrates the percentage of positive responses relative to the percentage of customers contacted. The P-values for IMP\_Age, IMP\_Gender, IMP\_ProsperityClass, and M\_Gender are all below 0.05, highlighting their effectiveness as strong predictors. Additionally, the fit statistics indicate that the average profit for validation is 3.26, accompanied by a commendable RMSE of 0.37 for both train and validation. This suggests that our model performs well and has the potential for further refinement.

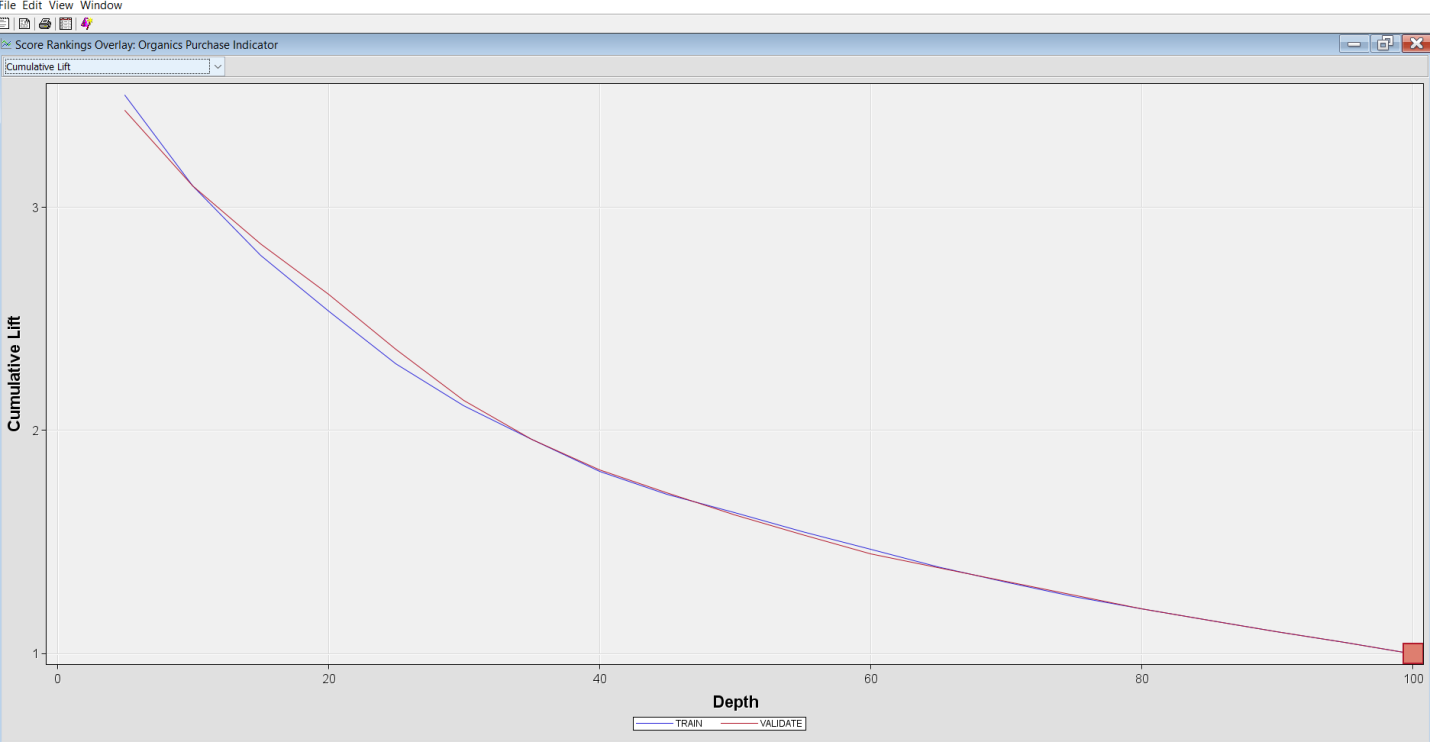


Figure 31: Cumulative lift of Stepwise regression

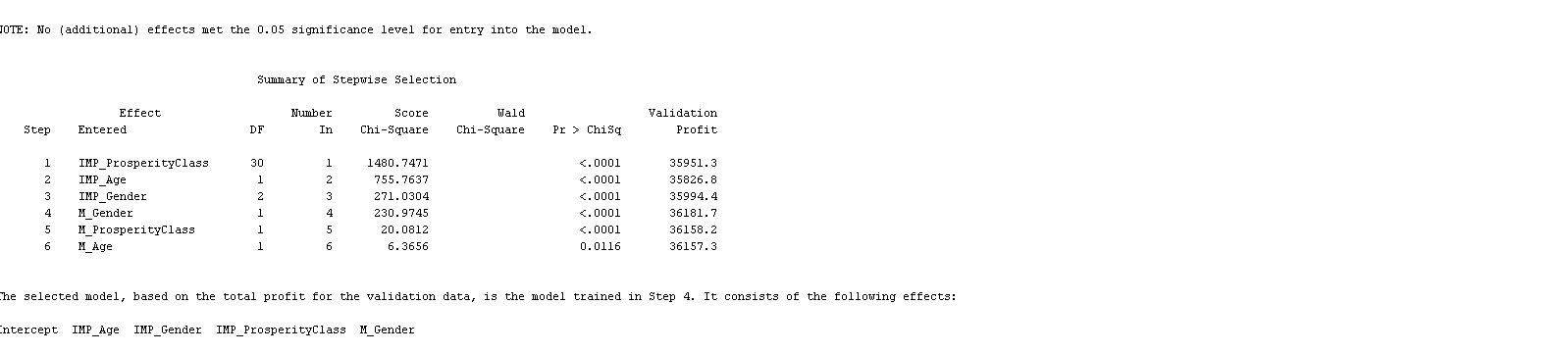


Figure 32: P-value for Regression (Stepwise)



Figure 33: Fit statistics results of Regression (Stepwise)

Another regression model used in this predictive modeling is polynomial regression. Polynomial regression in SAS is a powerful technique used to capture the intricate, non-linear connections between predictor and response variables. Unlike simple linear regression, which assumes a straight-line relationship, polynomial regression empowers analysts to elegantly fit curves by incorporating polynomial terms of the predictor variable, allowing for a more nuanced and accurate representation of the data. To use this model, we can select 'yes' for Polynomial terms.

The cumulative lift chart is distinct from others as it displays the highest score, demonstrating the percentage of positive responses relative to the percentage of customers contacted. The P-value indicates a significant relationship solely for IMP\_Age. During the validation phase, the average profit stands at 3.04, which falls short of our expectations. Furthermore, the elevated RMSE value highlights the model's inadequate performance. These results suggest that improvements are necessary to enhance the model's reliability and effectiveness.

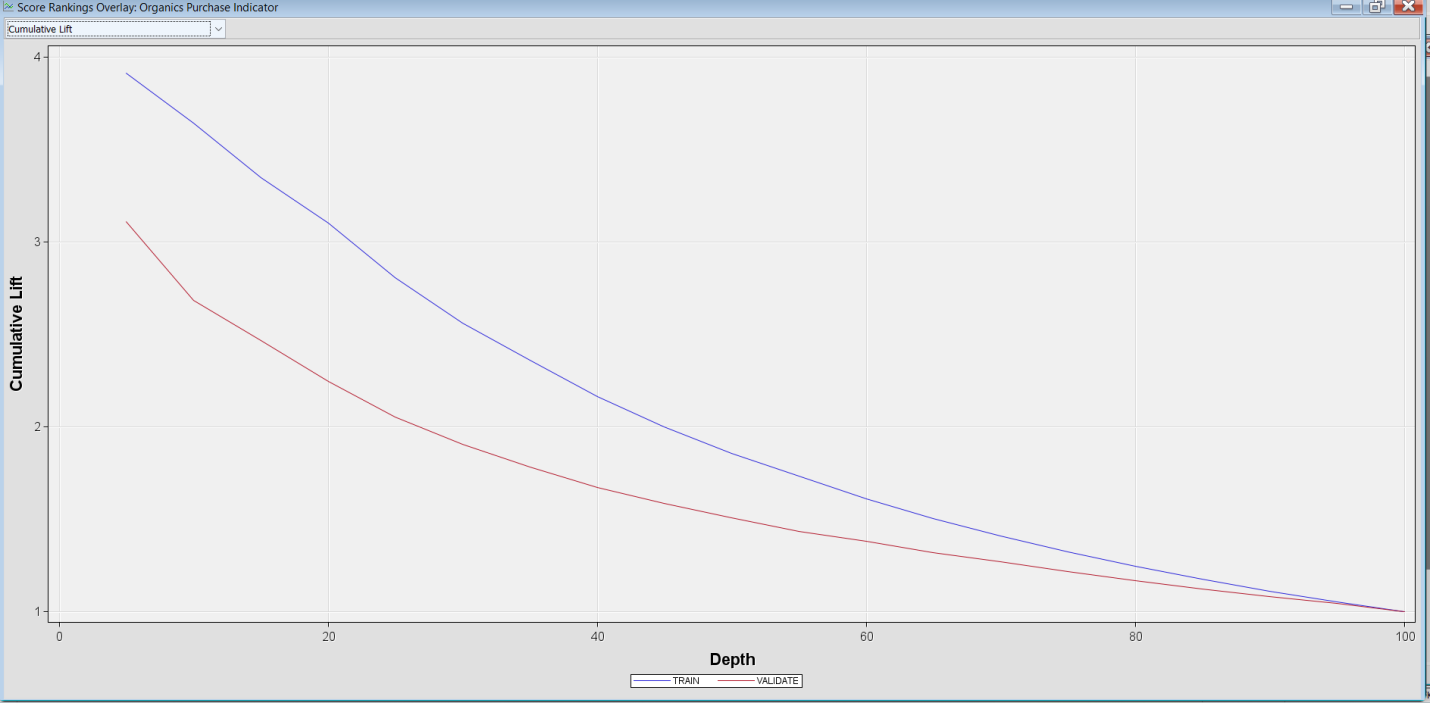


Figure 34: Cumulative Lift of Polynomial Regression.

1. **Support Vector Machine:**

Support Vector Machines (SVM) stand as a robust supervised machine learning technique primarily geared towards classification and regression tasks. Within the framework of SAS, the SVMACHINE procedure serves as the vehicle for implementing SVM, empowering users to develop classifiers tailored for binary pattern recognition challenges.

The Support Vector Machine (SVM) model requires a maximum of 25 iterations. In SAS, the SVMACHINE procedure configures SVM within the SAS Visual Data Mining and Machine Learning suite. This enables model training, hyperparameter tuning (such as kernel type and margin settings), and data preparation (including variable transformations and exploratory analysis).

The SVM fit statistics reveal some promising results. The total number of data reads is 22223, with 11112 utilized for the analysis. We have 3 input interval variables and 1 class variable encompassing 12 categories. The identification of 11112 support vectors suggests that the SVM is performing effectively, aligning well with our other models. Achieving an accuracy rate of 78% is a strong indicator of its potential. Furthermore, the cumulative lift highlights solid performance, while specificity stands at an impressive 97% for both training and validation datasets. The levels of sensitivity are also commendable, and the misclassification rate remains low, which points to the reliability of our model. Overall, these findings indicate that we are on the right track with our SVM implementation.

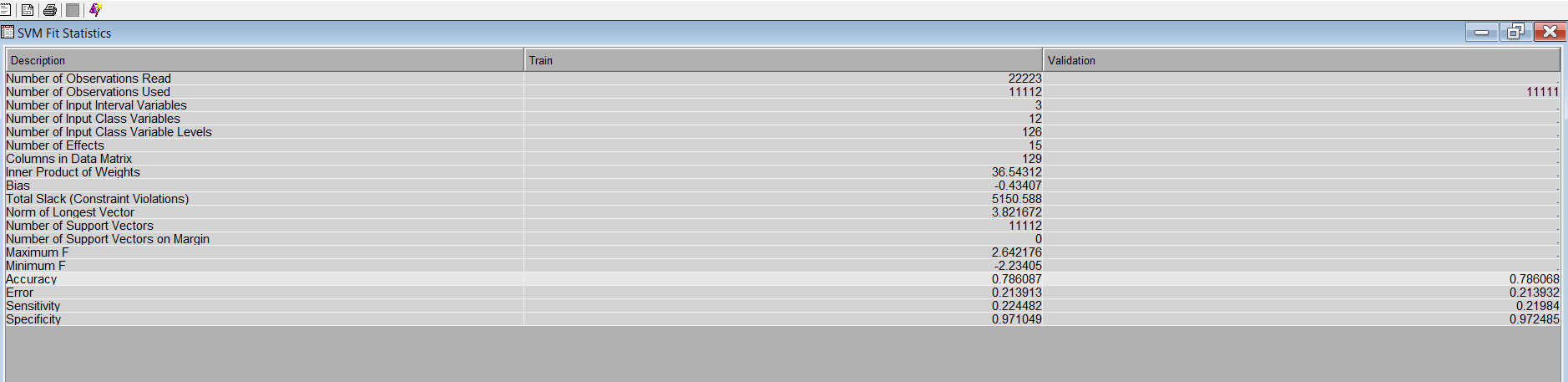


Figure 35: SVM fit statistics results.

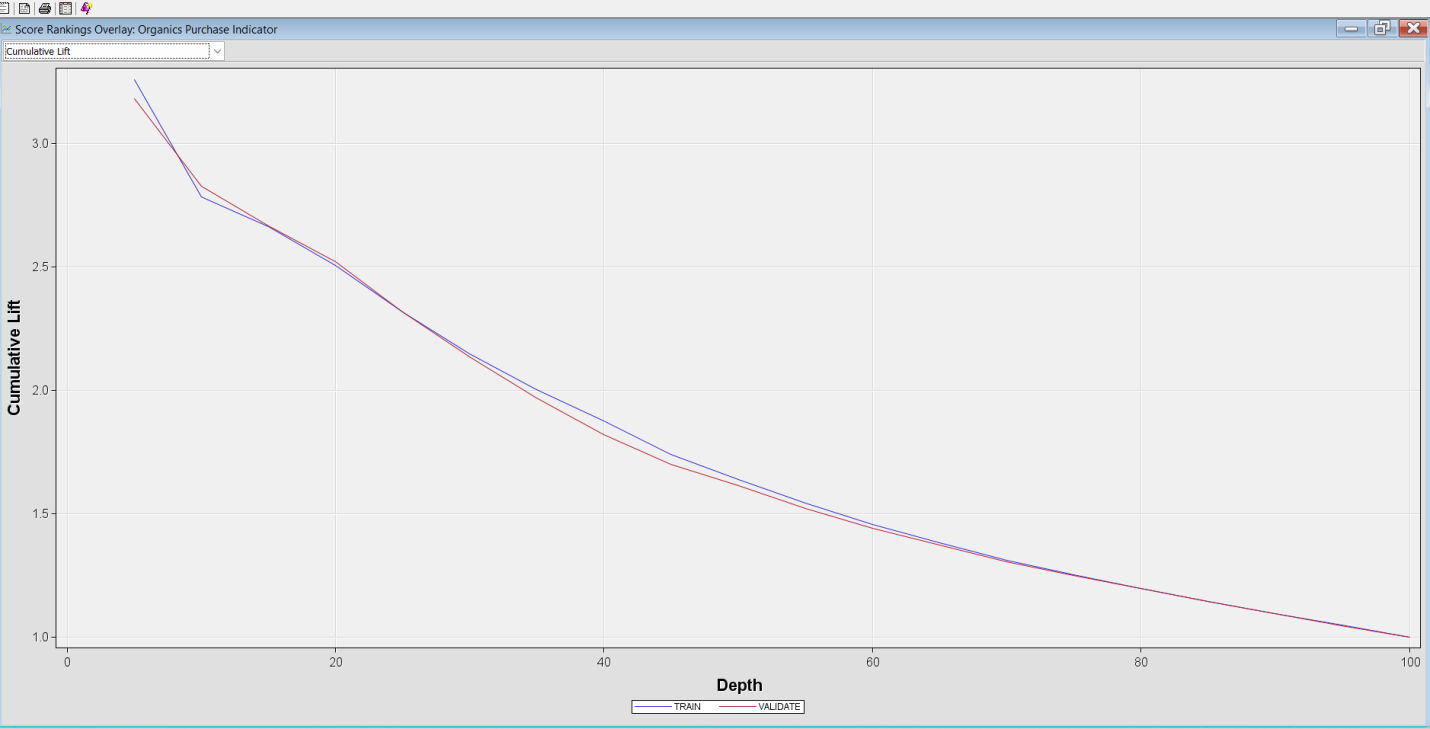


Figure 36: Cumulative lift chart of SVM

1. **Neural Network:**

Neural Networks are a type of parametric model that can be used to compare with Regression Models. They can handle various nonlinear relationships between a set of predictions and a target variable, often better than regression models. To configure the Neural Network, I opened the Network window and selected "Direct Connection" as "Yes." This enables the network to directly connect between input and output units, in addition to the connections made via hidden units. I chose "5" as the value of the "Number of Hidden Units," which means that our multilayer perception neural network will train with five units in the hidden layer. With our Neural Network node now configured for model training, The lift chart is a powerful tool that measures the effectiveness of our model. It compares the percentage of positive responses generated by the model to the percentage expected from random selection. A lift greater than 1 demonstrates the model's effectiveness by identifying more positive responses than random selection would.

The fit statistics reveal that the average squared error is consistently 0.13 for both the training and validation datasets, indicating a reliable model performance. Additionally, the misclassification rate is 0.18, which is an improvement compared to other models. Furthermore, the average profit for validation is 3.27, suggesting strong potential for positive returns. Overall, these results highlight the effectiveness of the model and its viability for future applications.

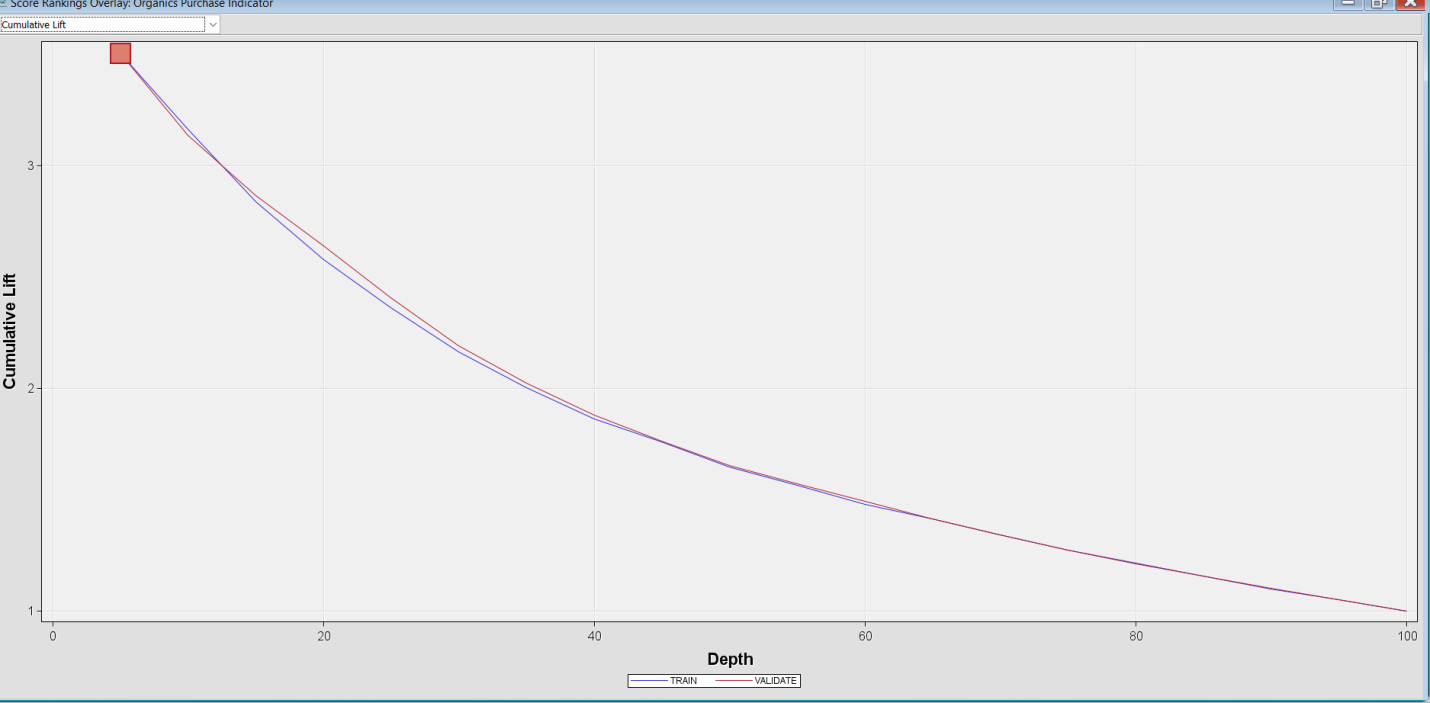


Figure 37: The Cumulative lift chart of Neural Network.



Figure 38: Fit statistics results of Neural Network

"I have improved the network flexibility by increasing it to six and incorporating another neural network in the predictive modeling process. Increasing flexibility allows neural networks to better adapt to the complexities of data, resulting in improved model fit and more accurate predictions. Flexible neural networks excel at handling complex data structures and relationships, making them particularly beneficial in domains such as finance, marketing, and insurance, where data tends to be highly nonlinear and multivariate. The adaptability of neural networks enables the utilization of different optimization methods and regularization techniques, which can enhance the model's performance and prevent overfitting. In addition, I have utilized an AutoNeural Network to compare our model against.

The Fit statistics results for the Extended Neural Network reveal promising findings. The average profit for the validation dataset stands at 3.28, which shows a notable improvement compared to the Neural Network. Furthermore, the average squared error is consistently maintained at 0.13 across both the training and validation datasets, indicating strong and reliable model performance. These results suggest that the Extended Neural Network is a valuable enhancement worth considering for future applications.

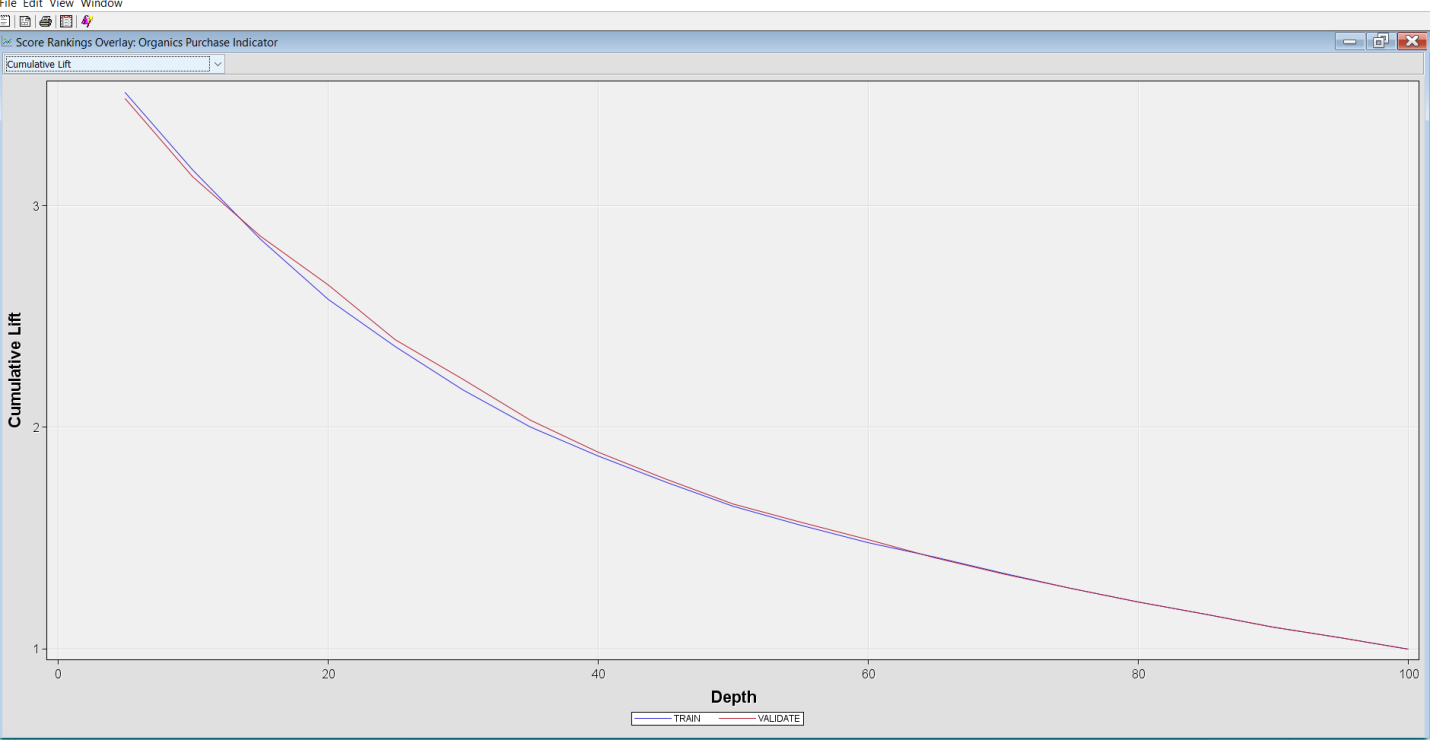


   Figure 39: Cumulative lift chart of Extended Neural Network.



Figure 40: Fit statistics results of Extended Neural Network

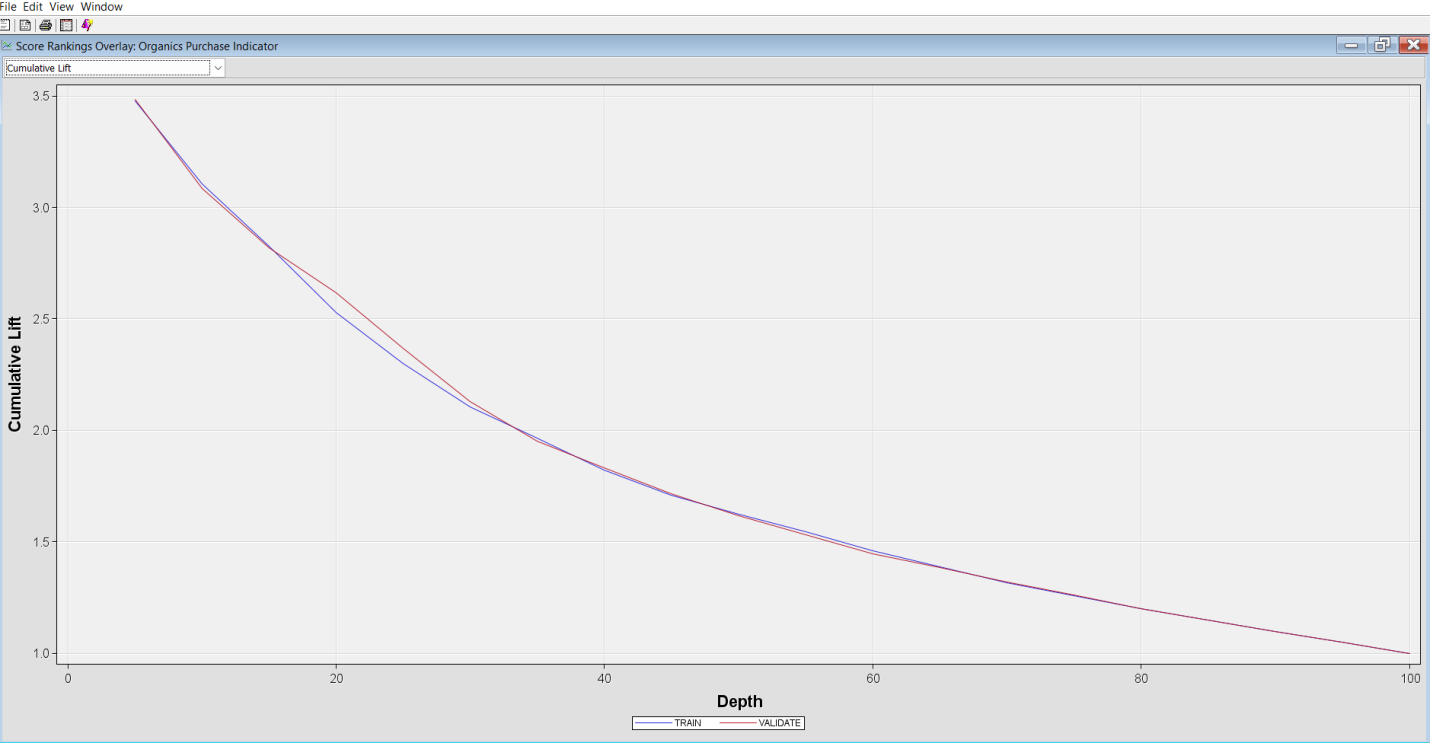


 Figure 41: Cumulative lift of Auto Neural Network



Figure 42: Fit statistics results of Auto Neural Network.

After comparing the three neural network models, it’s encouraging to see that the cumulative lift chart exhibits consistent positive performance across all of them. This finding suggests that each model has potential for the task at hand. Notably, the Extended Neural Network not only achieves a higher average profit but also demonstrates lower misclassification rates and average squared errors compared to the standard Neural Network and Auto Neural Network. Given these advantages, it would be beneficial to consider the Extended Neural Network as the leading model among the three, as it shows the most promise for future applications.

1. **Cluster:**

For my research project, I implemented two clustering techniques: Hierarchical clustering, which offers an automatic specification method, and K-means clustering, a user-specified approach that permits a maximum of three clusters. In the modeling phase, I utilized a filter node connected to the cluster node, employing all interval variables for the clustering process.The filtering yielded valuable outcomes, with an observation count of 22223, while 20942 data points were filtered out, and 1,281 were excluded. The analysis using the Cubic Clustering Criterion (CCC) indicated that the optimal number of clusters is 49, exhibiting a low root-mean-square standard deviation for cluster 1 among the four segments. From the segment results, I found that Segment 1 had the highest count, with 7,678 observations, representing 36.22% of the total. Age was identified as a key factor in this segment. Conversely, K-means clustering showed impressive results as well, achieving a count of 8,235 observations in Segment 2, accounting for 38.85% of the total. The variable "AmountSpent" emerged as significant in Segment 2.

**Decision: Identifying the Most Valuable Customers**

After analyzing the segments, we found that Segment 2 in K-Means has the largest customer base, with 8,235 members (38.85%), and is significantly influenced by the amount spent. This suggests that it may be the most promising cluster for effectively targeting high-value customers.

In contrast, Segment 1 from Hierarchical Clustering revealed a strong correlation with age. While this insight is valuable, we still need to explore whether age is linked to higher spending. By doing so, we can ensure that we make the most informed decisions about our targeting strategies.

Overall, my findings suggest that the K-means clustering method demonstrated superior performance, highlighting its effectiveness in this research context. This insight can guide future analyses and clustering applications.

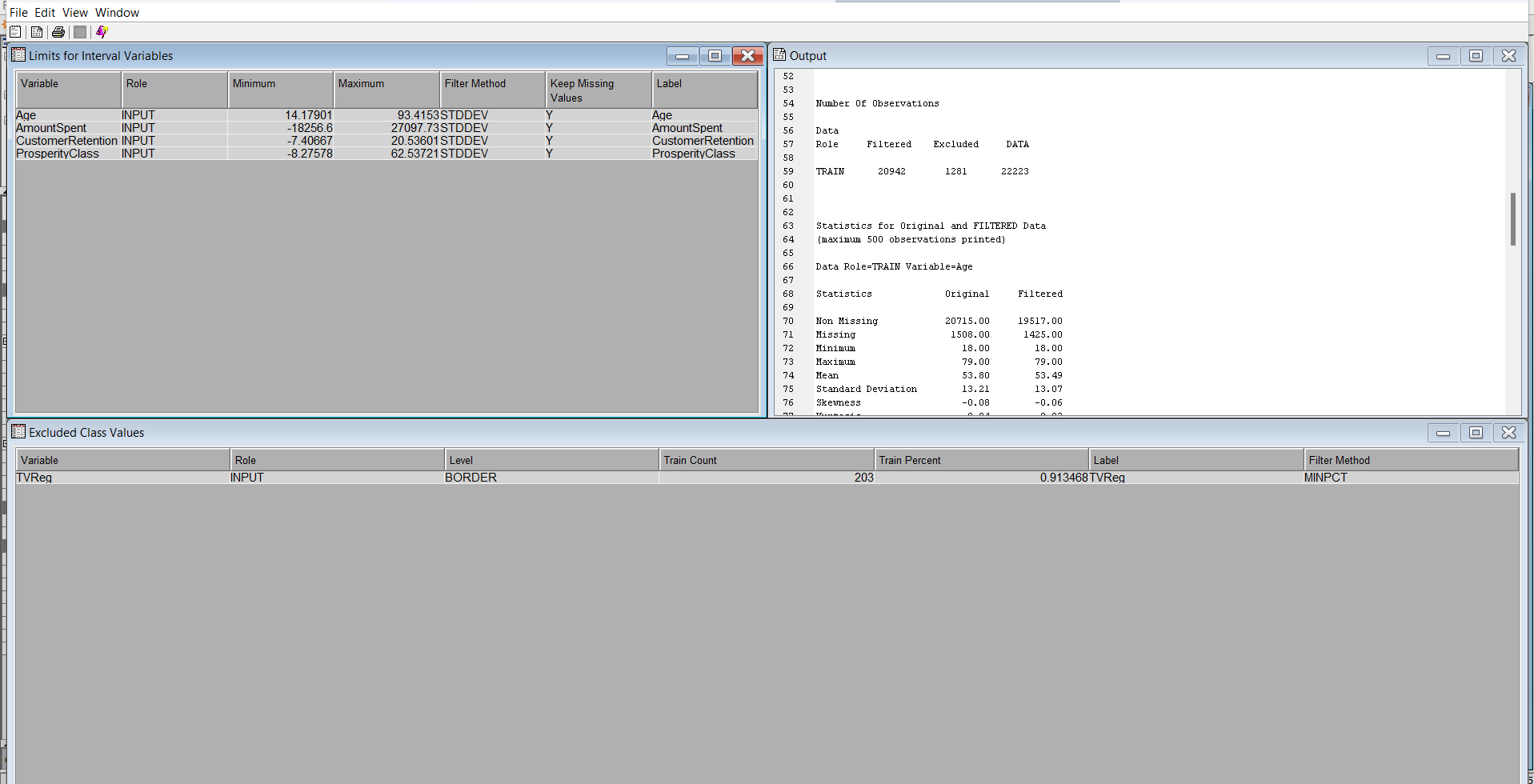


Figure 43: Statistical report of datasets.

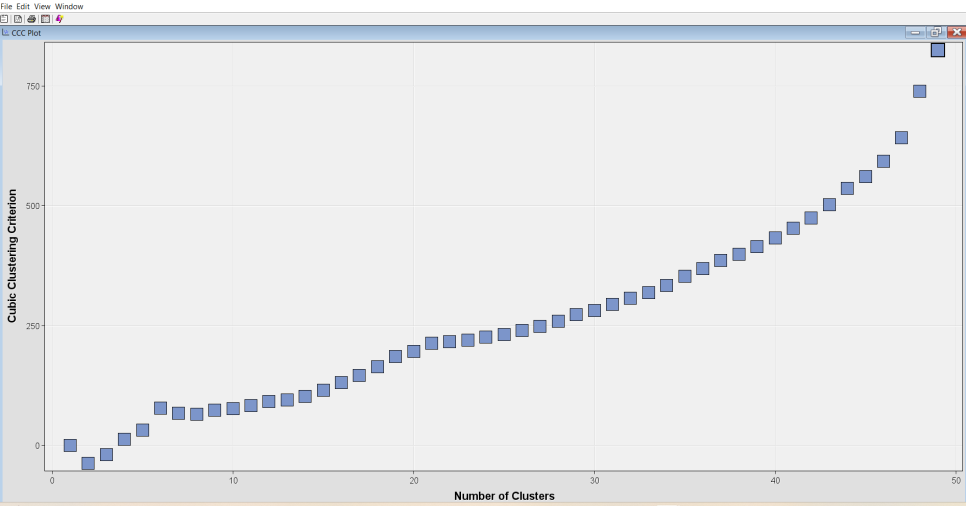
****

Figure 44: Optimum number of clusters



Figure 45: Segment results of Automatic Cluster

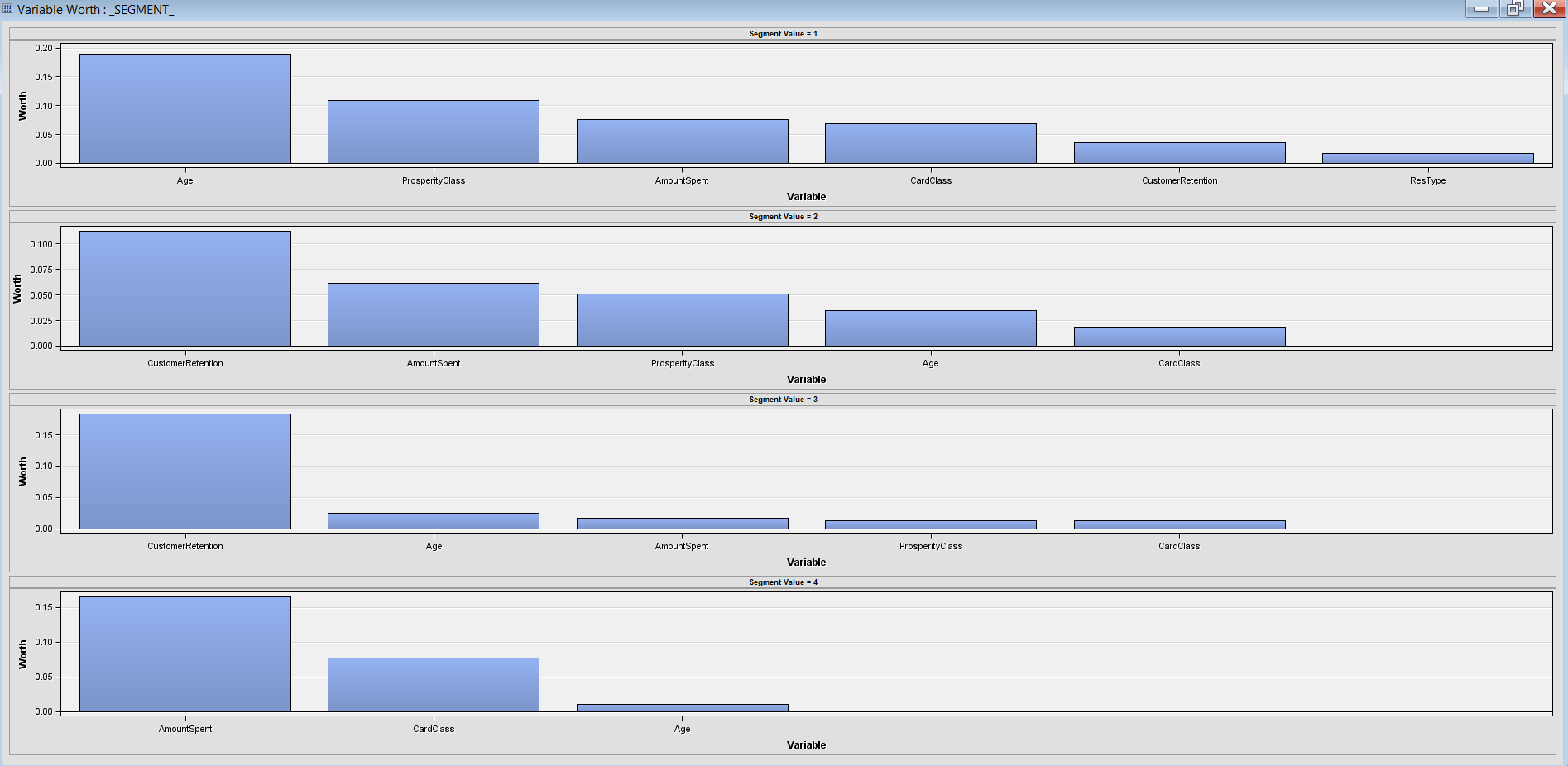


Figure 46: Variable worth of Automatic Cluster

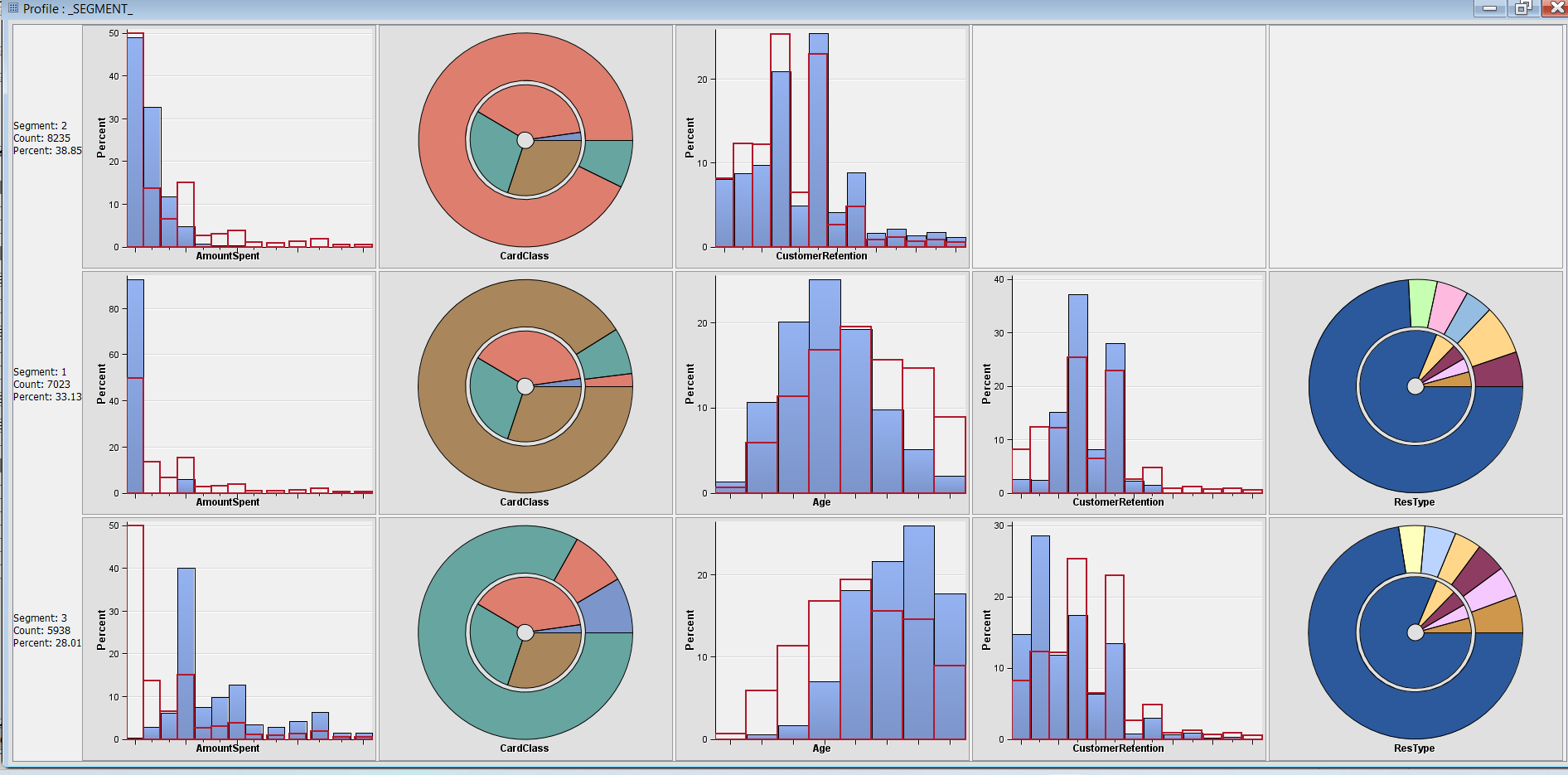


Figure 47: Segment results of K-means cluster

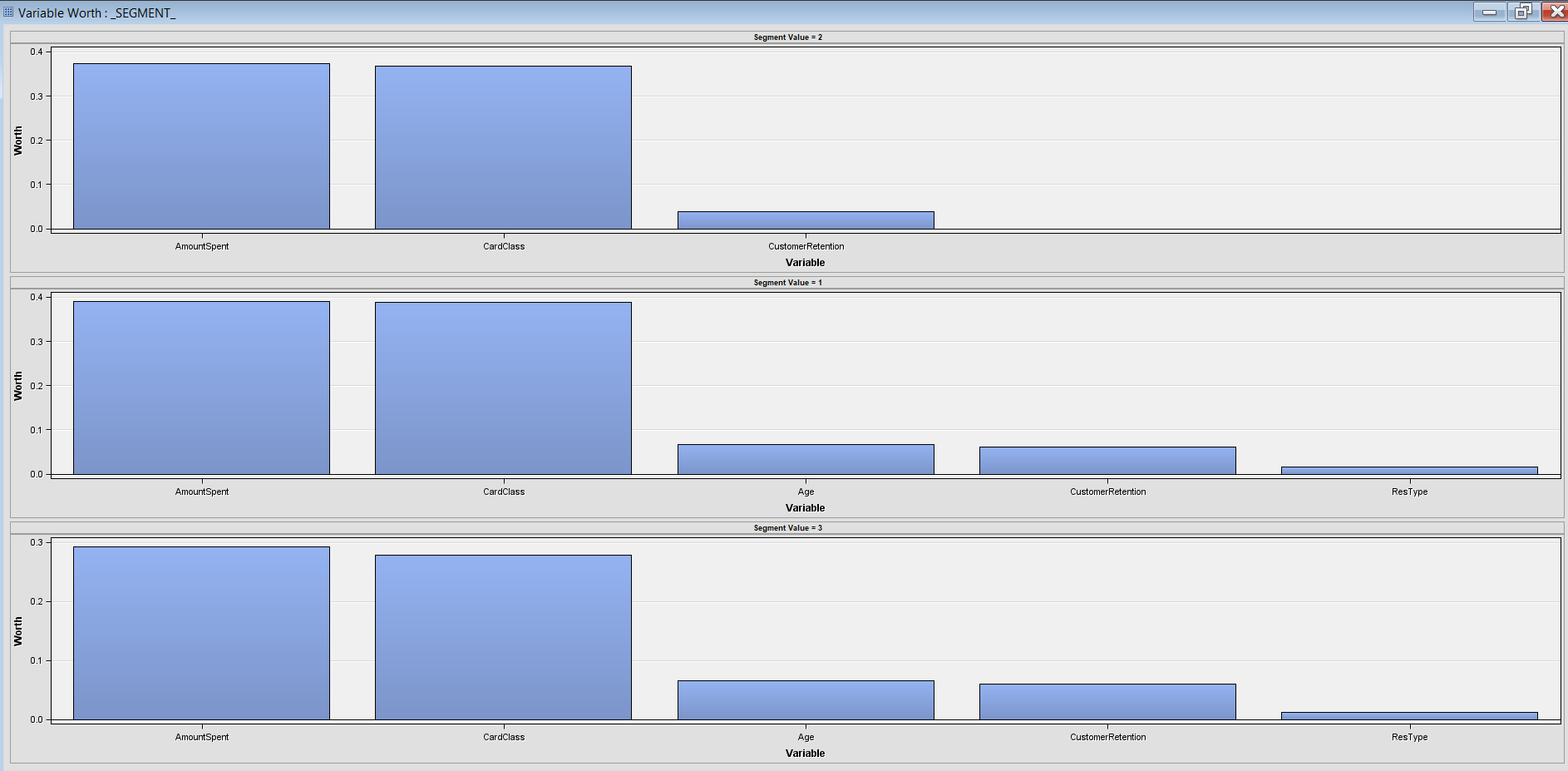


Figure 48: Variable worth of K-means Cluster

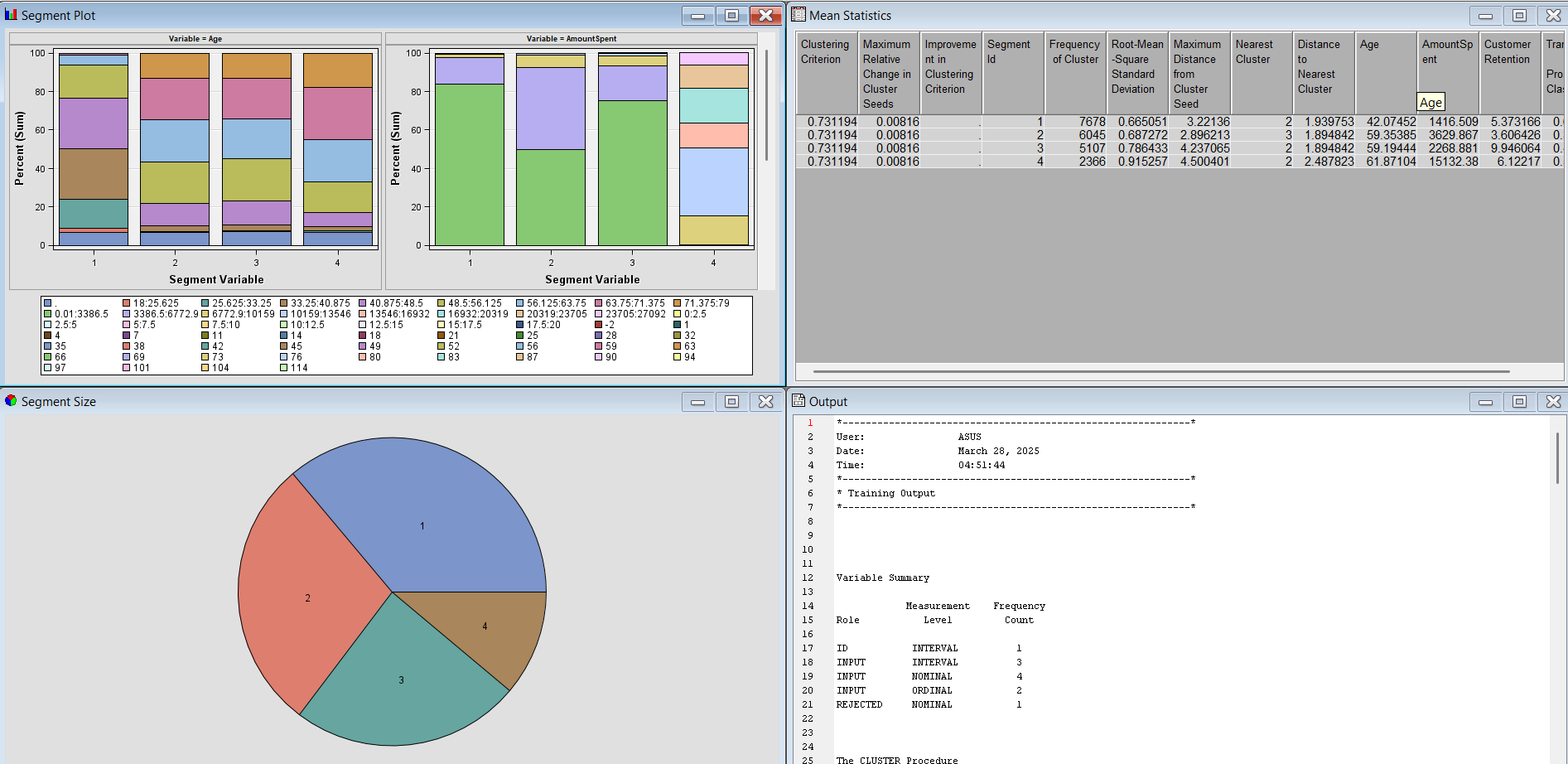


Figure 49: Optimal cluster results

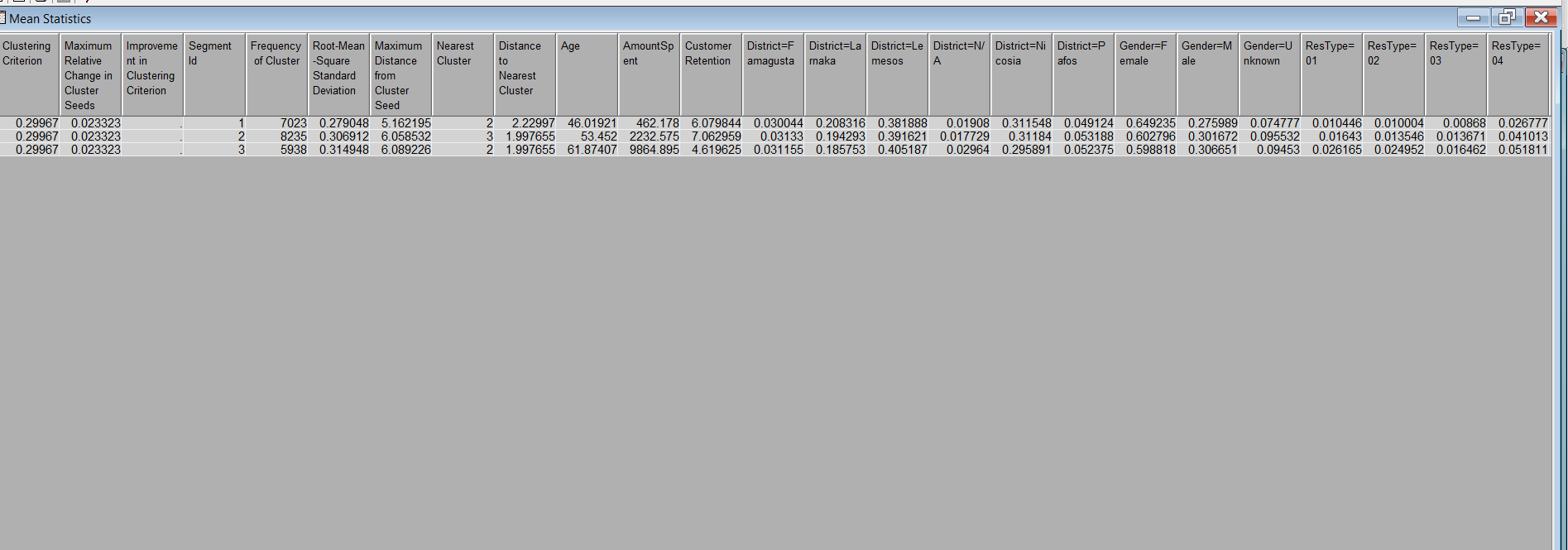
****

Figure 50: K-means cluster results

**4. Model Evaluation:**

I have utilized a Model Comparison node to assess potential models and choose the best one. This node is linked to our second Control Point. I also included all non-parametric and parametric models with a second Control Point. Notably, no property configuration was required for our data mining task with the Control Point node.

I adjusted the selection statistical property to measure the average profit/loss in the model comparison properties' model selection group. After that, I chose the Selection Table property for Validation. Using the validation data, we will use the average profit as our selection criterion for the Target statistic.

After conducting the model comparison, a results window will appear. To identify the champion model, we will carefully analyze the Fit Statistics window. The results show that the Neural Network with increased network flexibility achieves the highest average profit statistics and the lowest misclassification rate, at 3.28 and 0.18, respectively. Additionally, this model also has the lowest Average Squared Error and the fewest false classifications. While the Cumulative Lift is slightly higher for this Neural Network, it is very close to that of the Neural Network with increased flexibility. All accuracy models are valid for our Target = 1. Based on the highest selection criterion score, this model is chosen as the champion model. It is important to note that the Model Comparison node defaults to using the Average Profit criterion when the input data source contains a profit matrix.

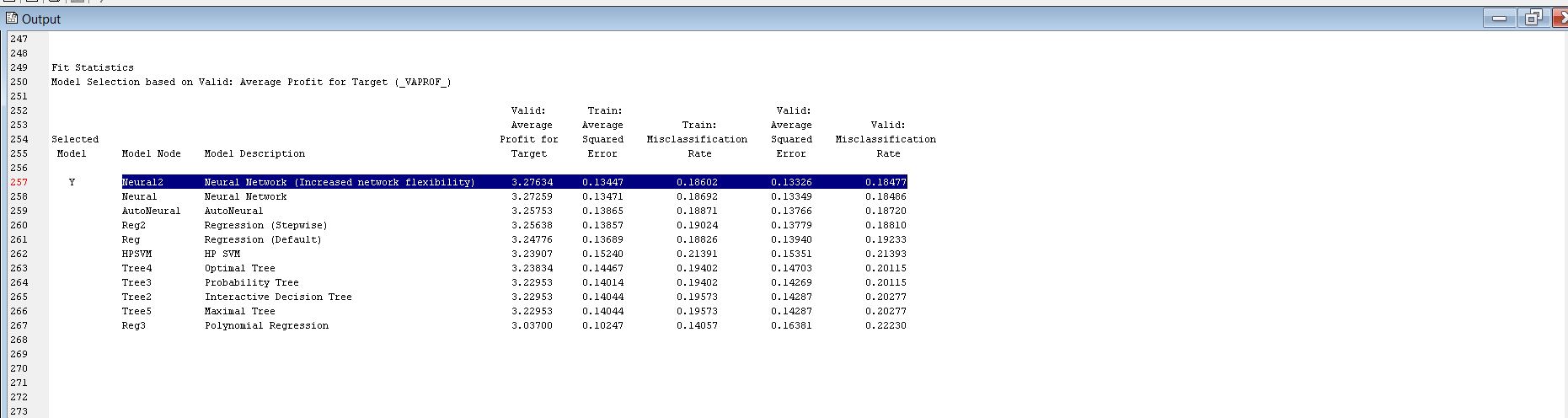


Figure 51: Average profit for target

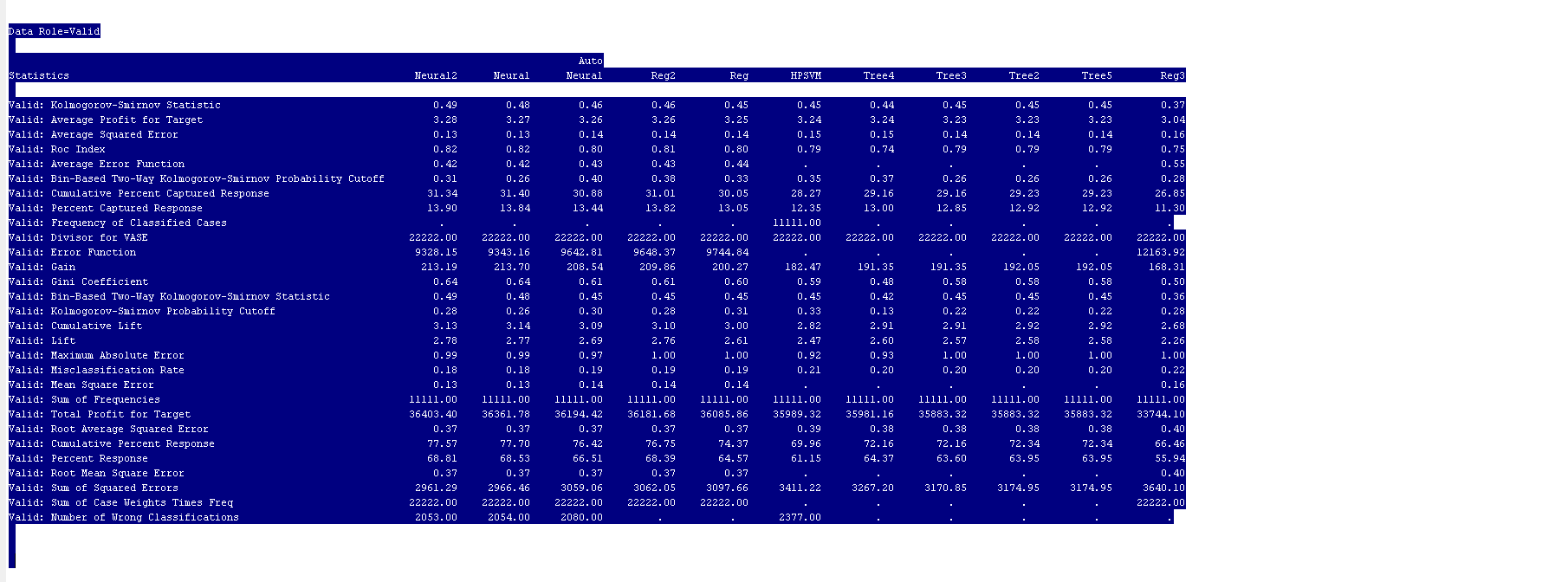


Figure 52: All accuracy metrics and average profit when Data role = Valid

The Total Computed Profit plot provides valuable insights into the total predicted profit for each candidate model's selection algorithm, evaluated on both training and validation data. This analysis allows us to identify models that deliver the most profit using the validation data. Notably, the Neural Network (with increased network flexibility), the standard Neural Network, and the AutoNeural Network models exhibit very similar total computed profits of $7,262, highlighting their competitive performance.

When we examine the Receiver Operating Characteristic (ROC) curves of all models, the curve for the Neural Network (with increased network flexibility) stands out as it remains close to the upper left corner, a desirable position that indicates strong predictive capabilities. Additionally, the model demonstrates a Cumulative Lift of 1.89, achieved with a depth of 40, which suggests effective targeting.

In terms of average profit statistics and misclassification rates on the validation data, the Neural Network (with increased network flexibility) emerges as the most promising champion model for prediction. Its precision surpasses that of the other models, successfully predicting an impressive average profit of 3.28 for the target customers. This positions the Neural Network as a highly effective tool for maximizing ROI and enhancing decision-making processes.



   Figure 53: Total Computed Profit for champion model

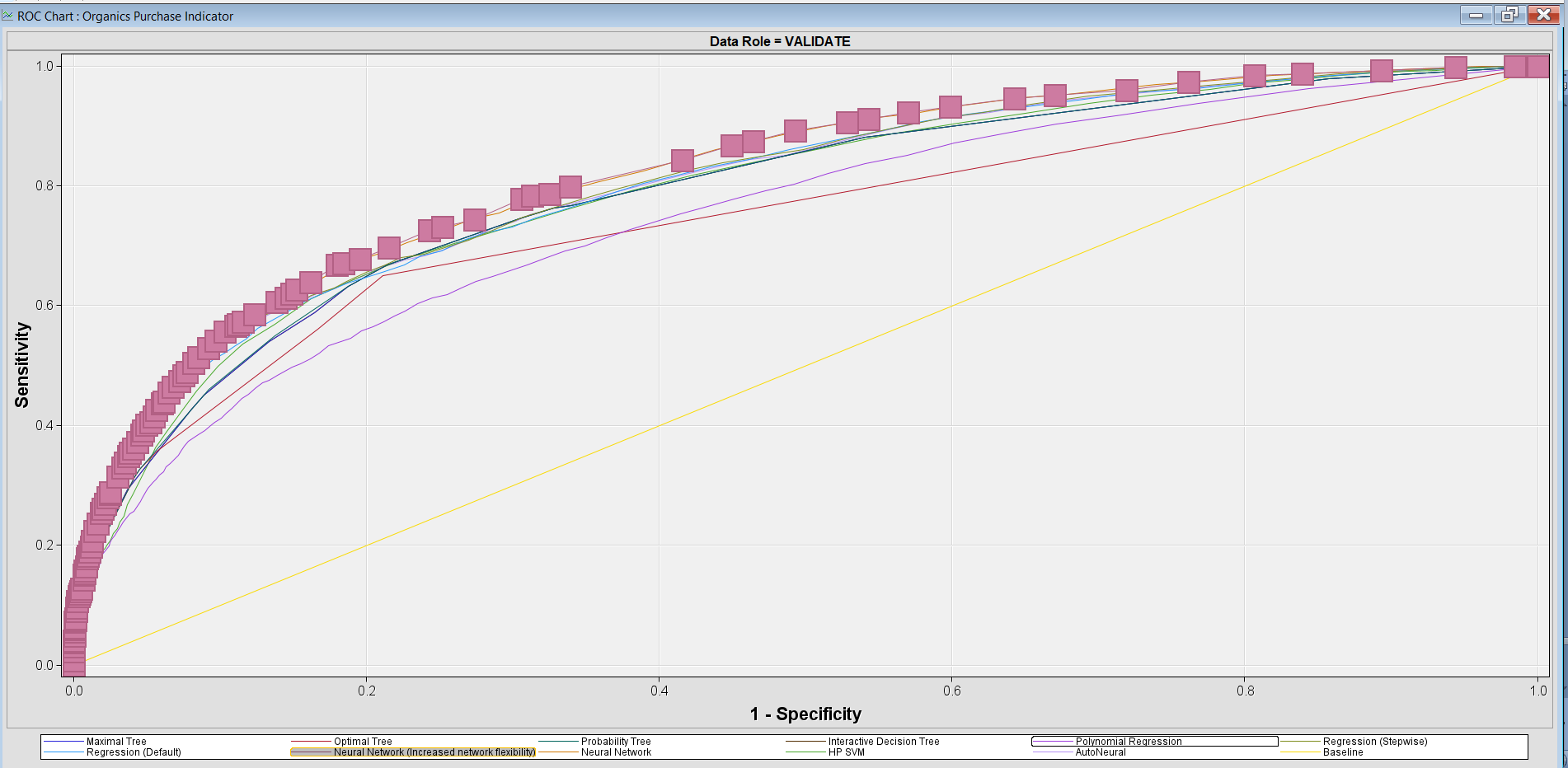


                       Figure 54: ROC curve for the champion model.

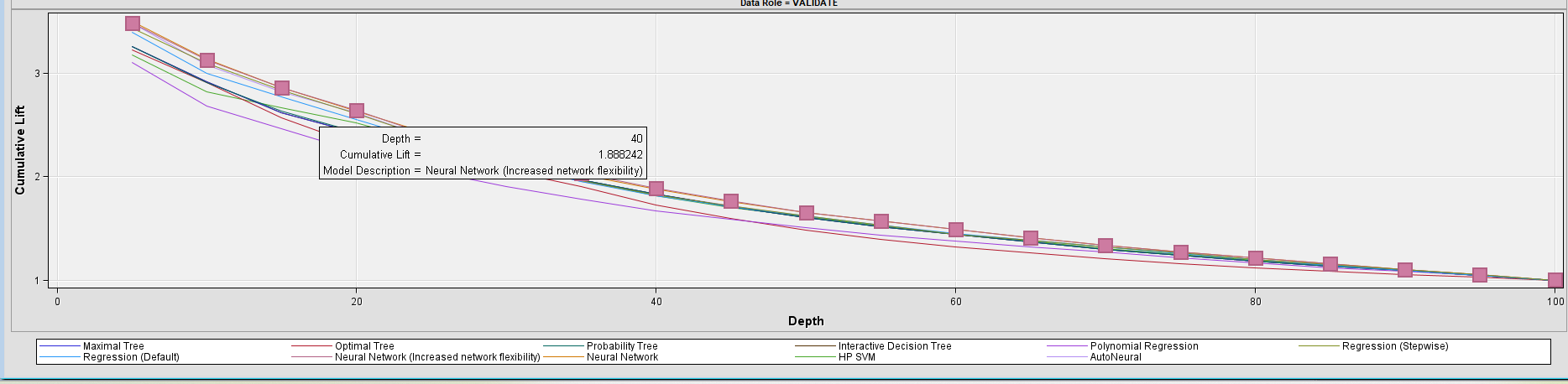


Figure 55: Cumulative lift result for Neural network 2 ( Extended NN)

Upon comparing all the trees, we found that the Default Decision tree shows the highest average profit for the target at 3.27, but it has a higher valid misclassification rate of 0.23850. The Probability tree has a profit which is slightly lower at 3.24, but it has the lowest valid misclassification rate of 0.20115. Hence, we can consider the Probability tree as the best option for modeling.

The analysis of our customers' data has been successful, as indicated by the clustering results. By examining the segment results, we can distinguish the preferences of different customer groups. The Mean statistics result of the K-means cluster shows that customers spent the highest amount of 10194, while segments one and three spent a lower amount. This information can be useful in predicting which customers are more likely to purchase our new products.

**5. Results And Discussion:**

I have used the champion model to score new data, which will help us identify the individuals who are most likely to love our latest products. To do this, I connected the CO4762\_SCOREDATASET in the data source with the score node, also linked to the Model Comparison node. The scoring code has been generated and used to score the potential purchasers on the list. The SAS code window displays the SAS DATA code required to append predictions from the champion model to a score data set. Once the purchasing customer list is scored, we can organize the raw score output and create a list of recommended individuals to buy the latest products. To save the data properties, I have utilized the Save Data node. We can see that the output from the Save Data node shows a total observation of 5000 of 15 variables, proving that we have successfully identified customers' potential preferences.

In our ScoreData results, we observed that when our Target = 1, the EM\_PROFIT is displayed, and when the Target = 0, the EM\_PROFIT is 0. We also noticed that when IMP\_ProsperityClass is high, both the EM\_PROFIT and EM\_probability are high, and the majority of our predicted customers are female. Hence, it is evident that most of our product enthusiasts are female, with an average age of 35. Based on my study, I recommend that we identify our most probable customers based on Target=1, EM\_ProsperityClass and EM\_PROFIT.

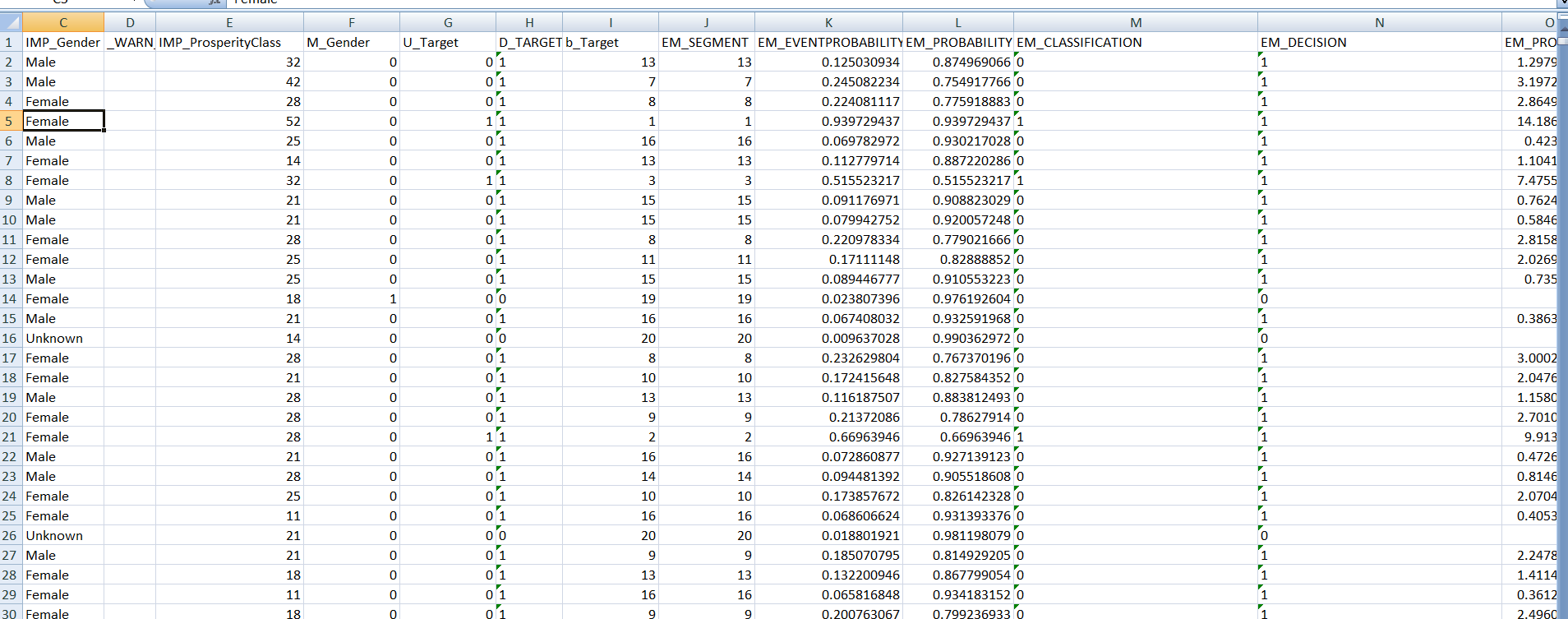


Figure 56: First 30 rows of the output