Diploma in Predictive Data Analytics

Course assignment

Student: Gareth Duffy

Student no: 21700349

Lecturer: Joe Fitzgerald

Introduction to the dataset and the purpose of predictive data analytics

Data is now arguably the world's most valuable resource. The fundamental purpose of predictive analytics is to serve as an effective tool which utilises this data in order to predict the probability of something occurring in the real world. This involves manipulating and testing data gathered from various sources, whether it be in business, economics or the sciences. Through scientific method, predictive analytics allows us to build rigorous theoretical models based on worldly phenomena, test these models and apply them to real world situations. This iterative process allows us to confirm or reject hypotheses, form assumptions, and then generalise and/or apply these findings to numerous fields of human endeavour.

This project focuses on two datasets. The first is comprised of 17 regular attributes relating to customer orders made via an online shopping website. Attributes include product category, shipping cost, and order quantity, and twelve attributes contain missing data. There are 8399 examples in this set. The second dataset contains 10 regular attributes relating to the same data, five of which contain some missing values. There are only 100 examples in this dataset. While attributes such as profit and sales are present in the first dataset, they are absent in the second. These datasets were used to build four models which demonstrate the effectiveness and value of predictive analytics.

Model 1

Validation of the performance of a decision tree model created from a partitioned dataset

Pre-processing stage

Firstly, I plugged the large dataset into the process field and used the S*elect Attributes* operator to omit attributes that I considered unnecessary i.e. customer name, row ID, order ID, order date, ship date, ship mode, product container, profit, sales and product-based margin. These attributes were not of immediate interest because I hypothesized that none would greatly affect my predicted label attribute (choice of product category), whereas attributes like shipping cost and discount might. They were omitted to also reduce the probability of creating noise within the model.

Next I employed *Work on Subsets* to refine certain attributes within its sub-process field. Here I incorporated *Normalize* in order to standardize the product based margin values. Product based margin accounts for the percentage of selling price that is turned into profit, and it was this percentage-type value that I intended to transform into more conventional numerical values. Next I plugged *Replace Missing Values* into the *Work on Subset* process so as to reconcile any missing data within attributes. I replaced missing values by "subset" (10 attributes had missing values), and used "average" for method of replacement (See appendices fig 1.0).

Predictive modelling stage

Next I employed *Split Data* to partition the dataset into two sub-datasets. I chose a stringent ratio of 0.8 and 0.2 to split the dataset. Partitioning the dataset would allow me to use 80% of the original set to build a labelled model which I could subsequently apply to predicting the label attribute in the smaller (20%) dataset. I could also determine how reliable the model would be at predicting a customer's choice of product type. Following this, I plugged *Set Role* into the upper *Split Data* port in order to assign a label to my model, and a second *Set Role* into the lower testing dataset port. I chose product category as my model label as I was interested in seeing if the model could predict what type of product a customer would buy. I also wanted to discern whether other independent attributes in the dataset would influence product type choice.

My label attribute was polynominal so I felt a Decision Tree model would be suitable and would adequately illustrate some of the determining factors that might influence product category choice. Next, I plugged *Apply Model* into the *Decision Tree* and lower port on the *Split Data* to apply my trained model to the testing dataset. Lastly, for validation, I attached the *Performance* operator in order to generate a list of performance criteria values and obtain an overall assessment of how the model performed. This would also generate a "predictive" output and "actual" output which would allow me to discern how accurate the model was.

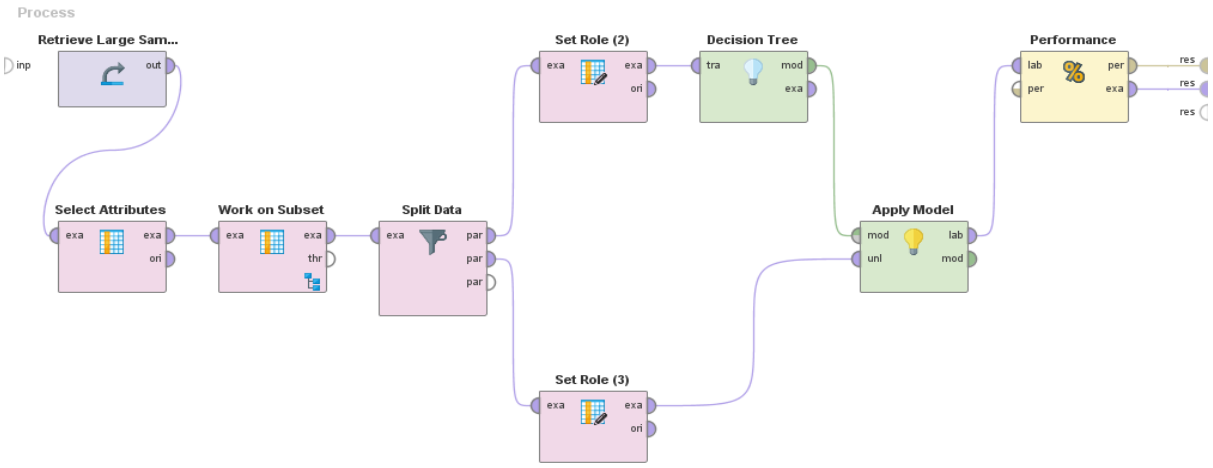


Fig 1.1 Decision tree and performance model process field

Outcome and interpretation stage

After running the model and performance processes, the performance vector showed that the model was highly accurate. Indeed, confidence percentages across all three product categories were very high and the overall accuracy was 80.95%. (See fig 1.2)

accuracy: 80.95%

	true Office Supplies	true Technology	true Furniture	class precision
pred. Office Supplies	889	141	124	77.04%
pred. Technology	21	307	18	88.73%
pred. Furniture	10	6	164	91.11%
class recall	96.63%	67.62%	53.59%	

Fig 1.2 Decision tree performance vector table

Upon analysis of the decision tree and model output (See appendices fig 1.3 & 1.4), it was inferred that some but certainly not all of the independent attributes contributed to predict product category choice. Shipping cost was the top predicting factor followed by unit price and

discount. Surprisingly, the type of segment a customer belonged to (Corporate, home office, consumer or small business) had little influence over product type choice. This was interesting as prior to analysis I was certain that a home office customer for example, might be more inclined to purchase office supplies over a conventional "consumer".

While the model was highly accurate, it nevertheless failed to account for other potential attributes which were absent from the dataset that might govern a customers' choice of product category. Factors such as necessity or basic needs (e.g. needing new furniture because someone is moving house), replacement, convenience, scarcity, value for money, emotional vacuum or brand recognition all play determining roles in a customers' choice of product type. The decision tree model was limited by these missing factors. It was somewhat convoluted and lacked crucial data pertaining to the real reasons why a customer choses a certain genre of product (See appendices fig 1.4).

Model 2

Using linear regression to predict future product sales levels

Pre-processing stage

To begin, I plugged both training and testing datasets into the process field. My regression model would not run correctly if all attributes were not similar across datasets. Thus, using *Select Attributes*, I omitted unnecessary attributes and those that were not were not present in the testing dataset. However, while "sales" was absent from the testing dataset, I retained it in order to serve as my predicted model label in the training dataset (See appendices fig 2.0). Next using *Remove Missing Values*, I substituted missing data for order quantity, order priority, sales, discount, unit price, shipping price, customer segment, and product base margin by method of averages. I then assigned sales as my label attribute using *Set Role*, to serve as my predicted attribute for the testing dataset.

Lastly, I plugged *Filter Examples* into the testing dataset. I needed to ensure that the attribute ranges in both datasets were equal for my model to function correctly. I verified range differences with the statistics window and adjusted for discount, unit price, shipping cost, and product base margin (See appendices fig 2.1).

Predictive modelling stage

Following pre-processing, I introduced the *Linear Regression* operator to the training data set and plugged this into the *Apply Model* operator. I also plugged the testing dataset into *Apply Model* to use my trained model on the testing data. I chose *Linear Regression* firstly because it is tailor-made for predicting label attributes of numerical type (compared to say

Neural Networks which better for polynominal data). *Linear Regression* would also afford the evaluation of each individual weight (coefficient) attached to each predicting attribute and would provide significance values each attributes in relation to my label attribute (See fig 2.2).

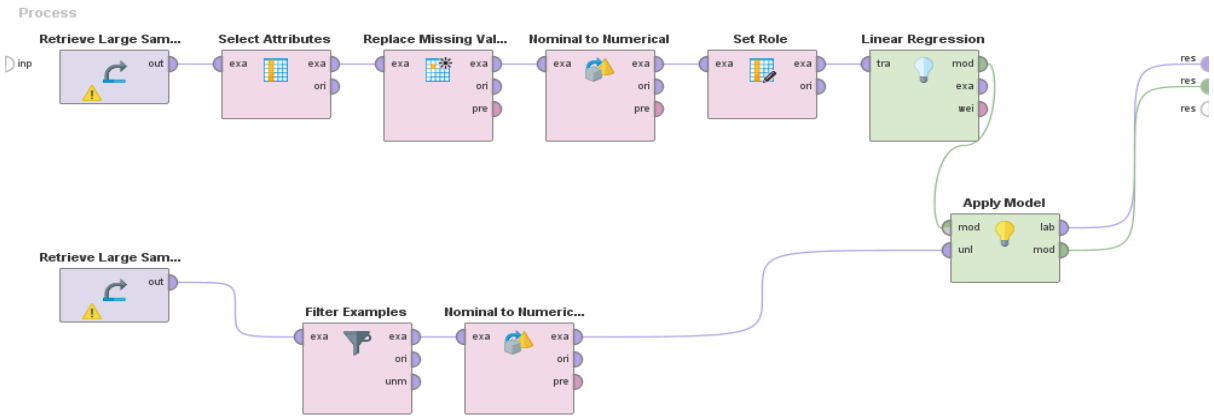


Fig 2.2 Linear regression model process field

Outcome and interpretation stage

After running the regression, it was revealed that the model significantly predicted sales levels in the testing dataset. Upon interpretation of the regression table, technology product category, order quantity and shipping cost were shown to be the most important predicting attributes for sales (See appendices fig 2.2 & 2.4). Technology product orders were the most important predicting factor for sales. Indeed, tech products contribute enormously to the day-to-day lives of most individuals in today's world, and it is of little surprise that the demand for these products would generate such huge returns for online shopping companies (See fig 2.3).

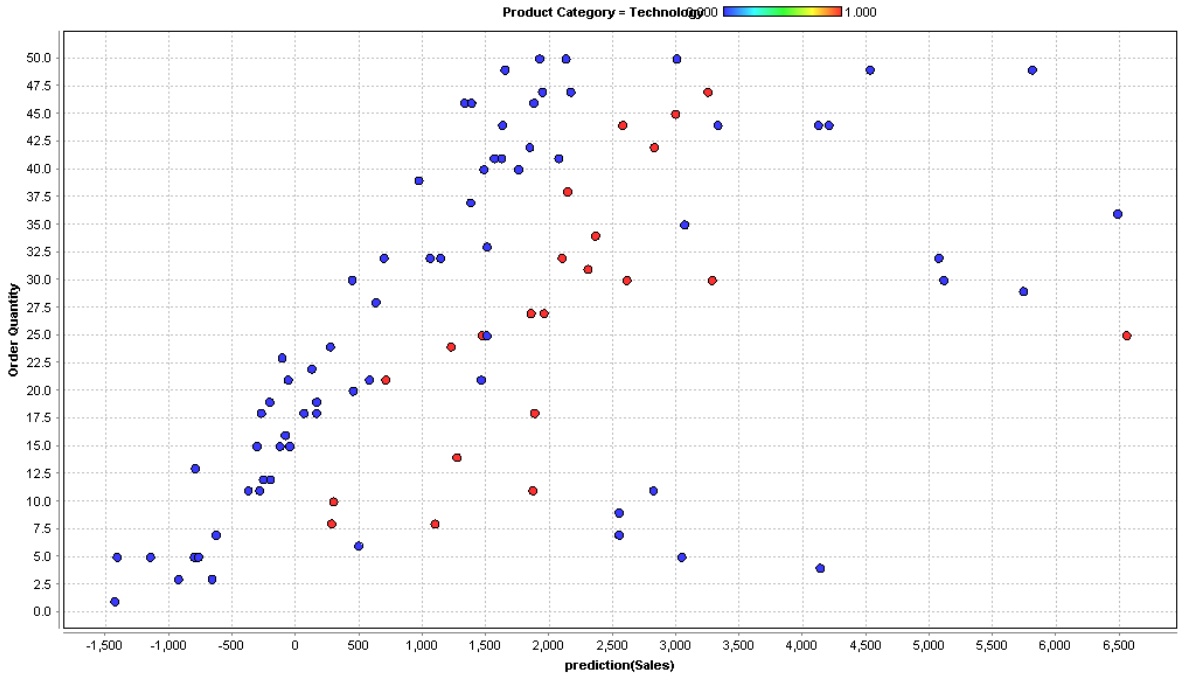


Fig 2.3 Linear Regression scatterplot showing positive correlation between quantity of Technology products ad Sales

Furniture was also a notable predictor for sales levels. Indeed, one need only look at the huge online furniture market which generates enormous profit from the preferences of eager home owners that companies like IKEA dominate. Business customer segment was found to be the least important factor at predicting sales, followed by item discount, albeit discount was still a significant predictor according to its p-Value. The regression results revealed that sales levels rested on a handful of intriguing predicting consumer factors.

Model 3

Cross validation and performance evaluation of a Naïve Bayes model to predict future profit

Pre-processing stage

First I employed a *Sub-process* operator to create a *Loop Values* process that would reduce and equalize order priority categories down to 1600 examples in each. To do this I used *Set Macro* to tell Rapidminer that the new Macro values would all become 1600 in size (See appendices Fig 3.0). With *Loop Values*, I set my loop attribute to order priority and gave the iteration macro the name of "loop_value". Next in the sub-process field, I used *Filter Examples* to tell Rapidminer that order priority groups were what I wanted to loop over and become equal to the new value I had assigned. (see appendices fig 3.1). Next I added the *Branch* operator to give Rapidminer and "if" "then" set of instructions which would initiate reducing the order priority sample sizes. Within *Branch*, I placed the *Sample* operator in the "then" process field so as to evoke my newly coded loop value (See appendices fig 3.2).

I returned to the main process field and employed a second *Sub-process* to create a smaller sample of my dataset by utilising *Macro* operators. I wanted to reduce the sample to 75% of its original size given the fact that it was notably large (8399 examples). I used *Set Macro* to add and store a new value to my dataset size which I could evoke later. I entered "0.75" into value parameters and "fraction" into the macro name box, which reduced the dataset to 6000 examples. Next I used *Extract Macro* to extract a value from the original dataset based on size. Here, I entered "size" for the macro name and selected "number of examples" in the macro type box.

Now I needed to name and load my new macro. For this I selected the *Generate Macro* operator. With the function descriptions box I created the name "new size", and entered the

functional expression: “round(eval(%{size})*eval(%{fraction}))”. This expression would evoke my previously created macro. It essentially rounds the new number to make it more manageable and evaluates the macro by multiplying the two macros together (See fig 3.3). Lastly, I needed an operator which would load what I had created from the *Generate Macro*. Accordingly, I used *Sample* to evoke my new macro by entering “%{new size} into the parameters window. Now my dataset was reduced to 75% of its original size.

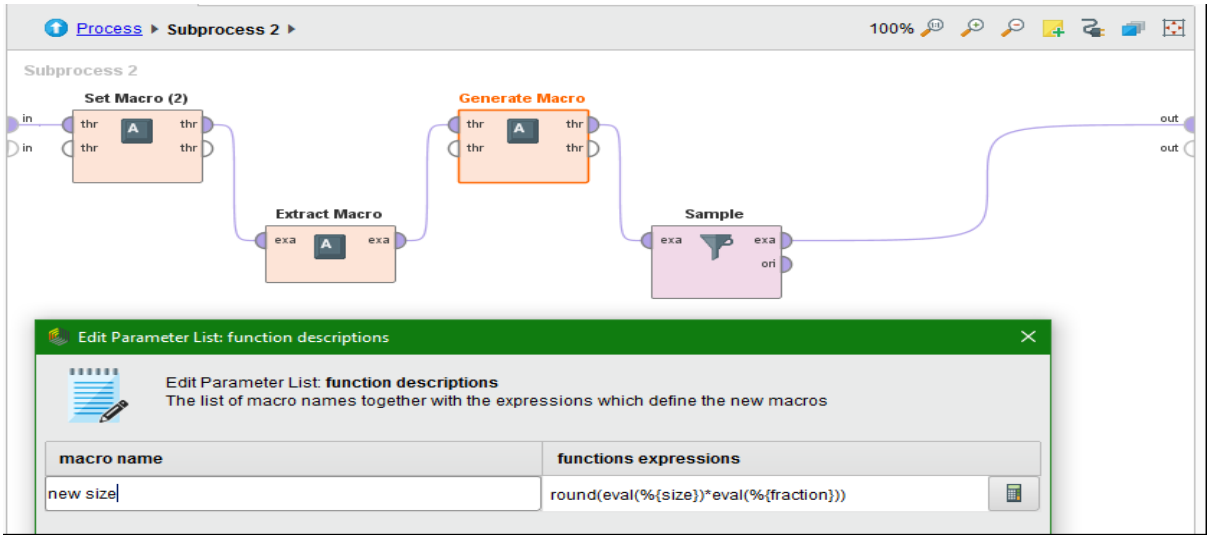


Fig 3.3. Process of using Macro operators to sample down the size of original data set

Returning to the main process field, I used *Select Attributes* to omit row ID, customer name, product container, ship mode, ship date and order date. My rationale here was that the product container or method of shipping would not directly affect profit made and thus I felt they were unnecessary for the purpose of my model. I then used *Replace Missing Values* to account for any missing data within attributes by method of averages. Next I added *Discretize* so that I could partition profit into 2 separate bins of equal ranges. I then assigned profit as my model label using *Set Role*, and assigned order priority as an additional ID role so as to identify profit levels by their order priority.

Predictive modelling stage

Next I chose the *Cross-Validation* operator to build a model based on profit and assess the performance of this model. *Cross-Validation* splits the dataset into subsets (folds) of equal size for substantiation, and would allow me to create a training sub-process, apply it to a testing sub-process, and subsequently measure its performance and validity (See Fig 3.4).

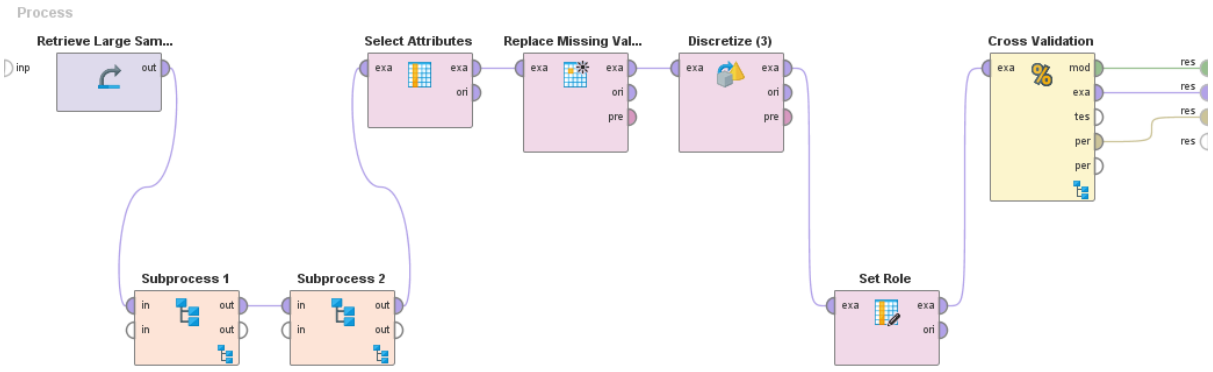


Fig 3.4 Cross-Validation model main process field

In the *Cross-Validation* parameters, I selected 600 folds to divide the data set into 600 subsets (and iterations). I had reduced my sample from 8399 to 6000 examples, and thus 600 folds was appropriate because it complied with an optimal ratio of approximately 1/10. Inside the *Cross-Validation* sub-process field, I chose *Naïve Bayes* as my model and placed it in the training window. I chose *Naïve Bayes* largely because its provides excellent output of how each individual attribute relates to and contributes to predicting the label attribute. This model assumes all attributes are independent and runs significance tests to compare the individual variances between each attribute and the label attribute. In this way, it provides a more comprehensive and decisive model than say, a decision tree. In the adjacent testing window, I

added *Apply Model* to apply my trained model to the testing dataset. Lastly, I attached the *Performance* operator to evaluate the performance of my model (See Fig 3.5).

Outcome and interpretation stage

After running the model and performance tests, the performance vector output revealed an overall accuracy of 98.02%, with a precision estimate of +/- 4% (See appendices fig 3.6). Therefore, the Naïve Bayes was confirmed to be a highly effective predictive model for profit.

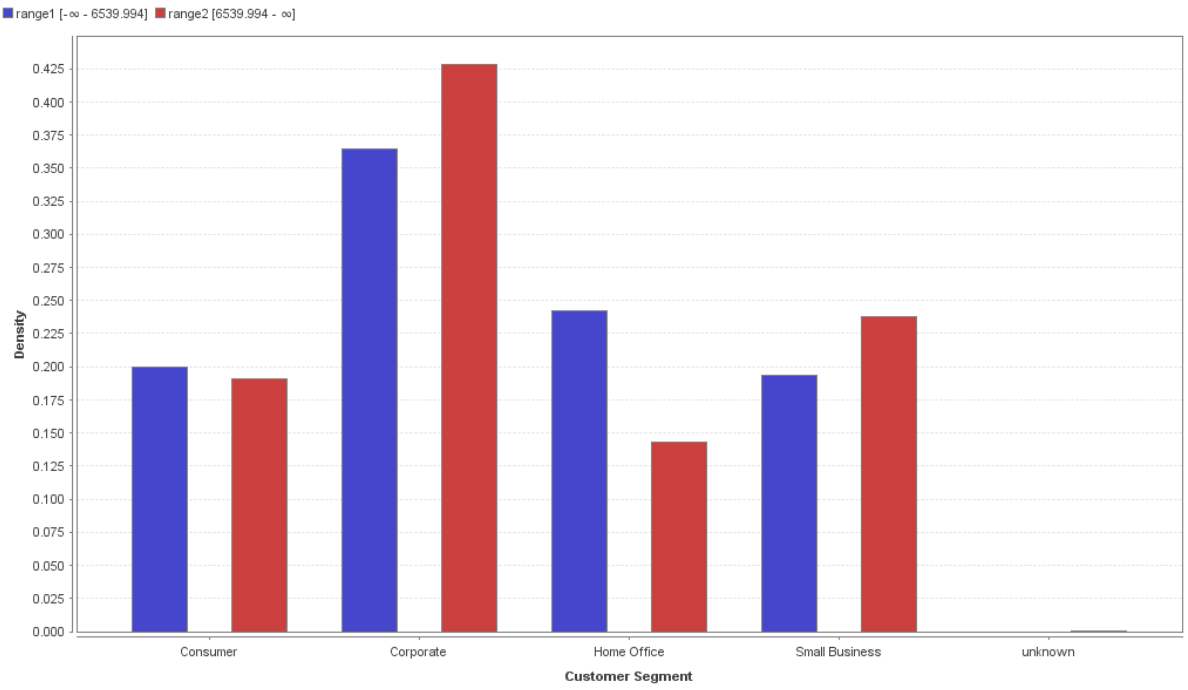


Fig 3.7 Naïve Bayes Histogram of profit based on customer segment

The range 2 profit bin represented extremely high profits (more than €6539 per item) compared to range 1 (less than €6539). A particularly noteworthy example extracted from the Naïve Bayes charts was that of customer segment in relation to profit (See fig 3.7). Upon interpretation of this histogram, one can infer that the corporate customer segment was significantly high for profits in the upper range (€6359 or more per item). Indeed, even profits

in the lower corporate range were significantly higher than all of the other customer segments. Perhaps because corporations tend to be notably large, they purchase more expensive products and in greater quantities compared to home office buyers or small businesses.

Another interesting finding was profit based on order priority. Here, orders of the lowest priority were in fact, those which generated the most profit out of *all* priority categories. Conversely, orders made with the highest critical priority were found to accrue the lowest profits in range 2 (See appendices Fig 3.8).

Model 4

Predicting product category choice based on customer segment with Neural Networks

Pre-processing stage

Firstly, I plugged the training and testing datasets into the process field. I was interested in building and testing the validity of a model that would predict a customer's choice of product category based on what segment they belonged to. After some consideration and because my variables of interest were polynominal, I felt a Neural Network would accomplish this. Neural Networks do not run if there are attributes in the training dataset which are absent in the testing dataset. Therefore, to match datasets I omitted row ID, order ID, order date, ship date, customer name, sales, and profit via *Select Attributes*.

With sales and profit now omitted from the prospective model, I felt it necessary to make the model more comprehensive by creating a new attribute I called "Order cost". This attribute would be based on: order quantity (times) unit price (plus) shipping cost (minus) discount. This new attribute would allow me to infer how the total order cost of an item would affect my overall network. So, using *Generate Attributes*, I assigned the new name (*Order Cost)* and used the functional expression: [Order Quantity]*[Unit Price]+[Shipping Cost]-[Discount]. (See Fig 4.1).

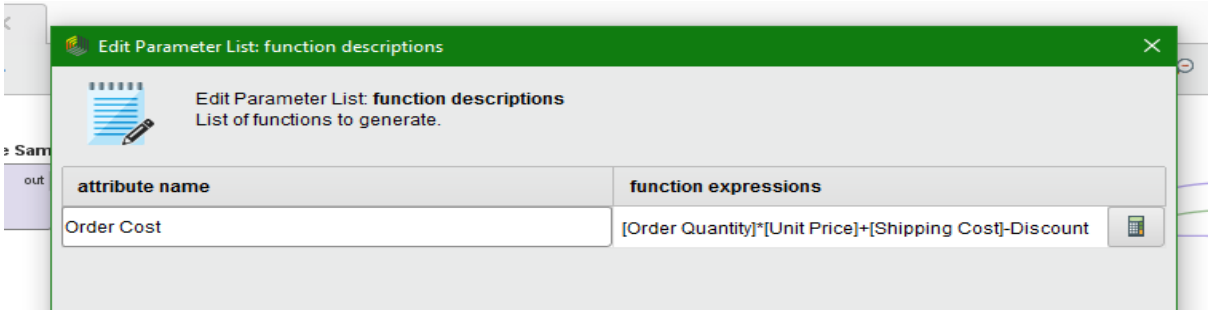


Fig 4.1 Process of generating new Order Cost attribute

This attribute would now feature in both datasets. Next, using *Replace Missing Values* I replaced any missing values by method of averages in both datasets.

Predictive modelling stage

Following pre-processing, I introduced a *Set Role* into both datasets. In the training dataset *Set Role* parameters, I assigned order priority to the attribute ID name, because I wanted to retain this attribute but not necessarily consider it in my model in case it disrupted results. In additional roles, I assigned product category as my target role label as this was the variable I wanted to predict. A product category column was already present in the testing dataset, which would allow me to infer the accuracy of my model (by comparing the predicted with the actual column), and serve as a means of model validation. For the testing dataset *Set Role*, I assigned customer segment as my target role ID, because this is what I wanted my product category choice prediction to be based on. Indeed, for the purpose of this particular analysis, Neural Networks was very suitable because my label attribute had a small number of different variables. Lastly, I attached Apply Model to the Neural Network and testing datasets before running my model (See fig 4.2).

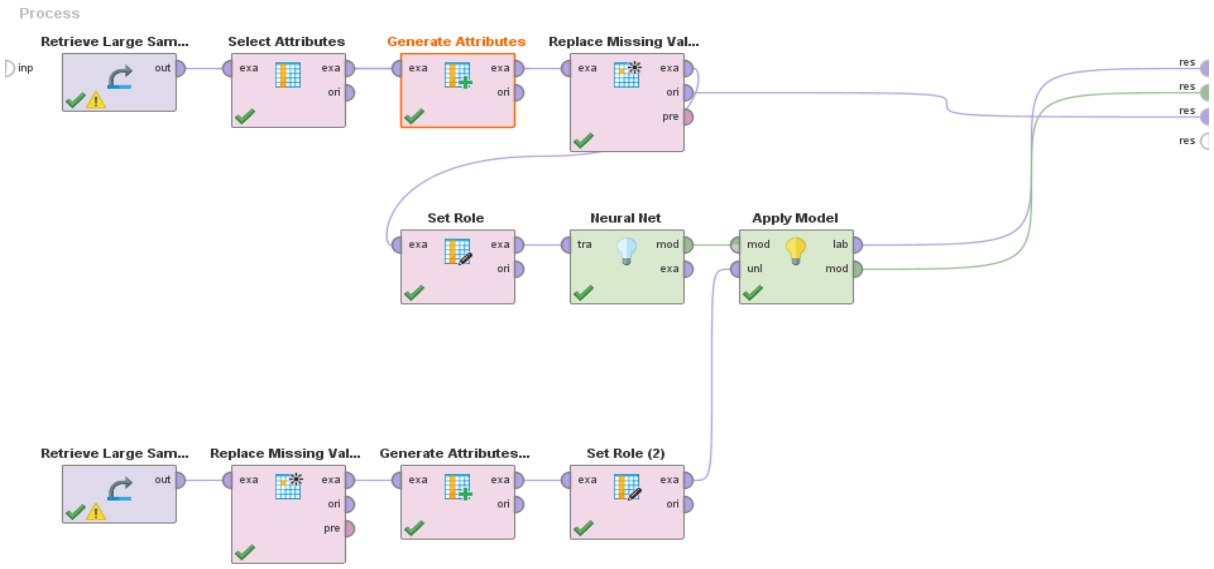


Fig 4.2 Neural Networks main process field

Outcome and interpretation stage

Upon analysis of the Neural Network output, confidence scores for predicting choice of product category were found to be notably high and mostly accurate. This was confirmed by comparing the predicted product category choices with the actual choices (See appendices fig 4.3). Furthermore, by comparing the predicted and actual product category choices via histograms, we can see clearly how accurate the prediction was (See appendices fig 4.4 & 4.5). We can also gauge how strong and associations of the new attribute (Order cost) were by looking at the nodes and neurons of the model graph (See appendix fig 4.6).

The Neural Network model for predicting product category choice based on customer segment was highly accurate and thus, reliable. Nevertheless, it remained limited in its explanatory power due to the absence of certain situational variables and the fact that customer segment alone cannot independently explain why a customer will choose a certain category of product over another.

Appendices

Fig 1.0 Work on subset sub-process field

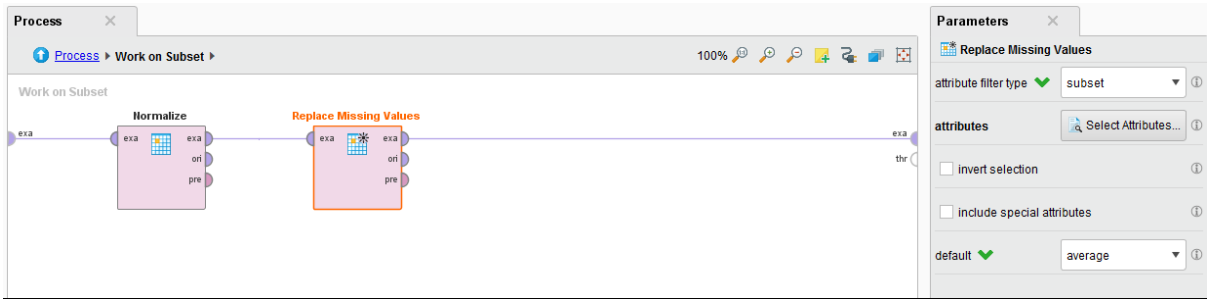


Fig 1.3 Model 1 data output table of predictive model showing predicted and actual values

Exam Display a list of results of this session's processes. attributes) Filter (1,680 / 1,680 examples): all

Row No.	Product Cat...	prediction(Product Cat...	confidence(...	confidence(...	confidence(...	Order Quant...	Discount	Unit Price	Shipping Cost	Order Priority	Custome
1	Technology	Technology	0.084	0.835	0.081	0.305	0.953	0.367	-0.512	High	Corporat
2	Furniture	Office Supplies	0.836	0.060	0.104	-0.317	0.011	-0.285	-0.457	Not Specified	Consum
3	Office Supplies	Office Supplies	0.836	0.060	0.104	-0.938	-0.618	-0.282	-0.296	High	Corporat
4	Office Supplies	Office Supplies	0.559	0.235	0.207	-0.248	1.268	-0.160	-0.383	High	Corporat
5	Office Supplies	Office Supplies	0.836	0.060	0.104	1.272	0.639	-0.290	-0.261	Low	Home Of
6	Office Supplies	Office Supplies	0.836	0.060	0.104	1.341	-1.247	-0.293	-0.385	Medium	Home Of
7	Office Supplies	Office Supplies	0.836	0.060	0.104	-0.662	0.639	-0.253	-0.663	Not Specified	Small Bu
8	Technology	Technology	0.084	0.835	0.081	-1.007	-0.304	0.126	-0.298	High	Home Of
9	Furniture	Furniture	0.049	0	0.951	1.272	0.953	0.010	1.552	Low	Home Of
10	Office Supplies	Office Supplies	0.836	0.060	0.104	0.995	-0.304	-0.285	-0.358	Low	Corporat
11	Furniture	Furniture	0.136	0.102	0.763	-0.248	-0.932	0.643	0.676	Low	Corporat
12	Office Supplies	Office Supplies	0.836	0.060	0.104	1.410	-1.247	-0.301	-0.700	High	Consum
13	Office Supplies	Office Supplies	0.978	0	0.022	-1.283	-0.304	0.655	1.285	Not Specified	Corporat
14	Office Supplies	Office Supplies	0.836	0.060	0.104	1.272	0.639	-0.293	-0.715	Medium	Corporat
15	Office Supplies	Office Supplies	0.559	0.235	0.207	1.686	-1.247	-0.252	0.020	Critical	Corporat
16	Furniture	Furniture	0.015	0	0.985	0.236	-1.247	0.595	2.892	Not Specified	Corporat
17	Office Supplies	Office Supplies	0.836	0.060	0.104	1.410	1.268	-0.301	-0.703	Low	Corporat

Fig 1.4 Decision tree predicting Product Category choice

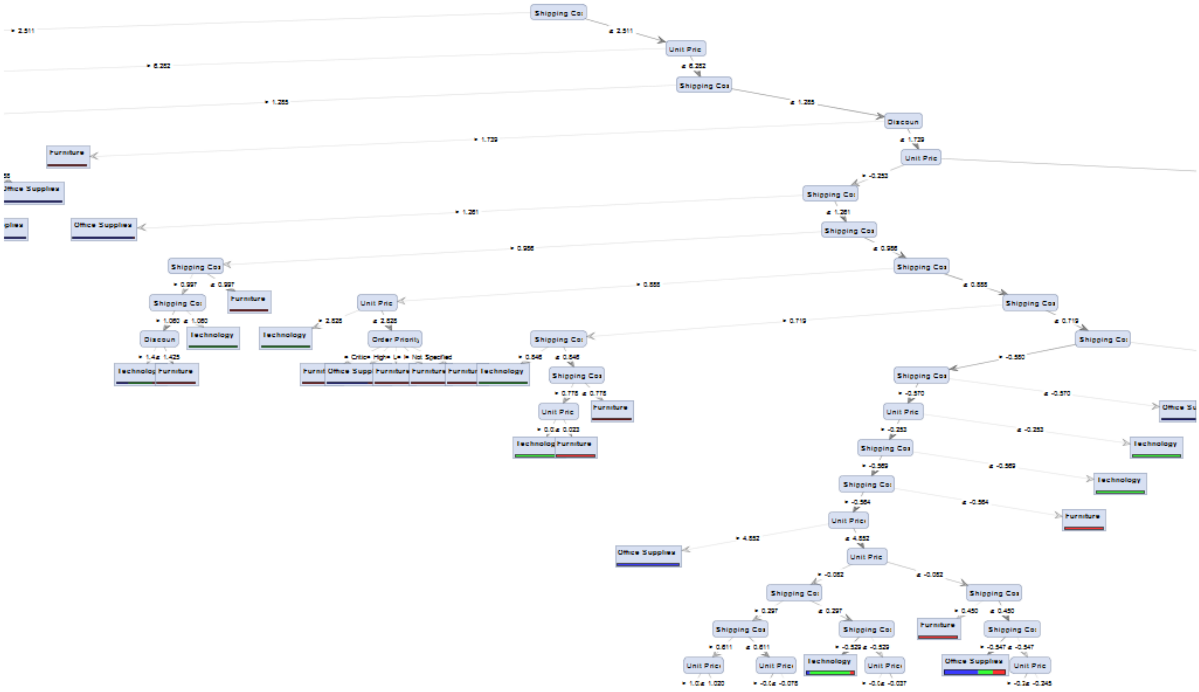


Fig 2.0 Selecting attributes to keep datasets similar

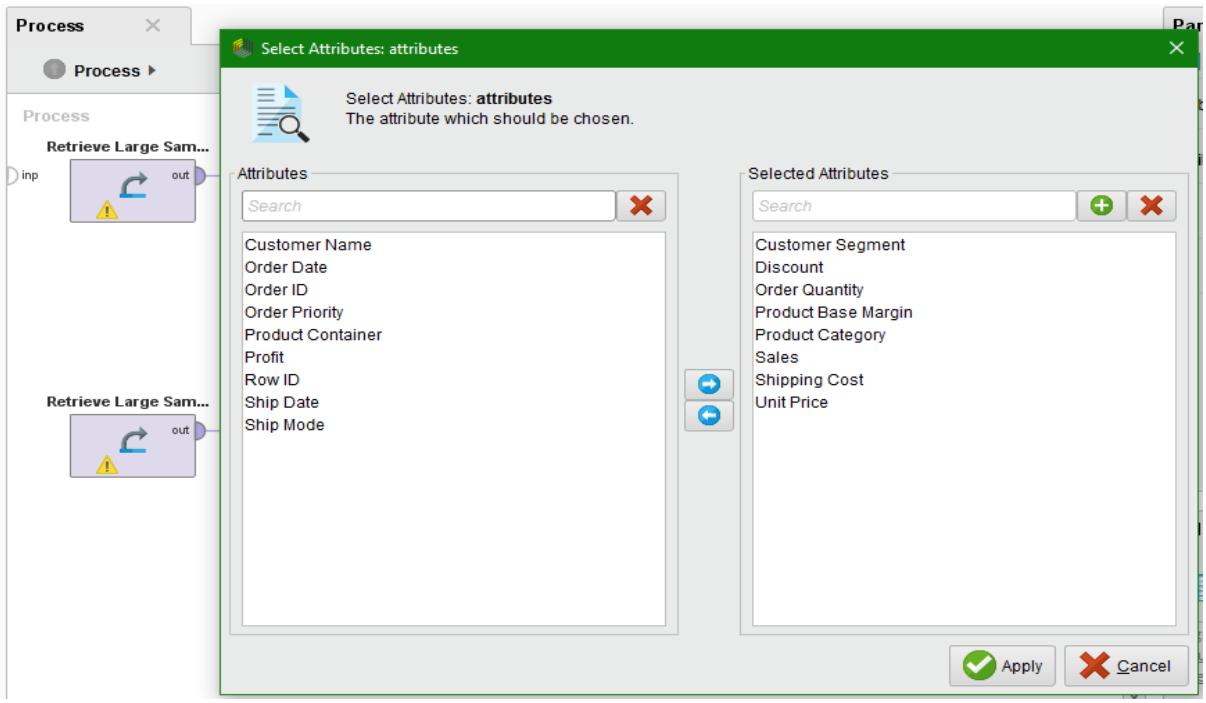


Fig 2.1 Using filter examples to adjust ranges across datasets

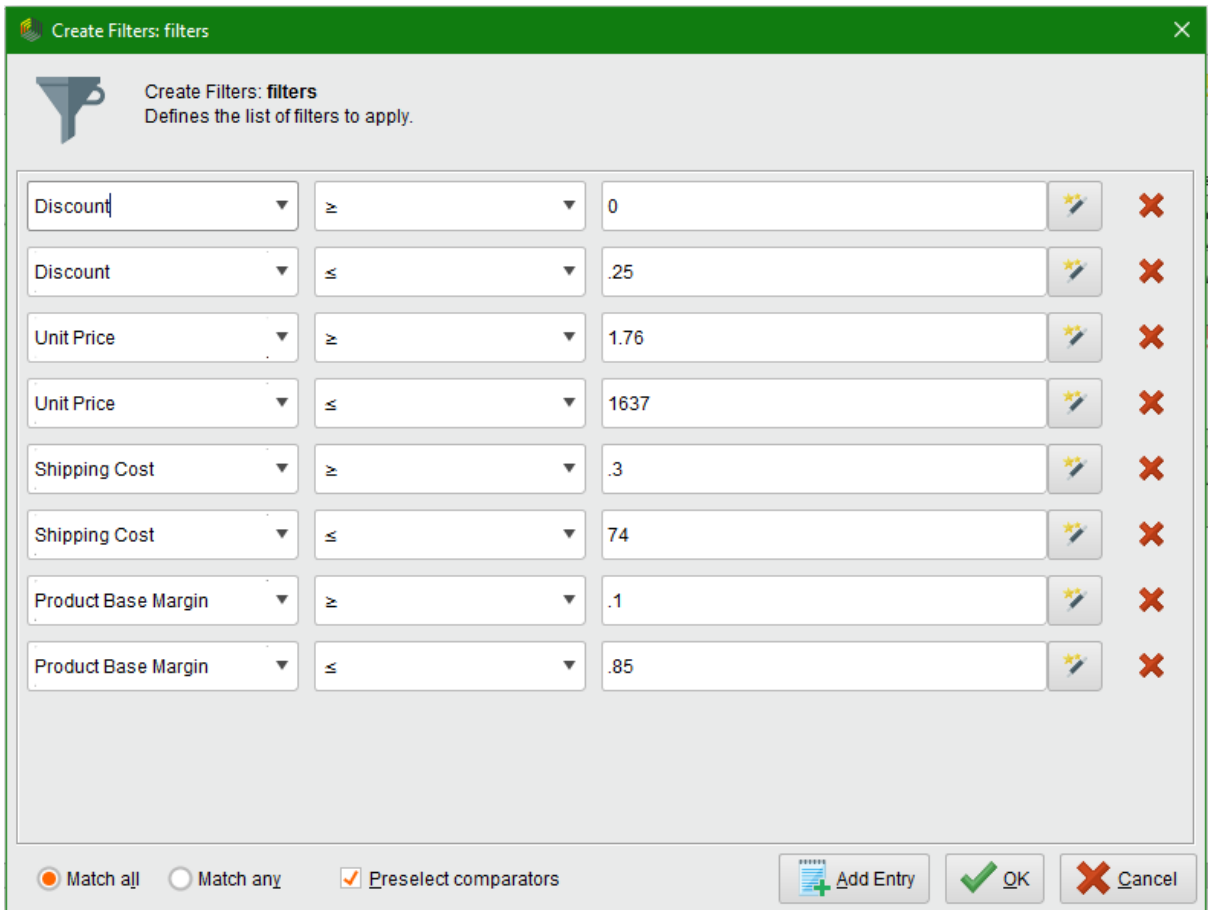
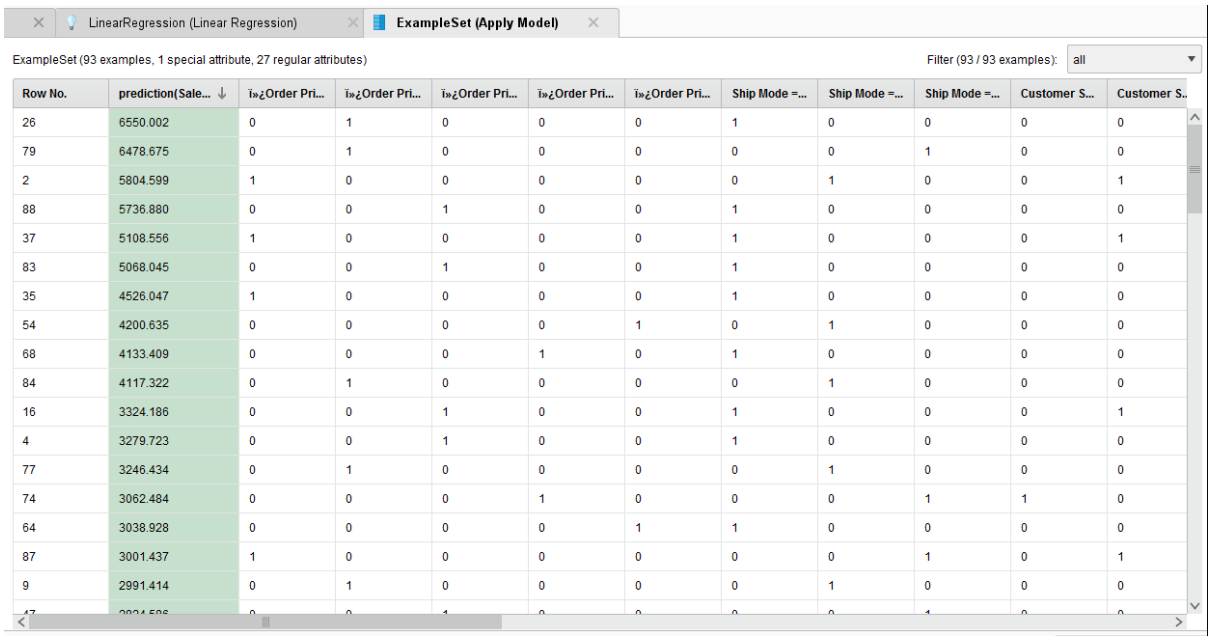


Fig 2.2 Linear Regression coefficient table with significance values

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value ↑	Code
Product Category = Technology	1162.035	69.090	0.140	0.991	16.819	0	****
Order Quantity	69.458	1.835	0.280	0.993	37.860	0	****
Unit Price	6.524	0.097	0.528	0.937	67.319	0	****
Shipping Cost	68.199	1.934	0.328	0.893	35.261	0	****
(Intercept)	-1280.903	124.550	?	?	-10.284	0	****
Product Base Margin	-805.914	224.571	-0.030	0.942	-3.589	0.000	****
Product Category = Furniture	162.794	82.162	0.018	0.952	1.981	0.048	**
Discount	-1574.536	830.439	-0.014	1.000	-1.896	0.058	*
Customer Segment = Small Business	-24.540	66.565	-0.003	1.000	-0.369	0.712	

Fig 2.4 Linear regression data table output



LinearRegression (Linear Regression) | ExampleSet (Apply Model)

ExampleSet (93 examples, 1 special attribute, 27 regular attributes) Filter (93 / 93 examples): all

Row No.	prediction(Sale...	ï»¿Order Pri...	ï»¿Order Pri...	ï»¿Order Pri...	ï»¿Order Pri...	ï»¿Order Pri...	Ship Mode =...	Ship Mode =...	Ship Mode =...	Customer S...	Customer S...
26	6550.002	0	1	0	0	0	1	0	0	0	0
79	6478.675	0	1	0	0	0	0	0	1	0	0
2	5804.599	1	0	0	0	0	0	1	0	0	1
88	5736.880	0	0	1	0	0	1	0	0	0	0
37	5108.556	1	0	0	0	0	1	0	0	0	1
83	5068.045	0	0	1	0	0	1	0	0	0	0
35	4526.047	1	0	0	0	0	1	0	0	0	0
54	4200.635	0	0	0	0	1	0	1	0	0	0
68	4133.409	0	0	0	1	0	1	0	0	0	0
84	4117.322	0	1	0	0	0	0	1	0	0	0
16	3324.186	0	0	1	0	0	1	0	0	0	1
4	3279.723	0	0	1	0	0	1	0	0	0	0
77	3246.434	0	1	0	0	0	0	1	0	0	0
74	3062.484	0	0	0	1	0	0	0	1	1	0
64	3038.928	0	0	0	0	1	1	0	0	0	0
87	3001.437	1	0	0	0	0	0	0	1	0	1
9	2991.414	0	1	0	0	0	0	1	0	0	0

Fig 3.0 Sub-process 1 Loop Values preparation

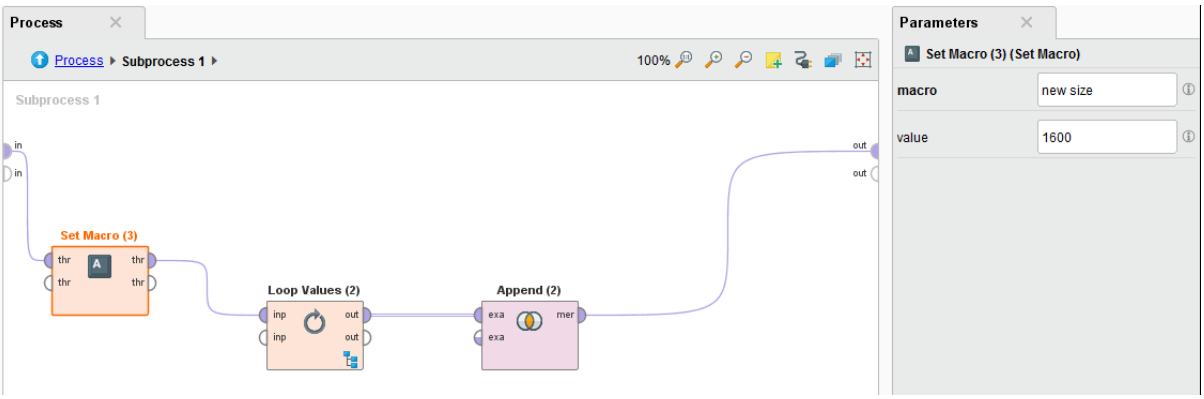


Fig 3.1 Assigning a filter to Loop Value operator

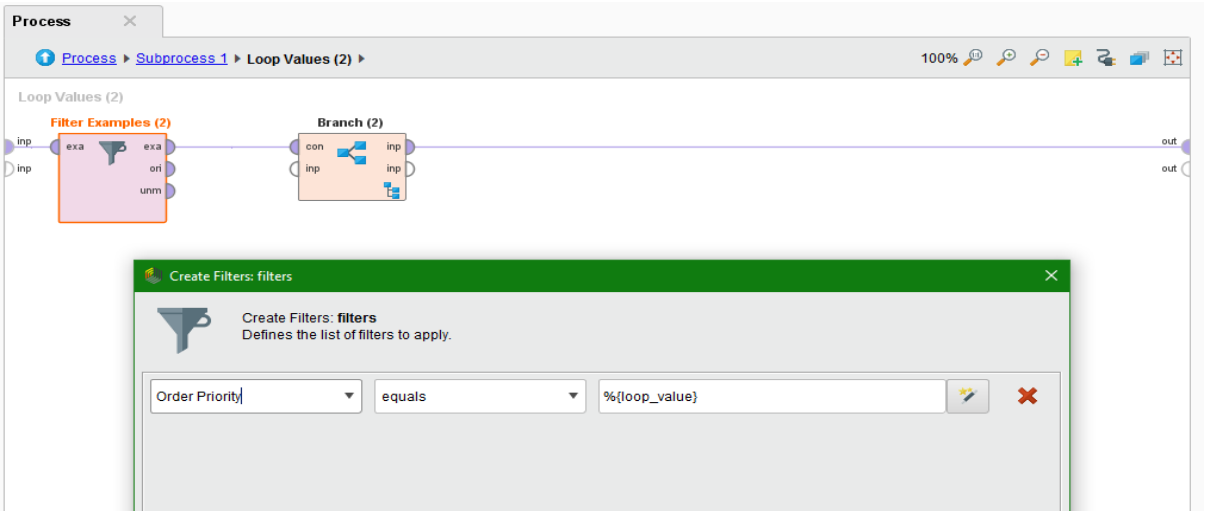


Fig 3.2 Branch operator process field

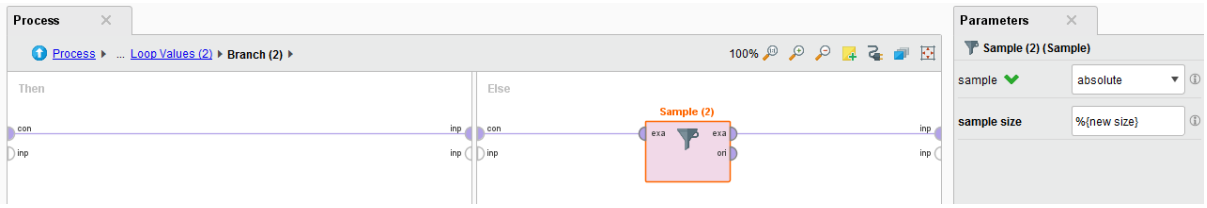


Fig 3.5 Cross Validation training and testing sub-process operators

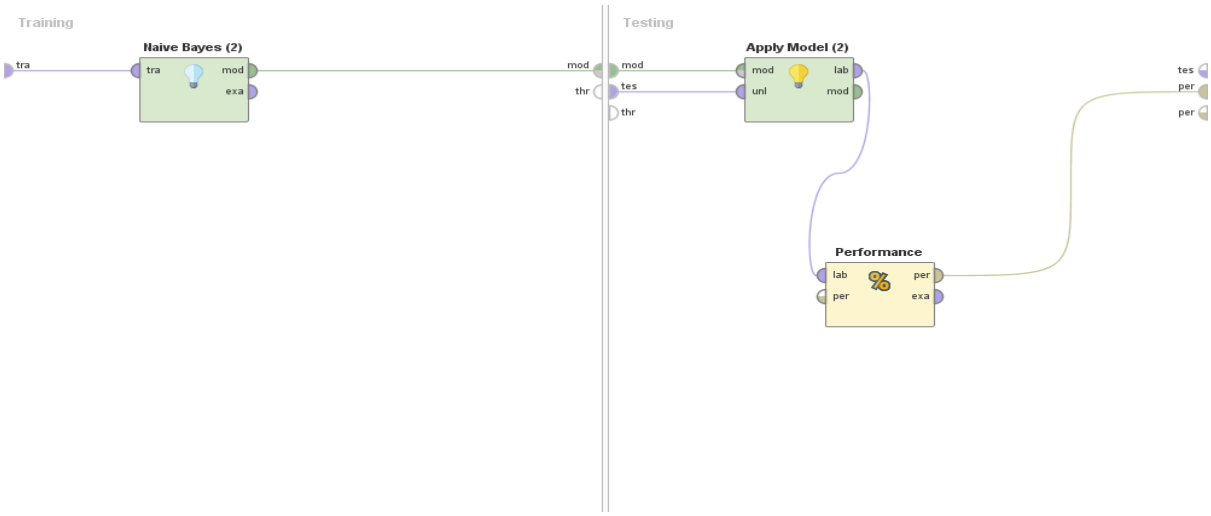
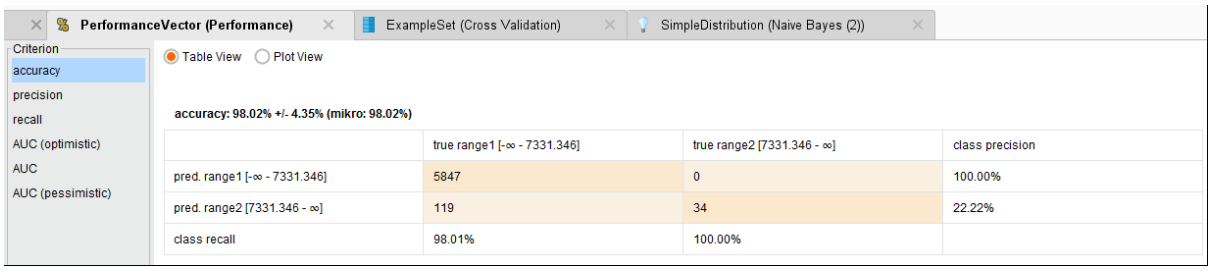


Fig 3.6 Performance vector table Naïve Bayes model of predicting profit



	true range1 [-∞ - 7331.346]	true range2 [7331.346 - ∞]	class precision
pred. range1 [-∞ - 7331.346]	5847	0	100.00%
pred. range2 [7331.346 - ∞]	119	34	22.22%
class recall	98.01%	100.00%	

Fig 3.7 Profits based on order priority

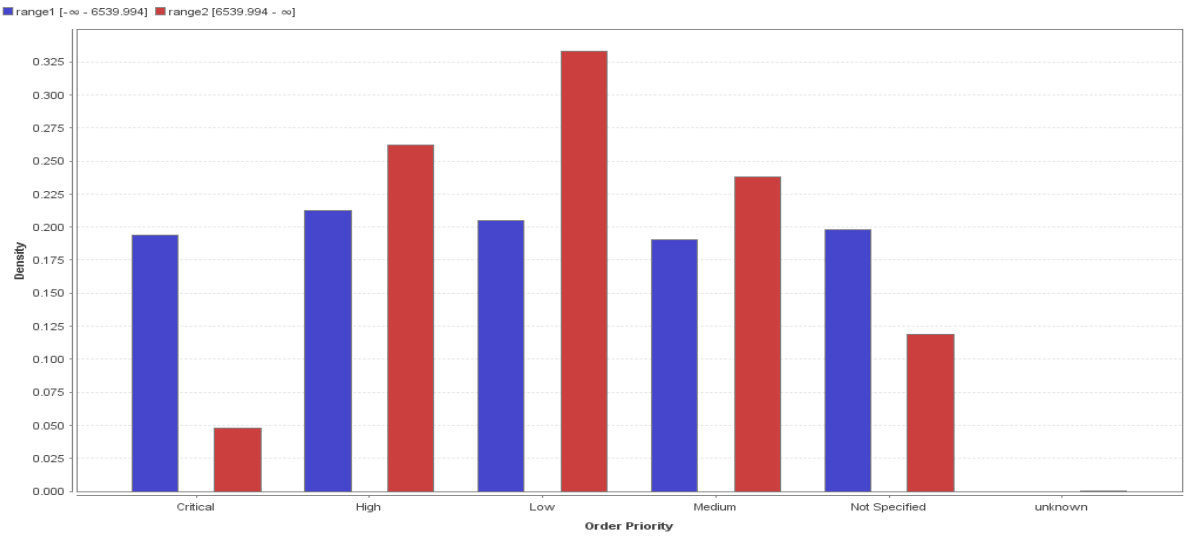
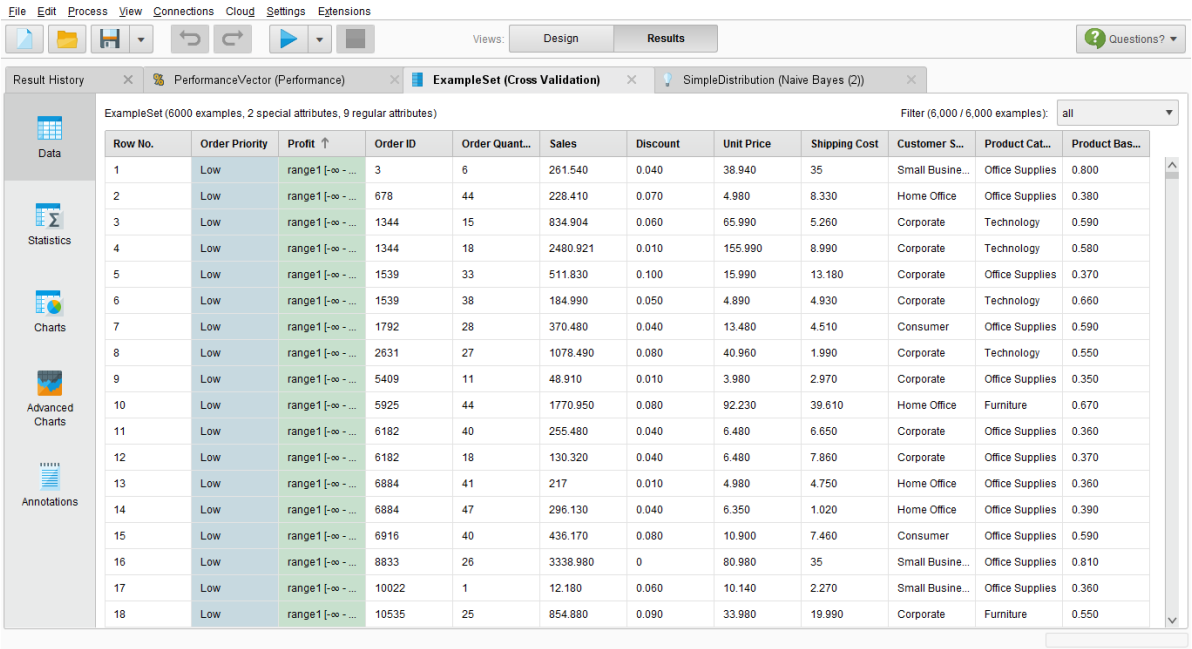
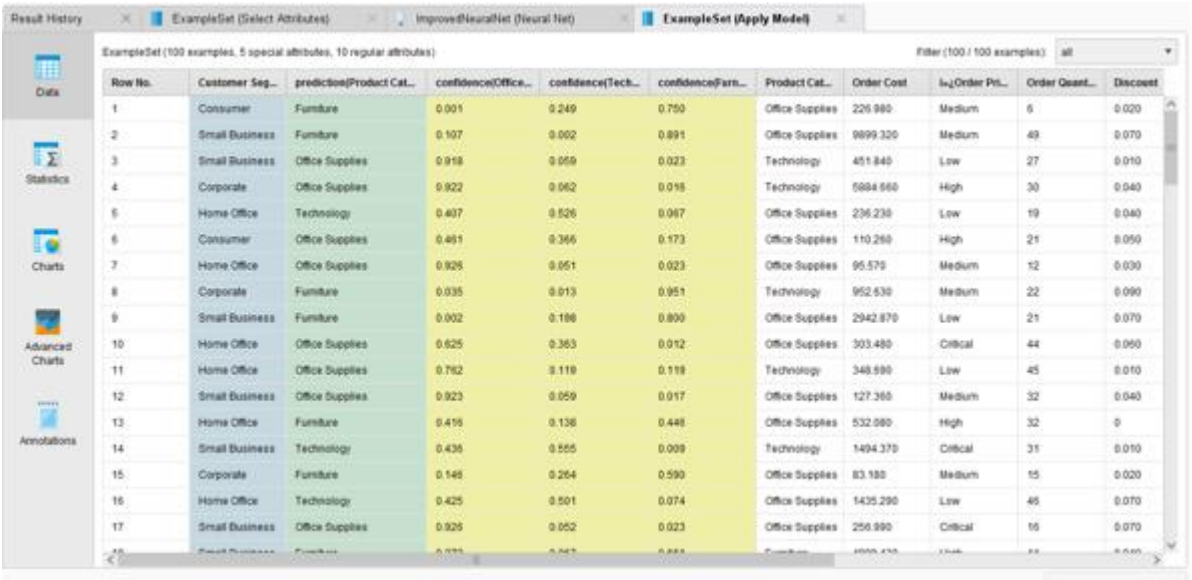


Fig 3.8 Data output table of predictive model for Profit levels with Order Priority IDs



Row No.	Order Priority	Profit	Order ID	Order Quant...	Sales	Discount	Unit Price	Shipping Cost	Customer S...	Product Cat...	Product Bas...
1	Low	range1 [-∞ - ...	3	6	261.540	0.040	38.940	35	Small Busine...	Office Supplies	0.800
2	Low	range1 [-∞ - ...	678	44	228.410	0.070	4.980	8.330	Home Office	Office Supplies	0.380
3	Low	range1 [-∞ - ...	1344	15	834.904	0.060	65.990	5.260	Corporate	Technology	0.590
4	Low	range1 [-∞ - ...	1344	18	2480.921	0.010	155.990	8.990	Corporate	Technology	0.580
5	Low	range1 [-∞ - ...	1539	33	511.830	0.100	15.990	13.180	Corporate	Office Supplies	0.370
6	Low	range1 [-∞ - ...	1539	38	184.990	0.050	4.890	4.930	Corporate	Technology	0.660
7	Low	range1 [-∞ - ...	1792	28	370.480	0.040	13.480	4.510	Consumer	Office Supplies	0.590
8	Low	range1 [-∞ - ...	2631	27	1078.490	0.080	40.960	1.990	Corporate	Technology	0.550
9	Low	range1 [-∞ - ...	5409	11	48.910	0.010	3.980	2.970	Corporate	Office Supplies	0.350
10	Low	range1 [-∞ - ...	5925	44	1770.950	0.080	92.230	39.610	Home Office	Furniture	0.670
11	Low	range1 [-∞ - ...	6182	40	255.480	0.040	6.480	6.650	Corporate	Office Supplies	0.360
12	Low	range1 [-∞ - ...	6182	18	130.320	0.040	6.480	7.860	Corporate	Office Supplies	0.370
13	Low	range1 [-∞ - ...	6884	41	217	0.010	4.980	4.750	Home Office	Office Supplies	0.360
14	Low	range1 [-∞ - ...	6884	47	296.130	0.040	6.350	1.020	Home Office	Office Supplies	0.390
15	Low	range1 [-∞ - ...	6916	40	436.170	0.080	10.900	7.460	Consumer	Office Supplies	0.590
16	Low	range1 [-∞ - ...	8833	26	3338.980	0	80.980	35	Small Busine...	Office Supplies	0.810
17	Low	range1 [-∞ - ...	10022	1	12.180	0.060	10.140	2.270	Small Busine...	Office Supplies	0.360
18	Low	range1 [-∞ - ...	10535	25	854.880	0.090	33.980	19.990	Corporate	Furniture	0.550

Fig 4.3 Neural Network data output with predicted product category choices by customer segment, confidence scores, and actual product category choices.



Row No.	Customer Seg...	prediction(Product Cat...	confidence(Office...	confidence(Tech...	confidence(Furn...	Product Cat...	Order Cost	Order Pri...	Order Quant...	Discount
1	Consumer	Furniture	0.001	0.249	0.750	Office Supplies	226.980	Medium	6	0.020
2	Small Business	Furniture	0.107	0.002	0.891	Office Supplies	9899.320	Medium	49	0.070
3	Small Business	Office Supplies	0.918	0.059	0.023	Technology	451.840	Low	27	0.010
4	Corporate	Office Supplies	0.922	0.062	0.016	Technology	5884.660	High	30	0.040
5	Home Office	Technology	0.407	0.526	0.067	Office Supplies	236.230	Low	19	0.040
6	Consumer	Office Supplies	0.461	0.366	0.173	Office Supplies	110.260	High	21	0.050
7	Home Office	Office Supplies	0.926	0.051	0.023	Office Supplies	95.570	Medium	12	0.030
8	Corporate	Furniture	0.035	0.013	0.951	Technology	952.630	Medium	22	0.090
9	Small Business	Furniture	0.002	0.198	0.800	Office Supplies	2942.870	Low	21	0.070
10	Home Office	Office Supplies	0.625	0.363	0.012	Office Supplies	303.480	Critical	44	0.060
11	Home Office	Office Supplies	0.762	0.119	0.119	Technology	348.680	Low	45	0.010
12	Small Business	Office Supplies	0.923	0.059	0.017	Office Supplies	127.360	Medium	32	0.040
13	Home Office	Furniture	0.416	0.138	0.446	Office Supplies	532.080	High	32	0
14	Small Business	Technology	0.436	0.555	0.009	Technology	1494.370	Critical	31	0.010
15	Corporate	Furniture	0.146	0.264	0.590	Office Supplies	83.180	Medium	15	0.020
16	Home Office	Technology	0.425	0.501	0.074	Office Supplies	1435.290	Low	46	0.070
17	Small Business	Office Supplies	0.926	0.052	0.023	Office Supplies	256.990	Critical	16	0.070

Fig 4.4. *Actual* Product Category choice based on Customer Segment

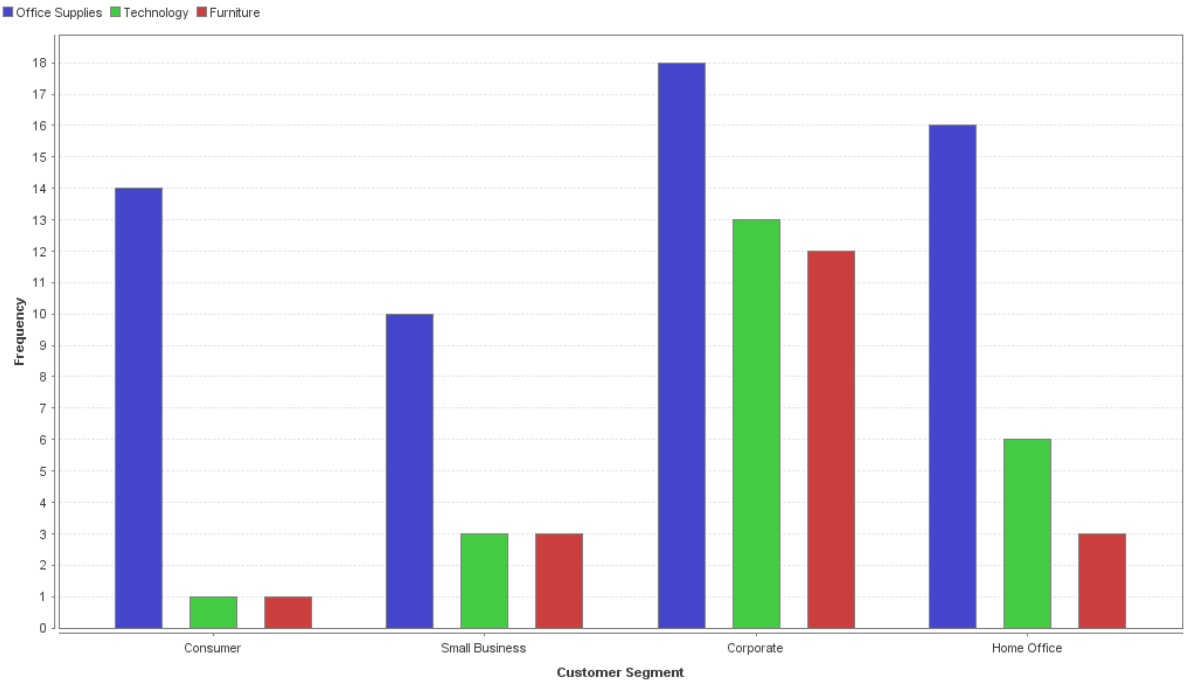


Fig 4.5 *Predicted* Product Category choice based on Customer Segment

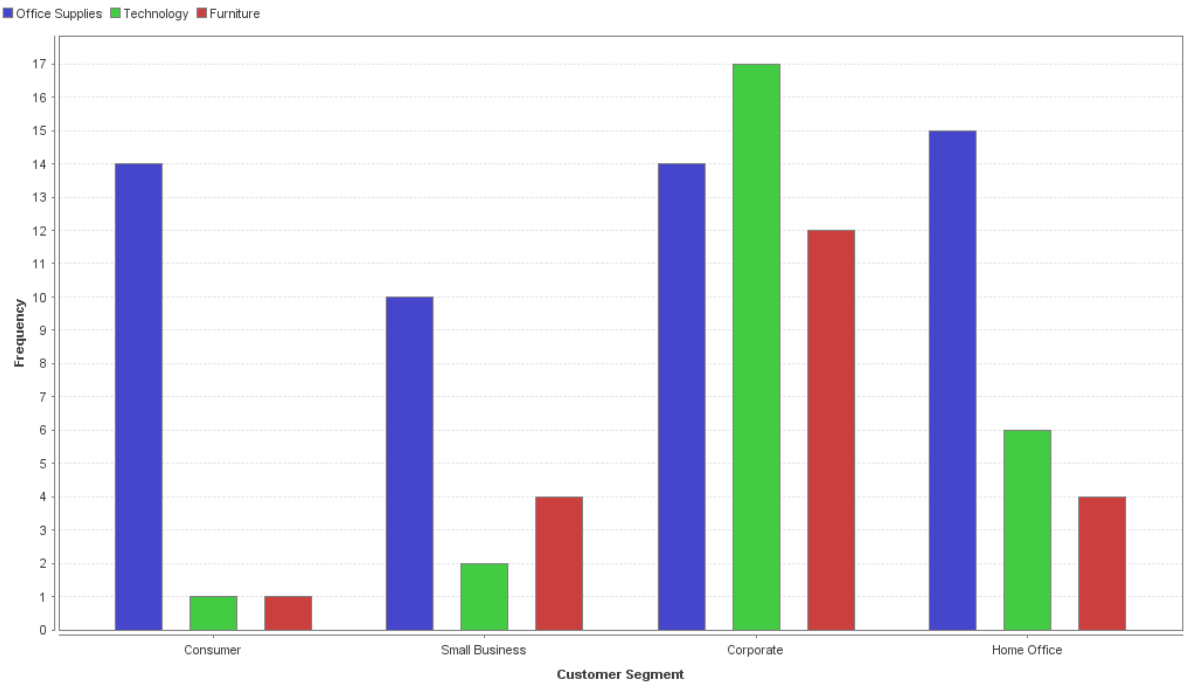


Fig 4.6 Neural Network output graph