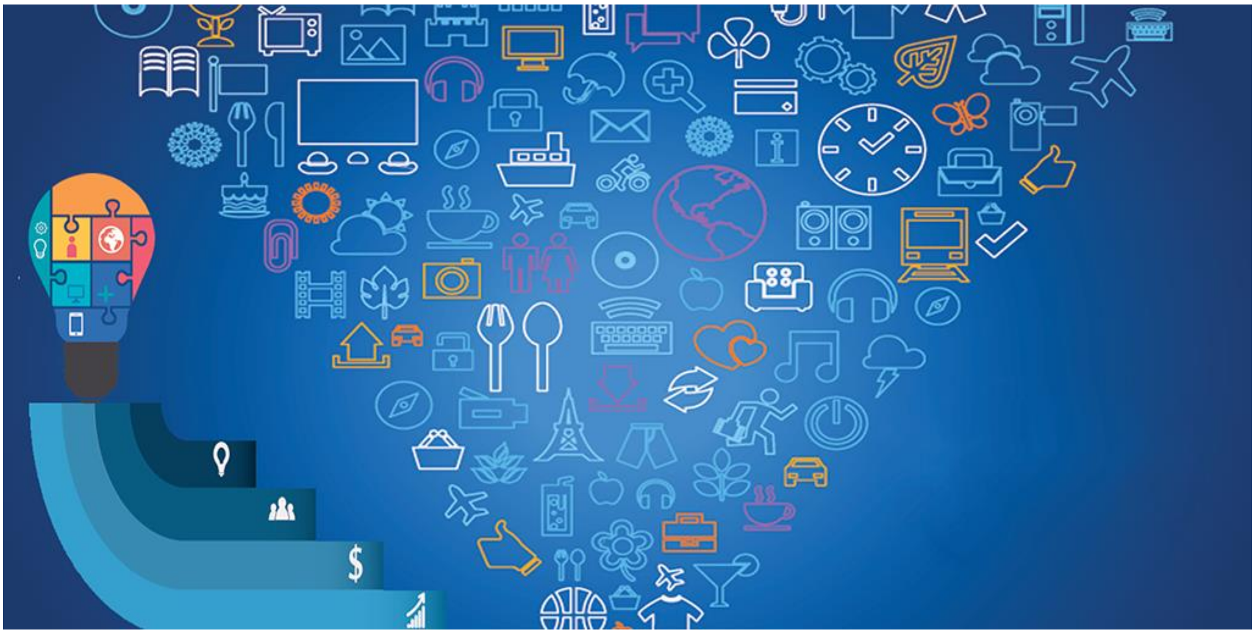
Last Name: Kushwanshi First Name: Anupriya

DATA MINING – INDIVIDUAL CASE 3

Professor: Yan Yu



Submitted By:

Anupriya Kushwanshi

Goal and Background

We have three problems at hand here.

1. **European Employment Data – Clustering**
   My goal is to cluster the dataset into meaningful number clusters and describe major differences in terms of employment structure between countries belonging to different clusters, using mean percentage of employment in each sector from amongst countries belonging to each cluster.
2. **Cincinnati Zoo Data – Association Rules**
   My goal is to replicate the zoo project using association rules using dataset food_4_association.xls
3. **Classification using SPSS Modeler**
   My goal is to create a classification model on Bankruptsy data using SPSS modeler.

Approach

I'll use RStudio and SPSS tools to accomplish the goals.

For Clustering:

- I did data cleaning as the data is in txt. I'll bring that into xlsx format and properly space using data tools in MS Excel.
- Random sample a data set that contains 90% of original data points.
- Perform K-means clustering and Hierarchical clustering.
- Choose optimal value of clusters.
- Calculate mean percentage of employment in each sector for each cluster and observe.

For Association Rules:

- I cleaned the data by converting all the levels into logical datatype.
- Found out the most frequent items in the data set
- Identified which items are important in the data set
- plot the item frequency for items with a support greater than 10%.
- Found all rules with a minimum support of 0.3% and a confidence of 0.5.
- Found the subset of rules with lift greater than 5:
- Visualized the rules

For Classification using SPSS Modeler:

- Imported Bankruptsy data and determined the target variable.
- Partitioned the data into 75% train and 25% test sets and ran GLM, Neural Net, and CART model
- Studied AUC, ROC, in-sample and out-of-sample performance.

Major findings

Chose 4 clusters as optimal for **problem 1**. Countries in cluster 2 have most of the population working in Agricultural field (which is highest among the clusters) and least in manufacturing. Cluster 4 has most of it's population working in SPS which is highest among the clusters. Clusters 3 and 4 are similar yet different if we see Agr and SPS industries. Cluster 1 has the largest population and 45% population works in Agr and Man industries.

In **problem 2,** we observed that the people who were buying 2 or more things together were very likely to buy French fries basket.

In **problem 3,** GLM and CART models gave 100% AUC while Neural Network gave 99.9%.

Clustering

I am using Employment data for clustering. The data has the following fields:

- Country: Name of country
- Agr: Percentage employed in agriculture
- Min: Percentage employed in mining
- Man: Percentage employed in manufacturing
- PS: Percentage employed in power supply industries
- Con: Percentage employed in construction
- SI: Percentage employed in service industries
- Fin: Percentage employed in finance
- SPS: Percentage employed in social and personal services
- TC: Percentage employed in transport and communications

There were 9 NA values in the data. It also had an extra field without a header so I removed it and took a sample of 90% data.

K – means clustering

The idea of k-means clustering is to define clusters then minimize the total intra-cluster variation (known as total within-cluster variation). In order to use k-means method for clustering and plot results, we can use kmeans function in R. It will group the data into a specified number of clusters.

Performed clustering at k = 2,3,4,5

# Clusters	Values in cluster 1	Values in cluster 2	Values in cluster 3	Values in cluster 4	Values in cluster 5	Values in cluster 6	Values in cluster 7
2	11	4					
3	7	7	1				
4	6	1	4	4			
5	3	4	1	4	3		
6	1	3	1	1	5	4	
7	3	4	1	2	1	1	3

Table 1: Number of items in each cluster for all values of k 2,3,4,5

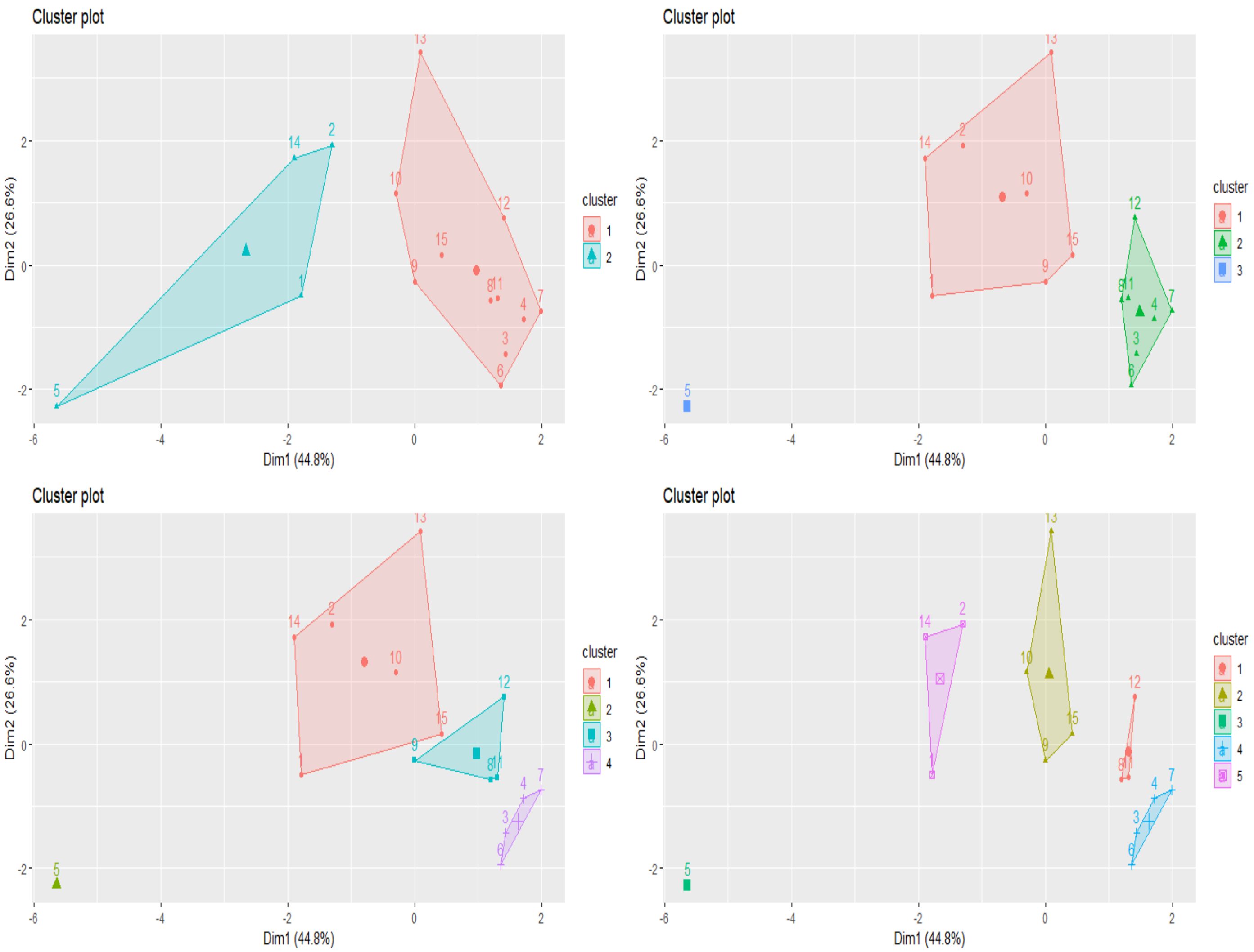


Image 1: Cluster plot of all k = 2,3,4,5

We can see that there is very little overlap among the clusters. Hence, I tried clustering with k= 6 and 7 and the results don't look satisfying.

Table 1: Number of items in each cluster for the values of k 6,7

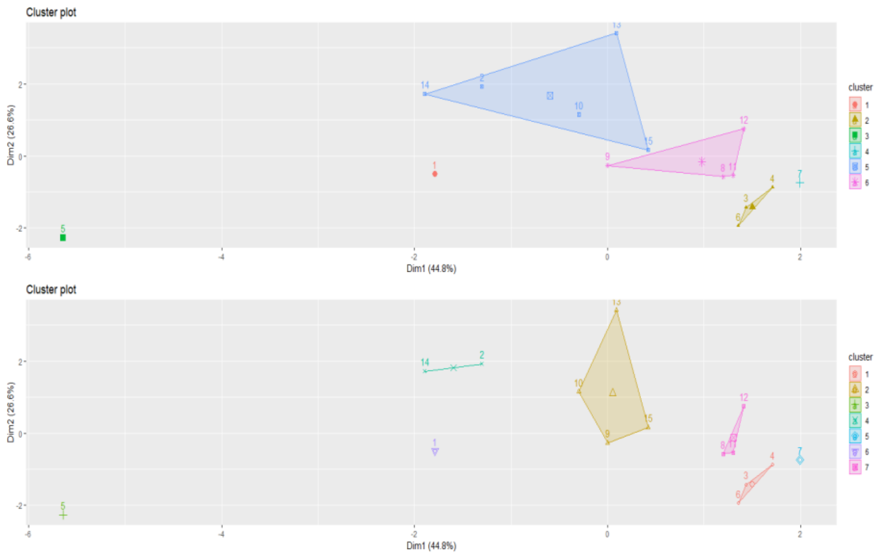


Image 2: Cluster plot for k = 6,7.

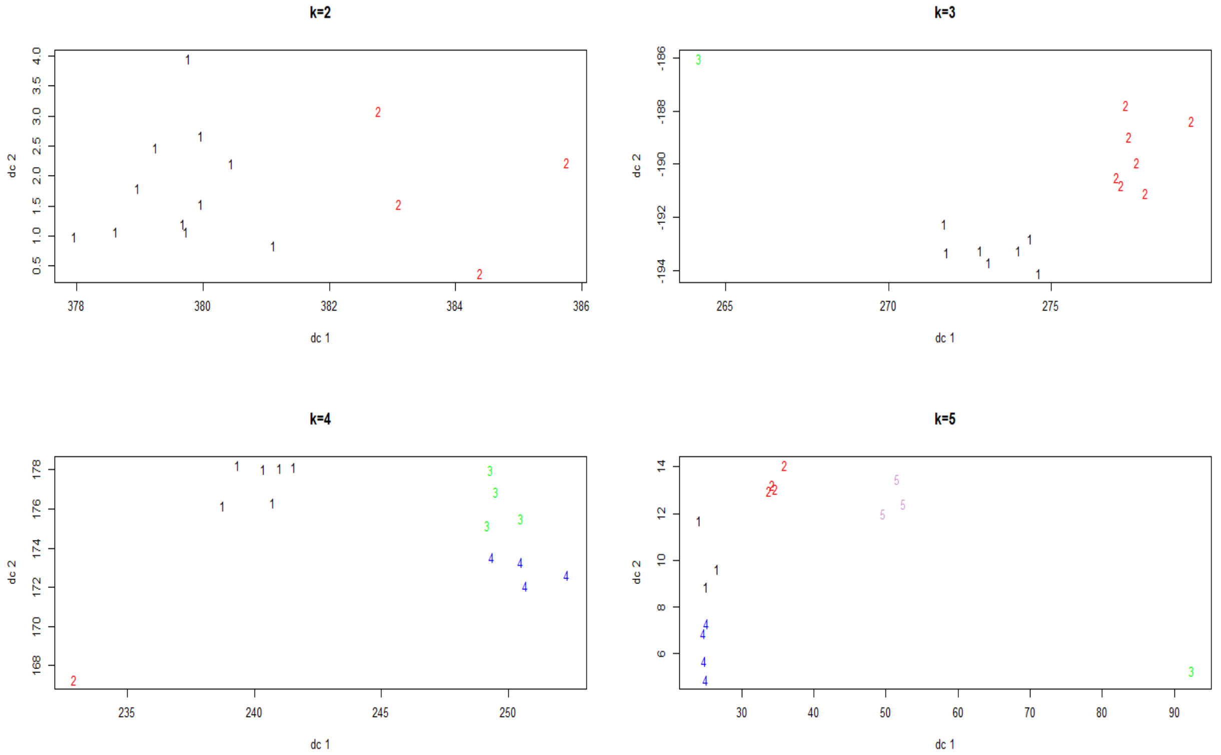


Image 3: The distribution of items in each cluster in the below figure.

Selecting optimal value of k using elbow method:

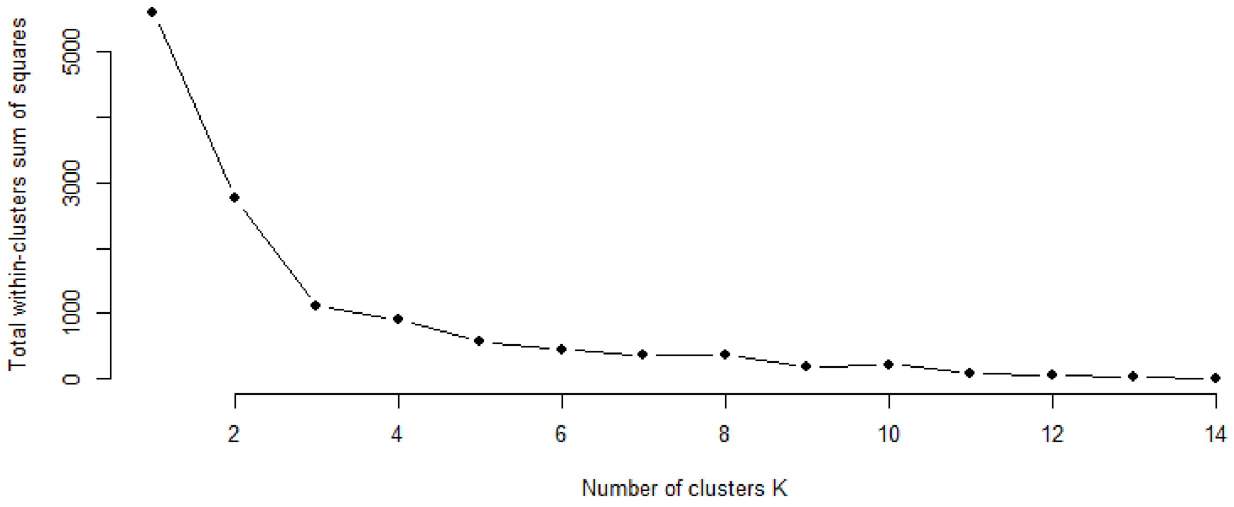


Image 4: Elbow method of optimal # cluster selection

After k=4, we can see the variance is not decreasing as much so I will choose this one.

Employment percentage in each cluster:

Cluster 1

Industry	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Mean Employment rate	29.25	1.75	26.21	0.88	9.01	8.35	2.41	15.23	6.9

Table 2: Employment percentage in cluster 1

Cluster 2

Industry	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Mean Employment rate	66.8	0.7	7.9	0.1	2.8	5.2	1.1	11.9	3.2

Table 3: Employment percentage in cluster 2

Cluster 3

Industry	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Mean Employment rate	14.92	0.83	26.07	1.22	8.2	16.27	4.80	21.12	6.6

Table 3: Employment percentage in cluster 3

Cluster 4

Industry	Agr	Min	Man	PS	Con	SI	Fin	SPS	TC
Mean Employment rate	6.9	0.47	24.42	0.77	8.07	16.25	5.85	29.70	7.62

Table 4: Employment percentage in cluster 4

Countries in cluster 2 has most of the population working in Agricultural field (which is highest among the clusters) and least in manufacturing. Cluster 4 has most of it's population working in SPS which is highest among the clusters. Clusters 3 and 4 are similar yet different if we see Agr and SPS industries. Cluster 1 has the largest population and 45% population works in Agr and Man industries.

Hierarchical clustering.

Hierarchical clustering is a bottom up approach which does not require that we commit to a particular choice of K. Hierarchical clustering has an added advantage over K-means clustering in that it results in an attractive tree-based representation of the observations, called a dendrogram. [Reference: ISLR seventh print textbook]

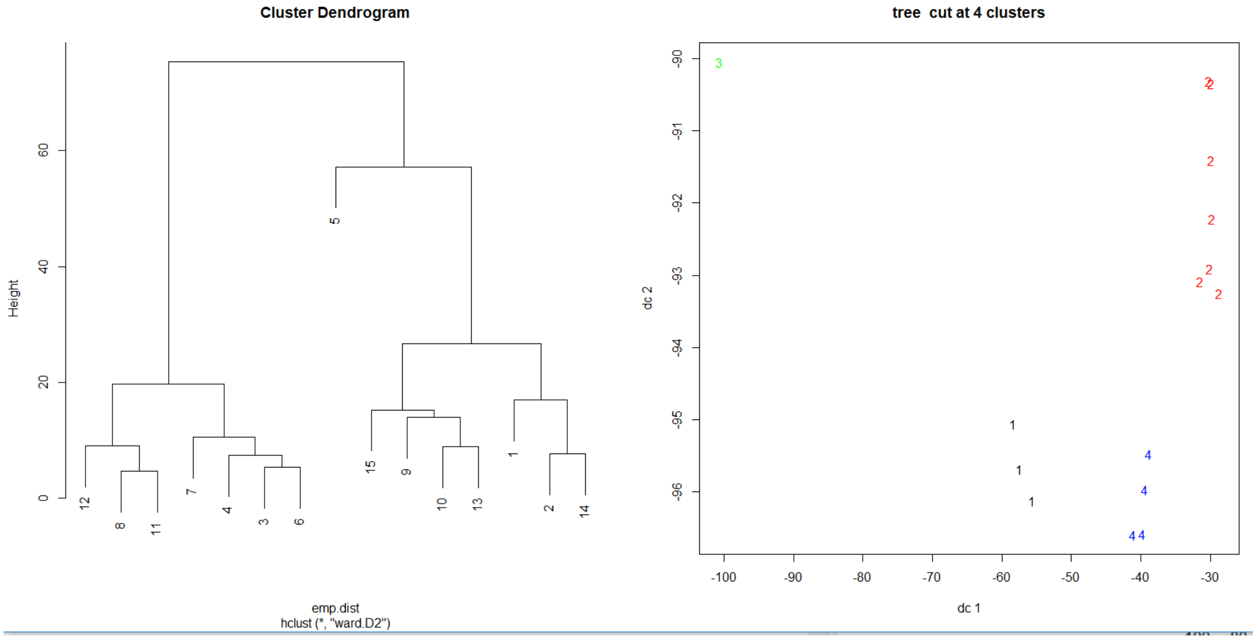


Image 5: Hierarchical clustering

These clusters are similar to the ones generated in k-means

Association Rules

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. The ultimate goal, assuming a large enough dataset, is to help a machine mimic the human brain's feature extraction and abstract association capabilities from new uncategorized data. [Reference: https://en.wikipedia.org/wiki/Association_rule_learning]

Apriori algorithm:

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.

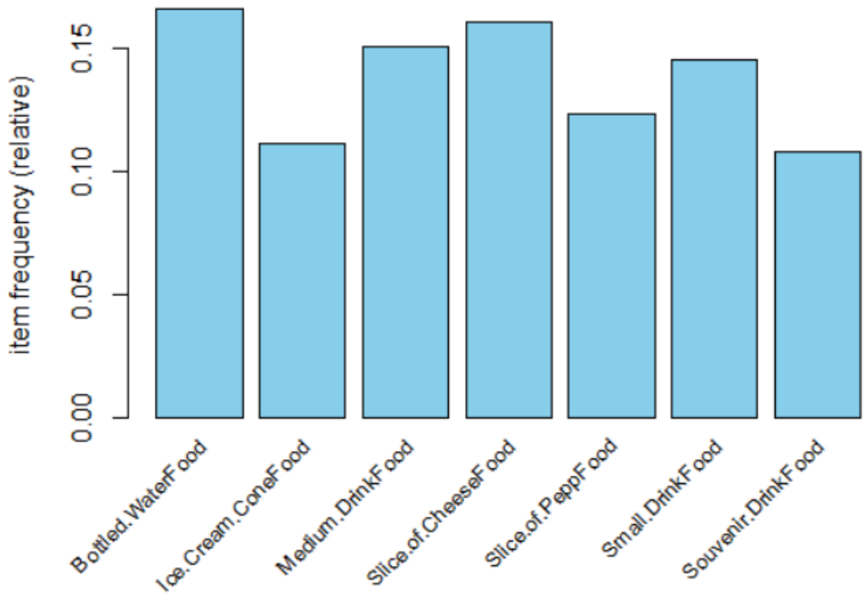
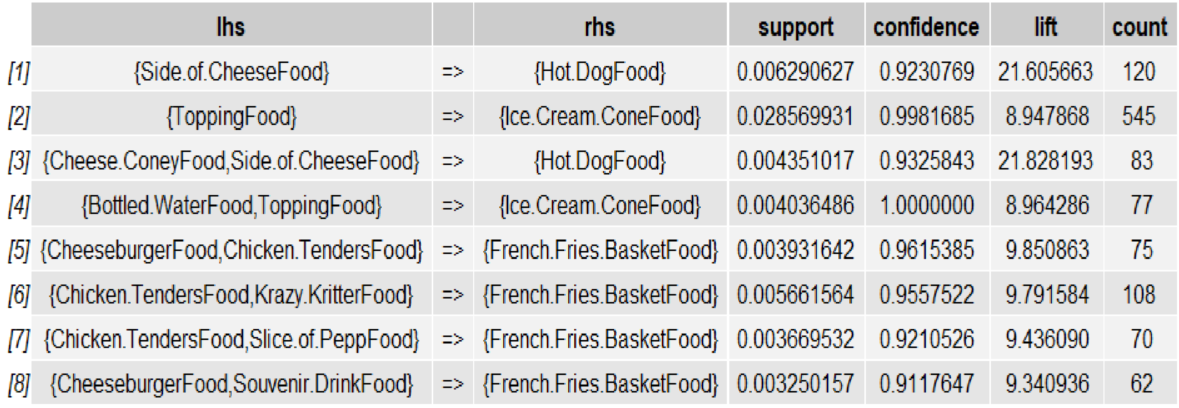


Image 6: Item frequency

The most frequent item bought is Bottled water and least frequent among these is ice cream cone food and souvenir drink food.

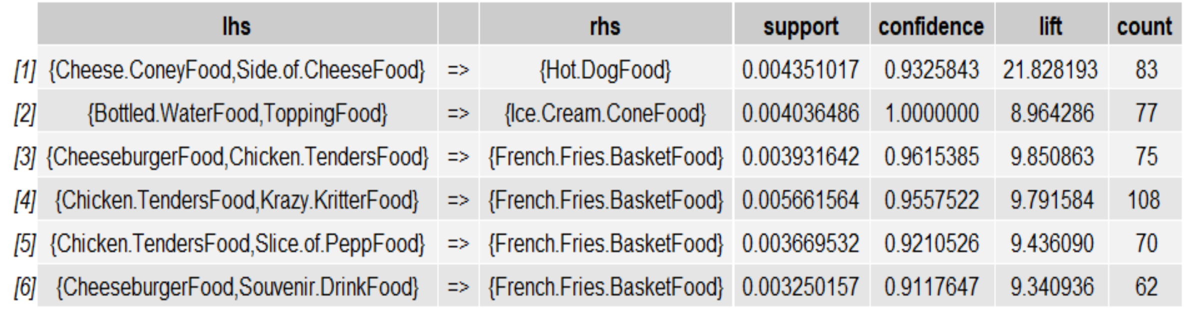
8 rules were applied on the entire dataset.



	lhs		rhs	support	confidence	lift	count
[1]	{Side.of.CheeseFood}	=>	{Hot.DogFood}	0.006290627	0.9230769	21.605663	120
[2]	{ToppingFood}	=>	{Ice.Cream.ConeFood}	0.028569931	0.9981685	8.947868	545
[3]	{Cheese.ConeyFood,Side.of.CheeseFood}	=>	{Hot.DogFood}	0.004351017	0.9325843	21.828193	83
[4]	{Bottled.WaterFood,ToppingFood}	=>	{Ice.Cream.ConeFood}	0.004036486	1.0000000	8.964286	77
[5]	{CheeseburgerFood,Chicken.TendersFood}	=>	{French.Fries.BasketFood}	0.003931642	0.9615385	9.850863	75
[6]	{Chicken.TendersFood,Krazy.KritterFood}	=>	{French.Fries.BasketFood}	0.005661564	0.9557522	9.791584	108
[7]	{Chicken.TendersFood,Slice.of.PeppFood}	=>	{French.Fries.BasketFood}	0.003669532	0.9210526	9.436090	70
[8]	{CheeseburgerFood,Souvenir.DrinkFood}	=>	{French.Fries.BasketFood}	0.003250157	0.9117647	9.340936	62

Image 7: Association rules

I inspected with Basket rules of size greater than 2.



	lhs		rhs	support	confidence	lift	count
[1]	{Cheese.ConeyFood,Side.of.CheeseFood}	=>	{Hot.DogFood}	0.004351017	0.9325843	21.828193	83
[2]	{Bottled.WaterFood,ToppingFood}	=>	{Ice.Cream.ConeFood}	0.004036486	1.0000000	8.964286	77
[3]	{CheeseburgerFood,Chicken.TendersFood}	=>	{French.Fries.BasketFood}	0.003931642	0.9615385	9.850863	75
[4]	{Chicken.TendersFood,Krazy.KritterFood}	=>	{French.Fries.BasketFood}	0.005661564	0.9557522	9.791584	108
[5]	{Chicken.TendersFood,Slice.of.PeppFood}	=>	{French.Fries.BasketFood}	0.003669532	0.9210526	9.436090	70
[6]	{CheeseburgerFood,Souvenir.DrinkFood}	=>	{French.Fries.BasketFood}	0.003250157	0.9117647	9.340936	62

Image 8: Association rules where LHS>=2

We can see that confidence is pretty much high for French Fries Basket Food in the consequent. This shows people who are buying 2 or more things together are very likely to buy French fries basket.

Matrix-based visualization techniques organize the antecedent and consequent itemsets on the x and y-axes, respectively. A selected interest measure is displayed at the intersection of the antecedent and consequent of a given rule. If no rule is available for a antecedent/consequent combination the intersection area is left blank.

[Reference: https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf]

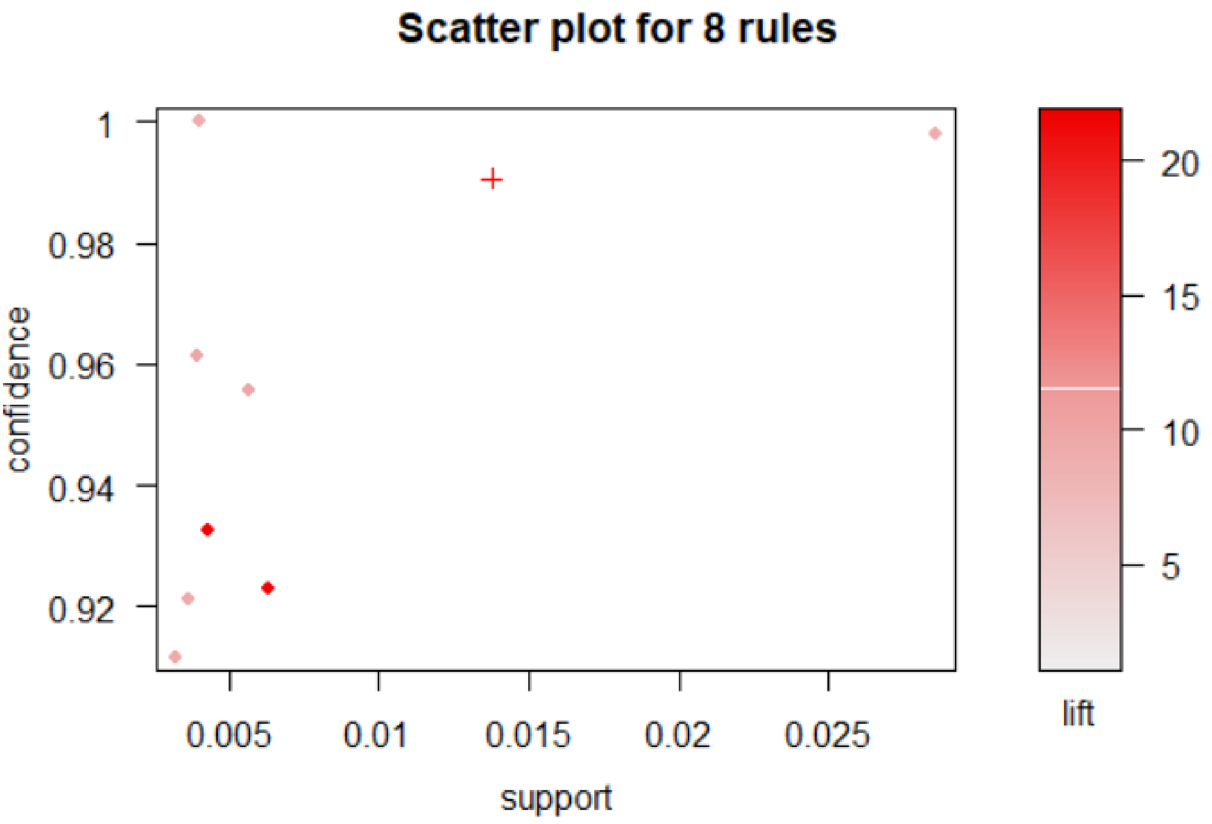


Image 9: Scatterplot for all the rules

Below graph shows the association between items bought together. The vertices are represented by items for the 8 rules with highest lift. The width of the arrows represents support and the intensity of the color represent confidence. For larger rule sets visual analysis becomes difficult since with an increasing number of rules also the number of crossovers between the lines increases

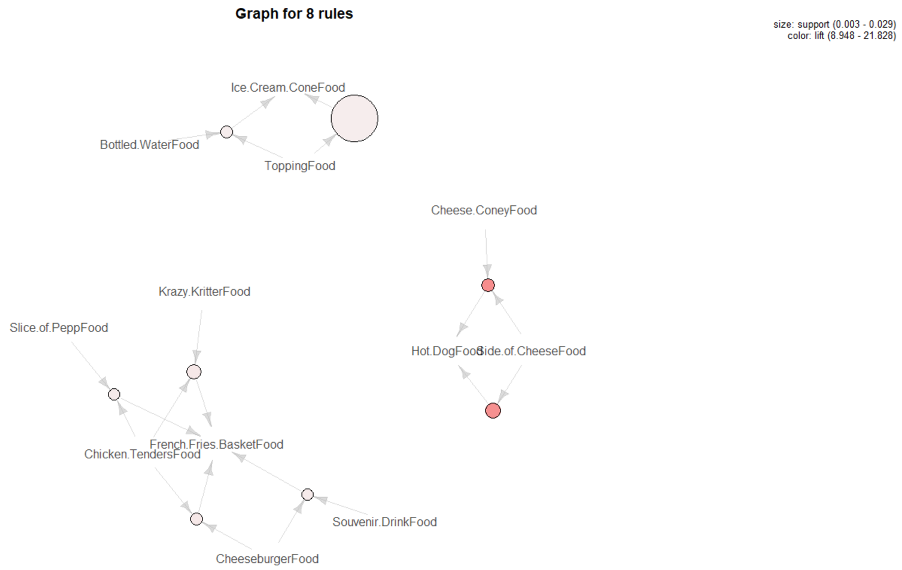


Image 10: Association between items

Last Name: Kushwanshi First Name: Anupriya

Classification – SPSS

Executive Summary

I am using Bankruptsy data where the response variable is binary. I am trying to predict chances of getting bankrupt. I am building a SPSS model for the same. I set the response variable as target and filtered the column that wasn't needed. I created three models:

- CART
- GLM
- Neural Network

Data Dictionary

R1=Working Capital/Total Asset

R2=Retained Earning/Total Asset

R3=Earning Before Interest & Tax/Total Asset

R4=Market Capital / Total Liability

R5=SALE/Total Asset

R6=Total Liability/Total Asset

R7=Current Asset/Current Liability

R8=Net Income/Total Asset

R9=LOG(SALE)

R10=LOG(Market Cap)

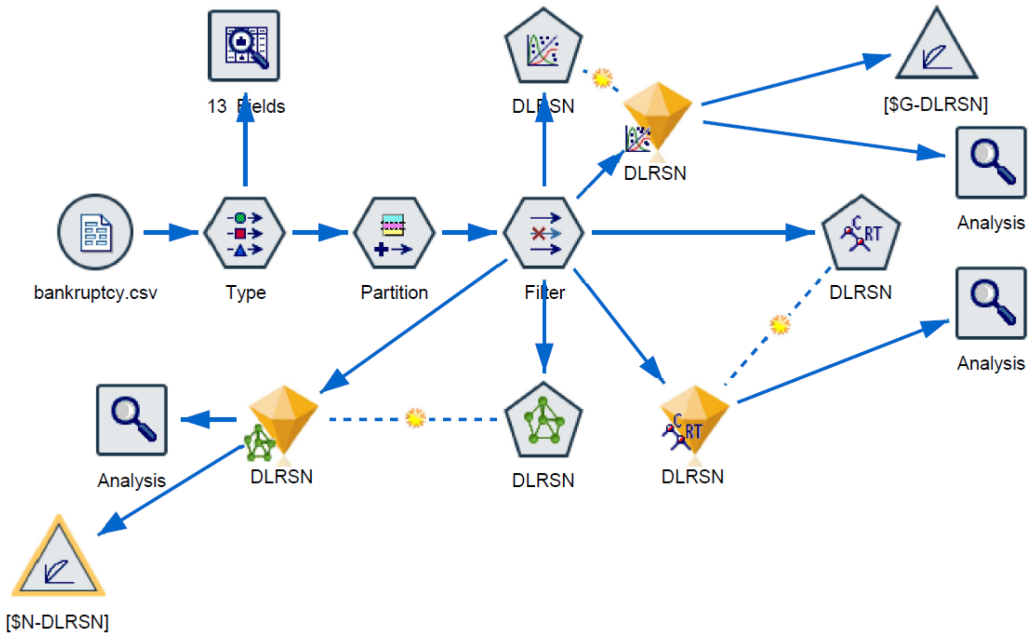
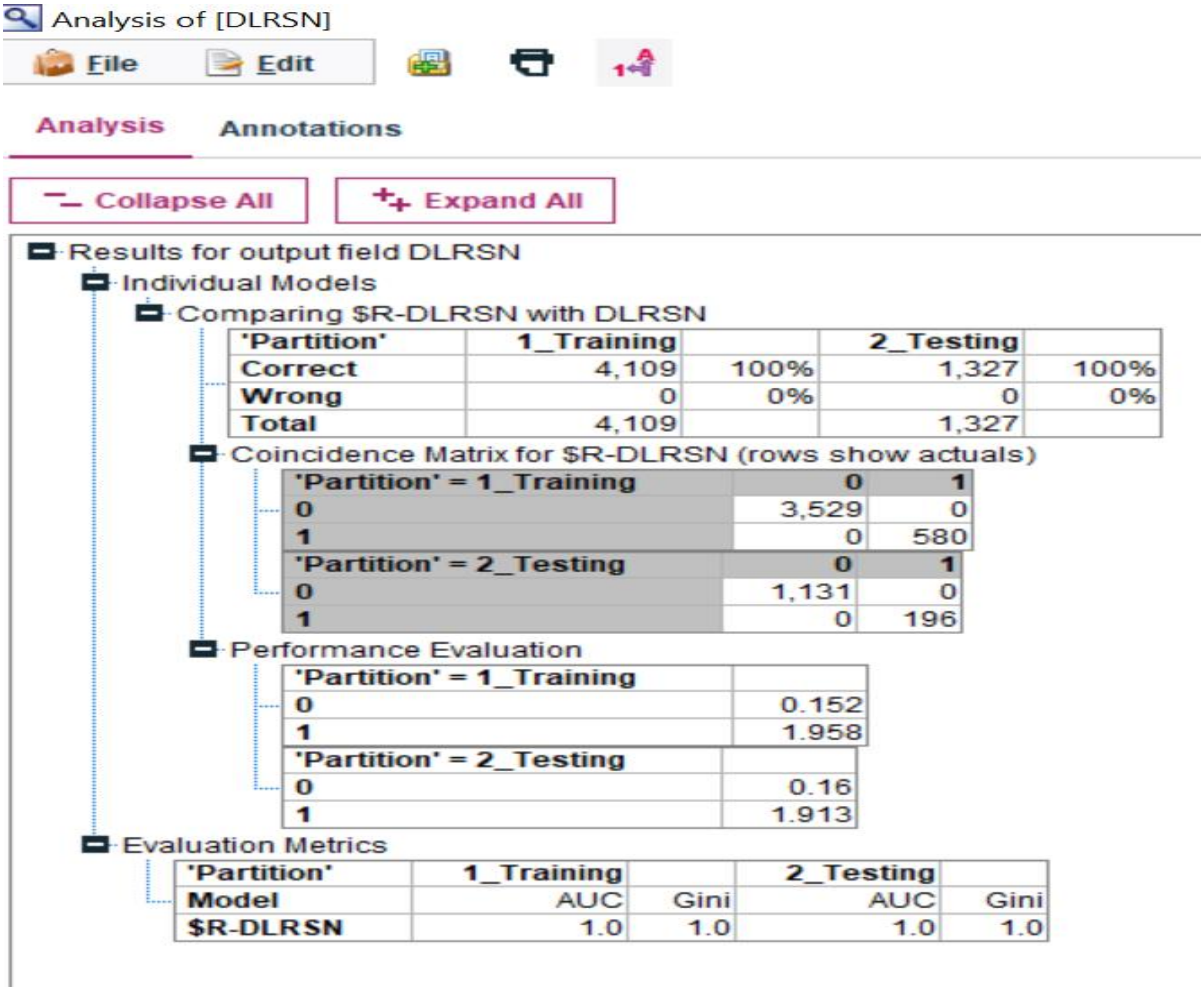


Image 11: SPSS model diagram

CART



Analysis of [DLRSN]

Results for output field DLRSN

Individual Models

Comparing $R-DLRSN with DLRSN

'Partition'	1_Training		2_Testing	
Correct	4,109	100%	1,327	100%
Wrong	0	0%	0	0%
Total	4,109		1,327	

Coincidence Matrix for $R-DLRSN (rows show actuals)

'Partition' = 1_Training	0	1
0	3,529	0
1	0	580

'Partition' = 2_Testing	0	1
0	1,131	0
1	0	196

Performance Evaluation

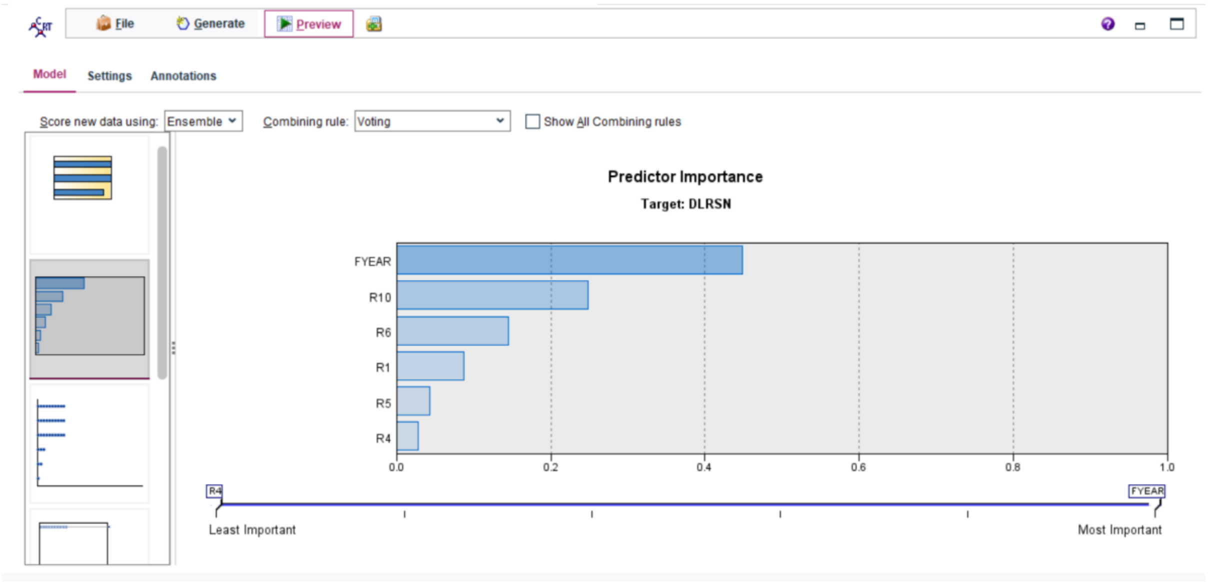
'Partition' = 1_Training	
0	0.152
1	1.958

'Partition' = 2_Testing	
0	0.16
1	1.913

Evaluation Metrics

'Partition'	1_Training		2_Testing	
Model	AUC	Gini	AUC	Gini
$R-DLRSN	1.0	1.0	1.0	1.0

Image 12: AUC of CART model

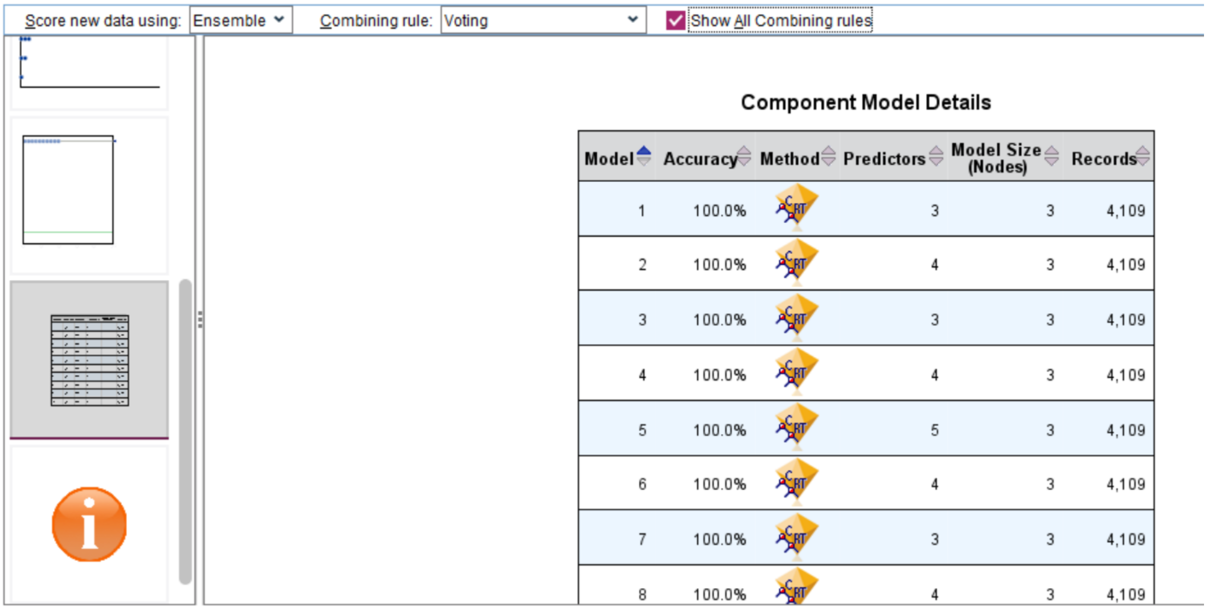


Score new data using: Ensemble

Combining rule: Voting

☐ Show All Combining rules

Image 12: AUC of CART model



Score new data using: Ensemble Combining rule: Voting ☑ Show All Combining rules

Component Model Details

Model	Accuracy	Method	Predictors	Model Size (Nodes)	Records
1	100.0%	C&RT	3	3	4,109
2	100.0%	C&RT	4	3	4,109
3	100.0%	C&RT	3	3	4,109
4	100.0%	C&RT	4	3	4,109
5	100.0%	C&RT	5	3	4,109
6	100.0%	C&RT	4	3	4,109
7	100.0%	C&RT	3	3	4,109
8	100.0%	C&RT	4	3	4,109

Image 13: Model component of CART model

GLM

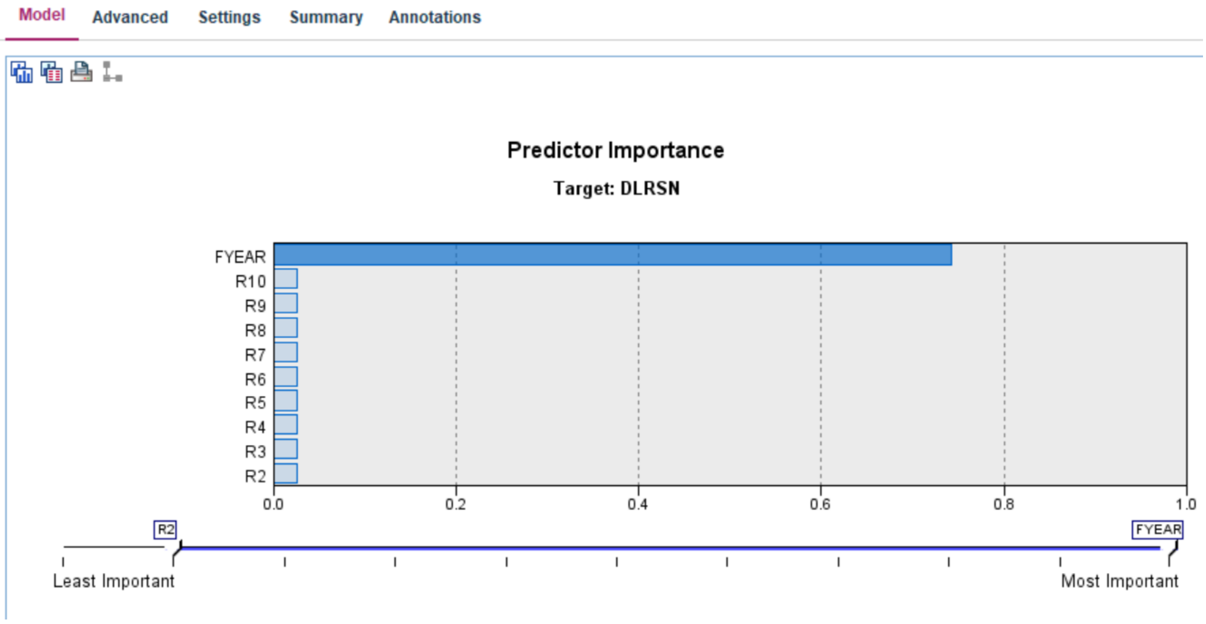
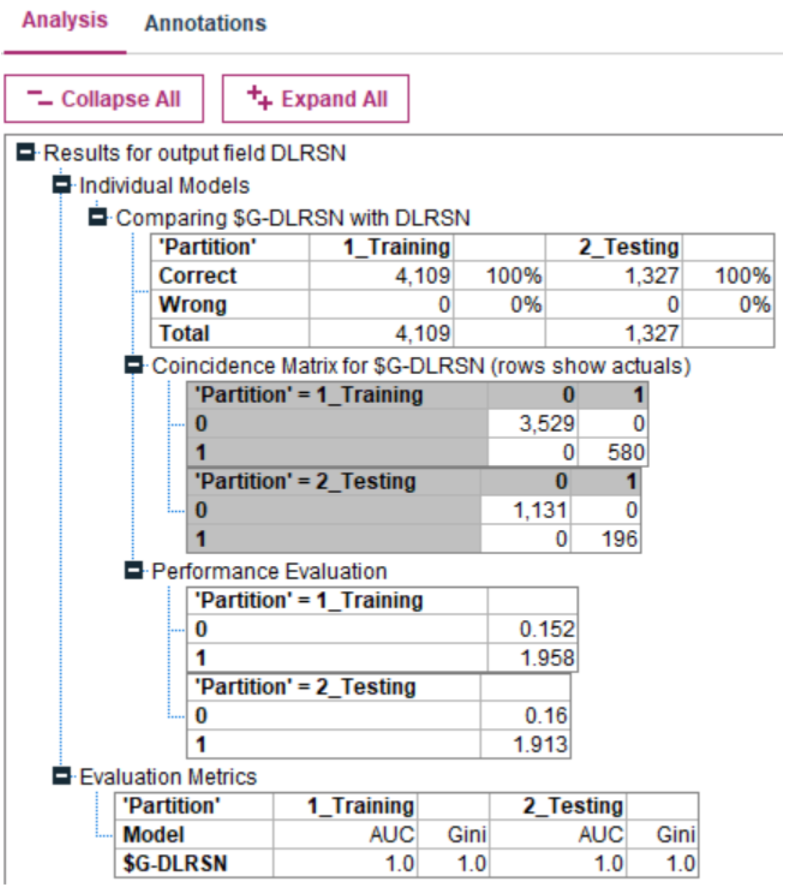


Image 14: Prediction importance on GLM model



Results for output field DLRSN

Individual Models

Comparing $G-DLRSN with DLRSN

'Partition'	1_Training		2_Testing	
Correct	4,109	100%	1,327	100%
Wrong	0	0%	0	0%
Total	4,109		1,327	

Coincidence Matrix for $G-DLRSN (rows show actuals)

'Partition' = 1_Training	0	1
0	3,529	0
1	0	580

'Partition' = 2_Testing	0	1
0	1,131	0
1	0	196

Performance Evaluation

'Partition' = 1_Training	
0	0.152
1	1.958

'Partition' = 2_Testing	
0	0.16
1	1.913

Evaluation Metrics

'Partition'	1_Training		2_Testing	
Model	AUC	Gini	AUC	Gini
$G-DLRSN	1.0	1.0	1.0	1.0

Image 15: AUC of GLM model

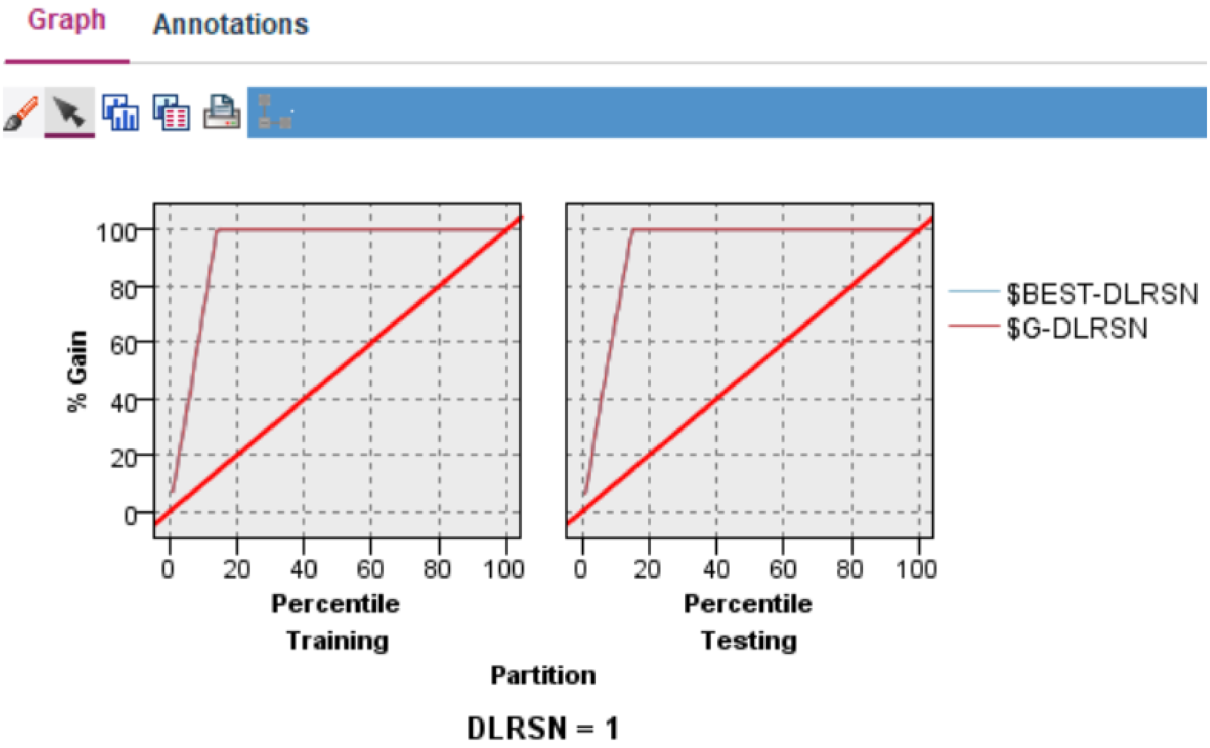


Image 16: ROC of GLM model

Neural Net

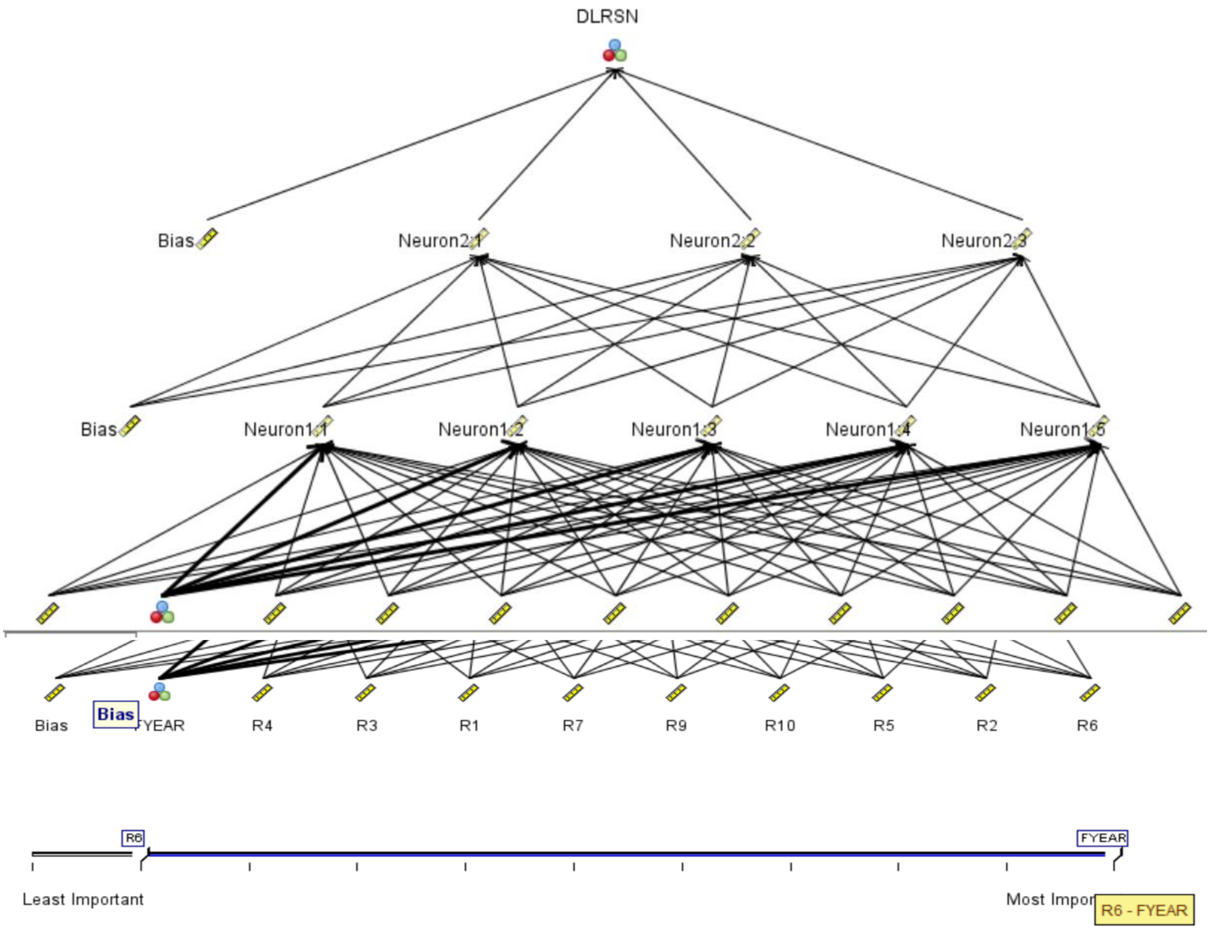
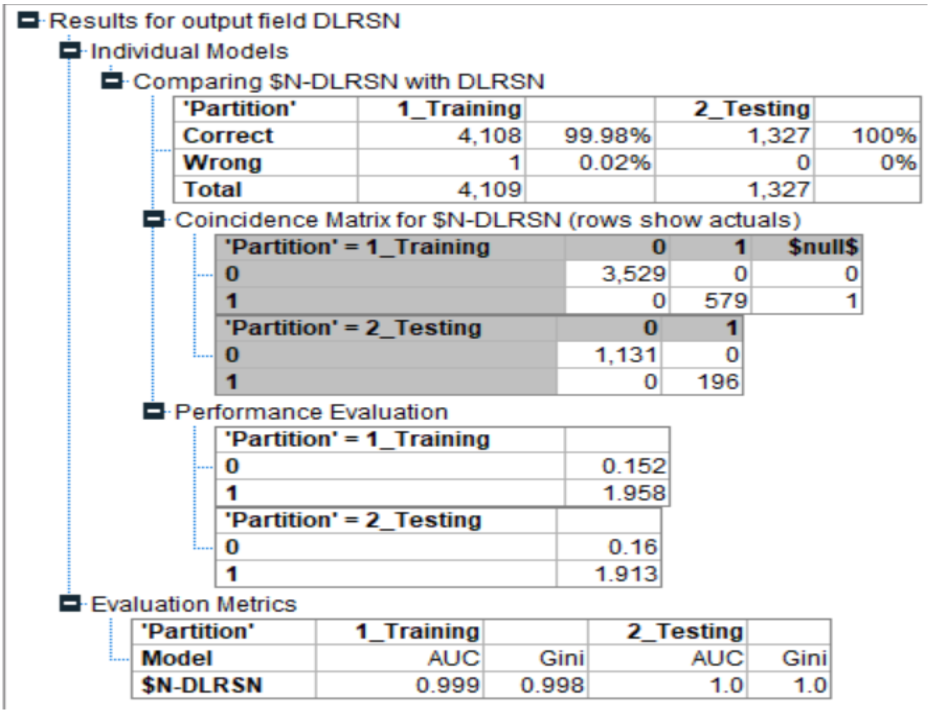


Image 17: Neural Network



Results for output field DLRSN

Individual Models

Comparing $N-DLRSN with DLRSN

'Partition'	1_Training		2_Testing	
Correct	4,108	99.98%	1,327	100%
Wrong	1	0.02%	0	0%
Total	4,109		1,327	

Coincidence Matrix for $N-DLRSN (rows show actuals)

'Partition' = 1_Training	0	1	$null$
0	3,529	0	0
1	0	579	1

'Partition' = 2_Testing	0	1
0	1,131	0
1	0	196

Performance Evaluation

'Partition' = 1_Training	
0	0.152
1	1.958

'Partition' = 2_Testing	
0	0.16
1	1.913

Evaluation Metrics

'Partition'	1_Training		2_Testing	
Model	AUC	Gini	AUC	Gini
$N-DLRSN	0.999	0.998	1.0	1.0

Image 18: AUC of Neural Network

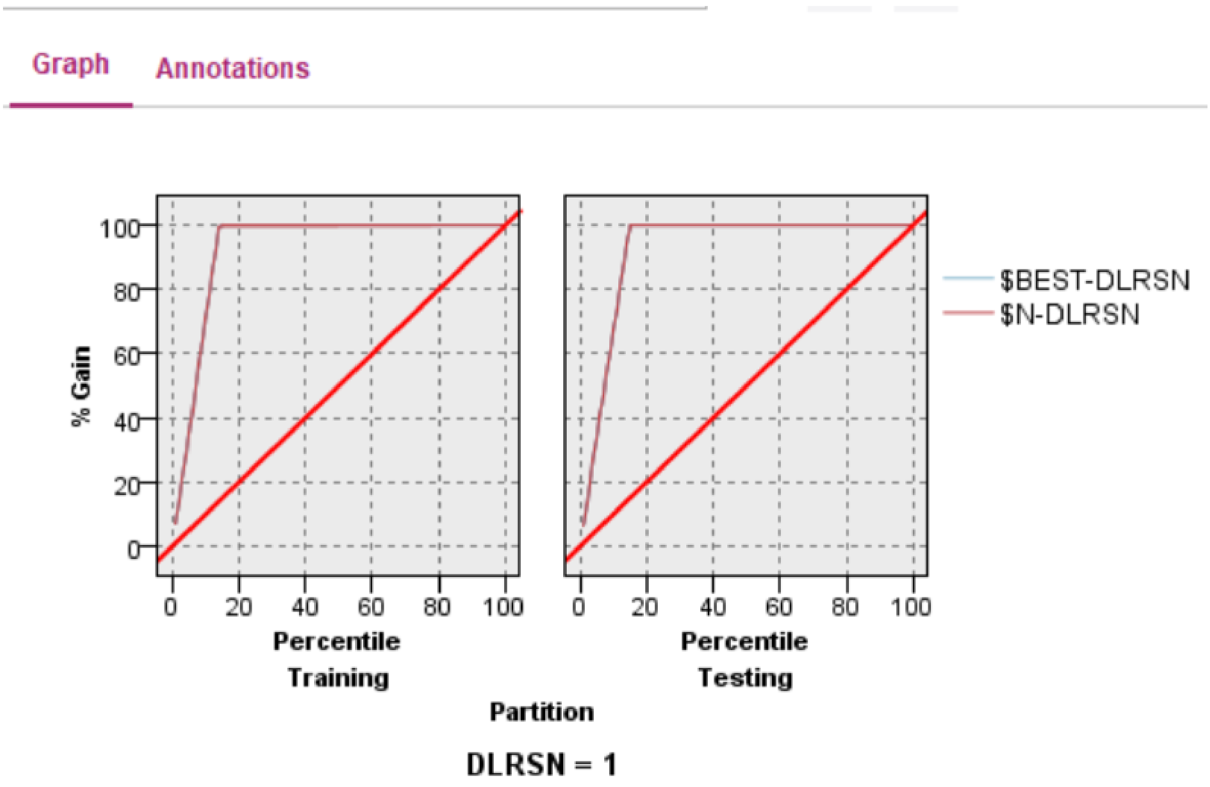


Image 19: ROC of Neural Network