A REPORT

ON

DATA MINING

BY

SIDDHARTH NAGPAL 2014B3A70743P
APURVA MITTAL 2014B4A70658P

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

November, 2017

A REPORT

ON

Analysis of data of All Night Canteen data using Data Mining

By

Siddharth Nagpal (2014B3A70743P; Eco+CS)
Apurva Mittal (2014B4A70658P; Maths+CS)

Prepared in partial fulfillment of the course

CS F415

Submitted To:

Dr. Poonam Goyal

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE,

PILANI

(November, 2017)

ACKNOWLEDGEMENTS

We would like to express our deep and sincere gratitude to our course instructor, Poonam Goyal who gave us the opportunity to work on this project on ‘Data Analysis on real time data of All night canteen’ which strengthened our knowledge and understanding of this topic and the entire course in general. Further, it allowed us to do a lot of research as a result of which we learnt many new things.

We are also duty to record our thankfulness to our seniors and some batch mates who helped us in detailed understanding of some subtopics. They showed keen interest and gave valuable guidance.

Also, we are grateful to the staff of Information Processing Centre (IPC) BITS Pilani, for providing us an opportunity to work with IBM SPSS Modeler without which this report wouldn’t have been possible.

Lastly we would thank our parents and friends for their kind co-operation and encouragement that helped us in completion of this project.

With warm regards,

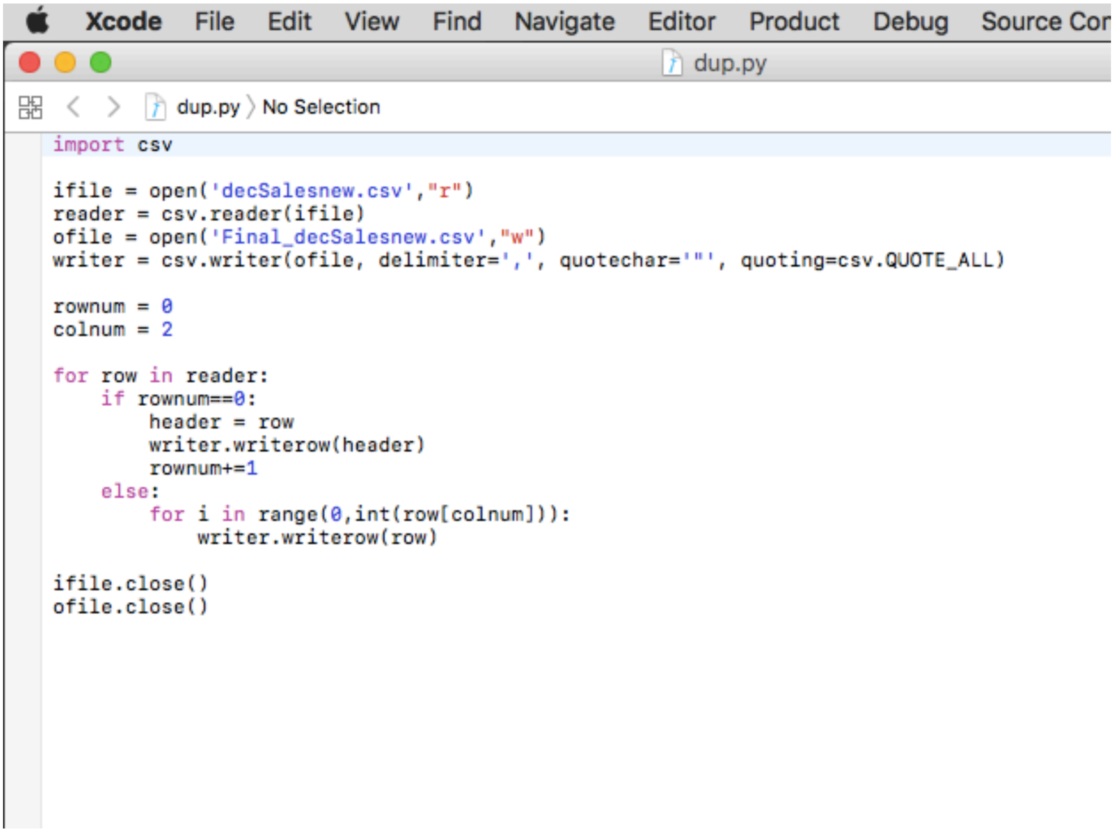
Siddharth Nagpal and Apurva Mittal

TABLE OF CONTENTS

1. PROBLEM 1

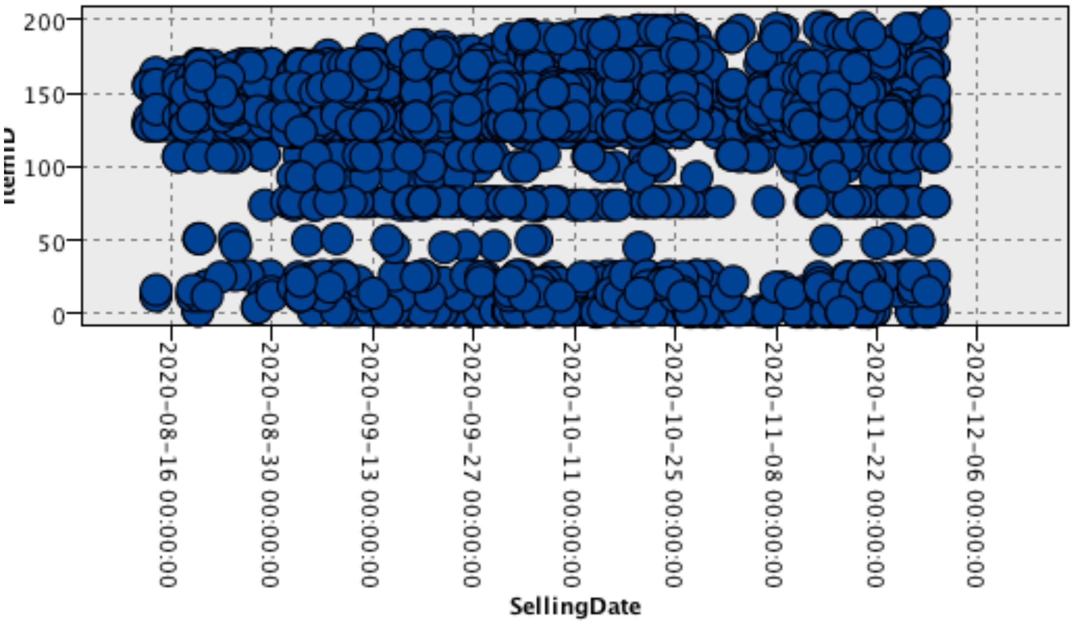
DATA PREPROCESSING

After going through question 1 we thoroughly read the input of August, September, October and November sales that were given to us as the training data. While analyzing the data and the problem of maximizing revenue and minimizing penalty we realized that it will require an unsupervised technique on 3 inputs 'item-id', 'time' and 'student-id'. Also, quantity will be an input that will affect number of points in rule determination. Keeping that in mind we ran a python code on our input, duplicating the items according to their quantity. The screenshot of the same is attached in Figure 1.

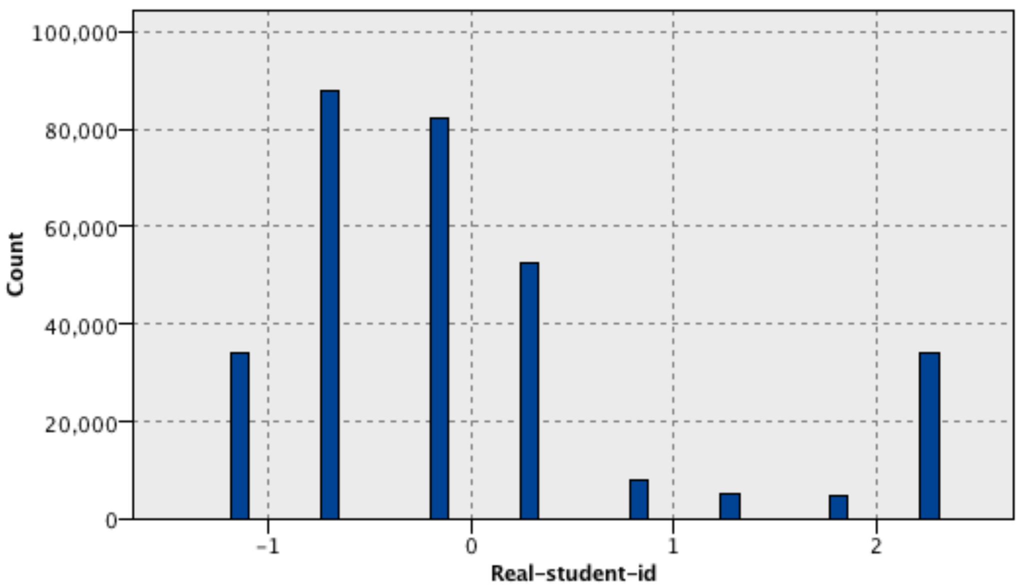


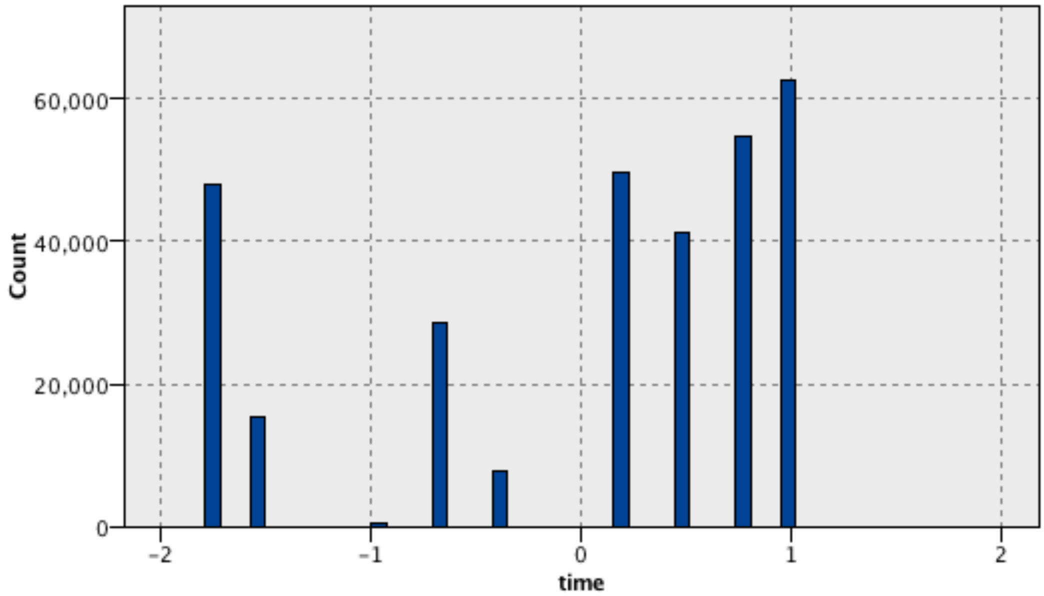

```
import csv

ifile = open('decSalesnew.csv',"r")
reader = csv.reader(ifile)
ofile = open('Final_decSalesnew.csv',"w")
writer = csv.writer(ofile, delimiter=',', quotechar='"', quoting=csv.QUOTE_ALL)

rownum = 0
colnum = 2

for row in reader:
    if rownum==0:
        header = row
        writer.writerow(header)
        rownum+=1
    else:
        for i in range(0,int(row[colnum])):
            writer.writerow(row)

ifile.close()
ofile.close()
```

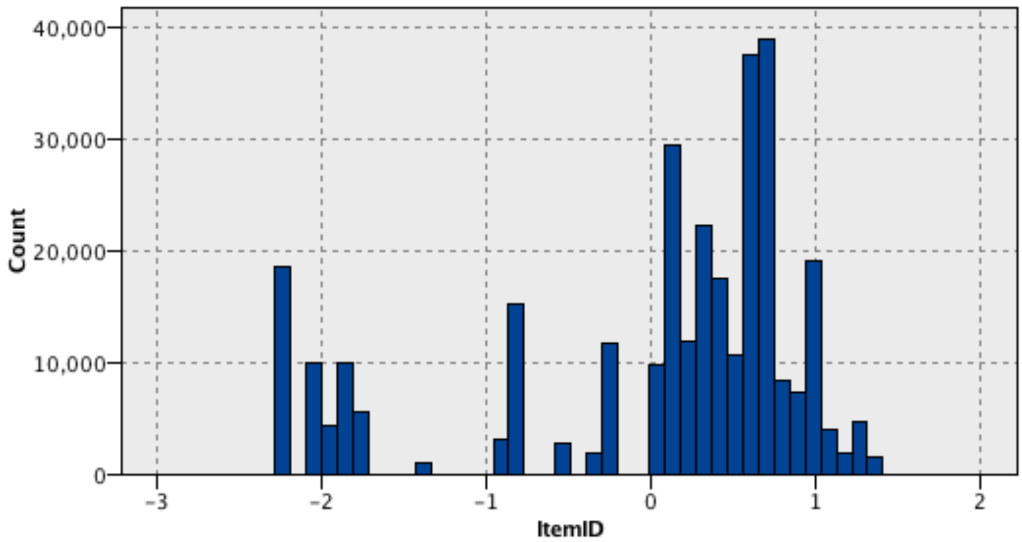
Further analyzing the data through 'Data Audit ' node confirmed that data needs cleaning. There were item ids' with large values that represented fine, reimbursement etc and were not required as input. Also sparse item-id's were removed which did not effect the model and would act as outliers. The student id 'F0' who paid by cash included students from all ids and thus needed to be removed for data pre-processing.



From the 'data audit' graph in figure 2 above we can see data also needs normalization. For normalizing all the fields we needed to mad timestamp and student id to hours and integers respectively. Formula we used for normalization is: (@FIELD-@GLOBAL_MEAN(@FIELD))/(@GLOBAL_SDEV(@FIELD)). Figure 3 below represents final graphs of data.



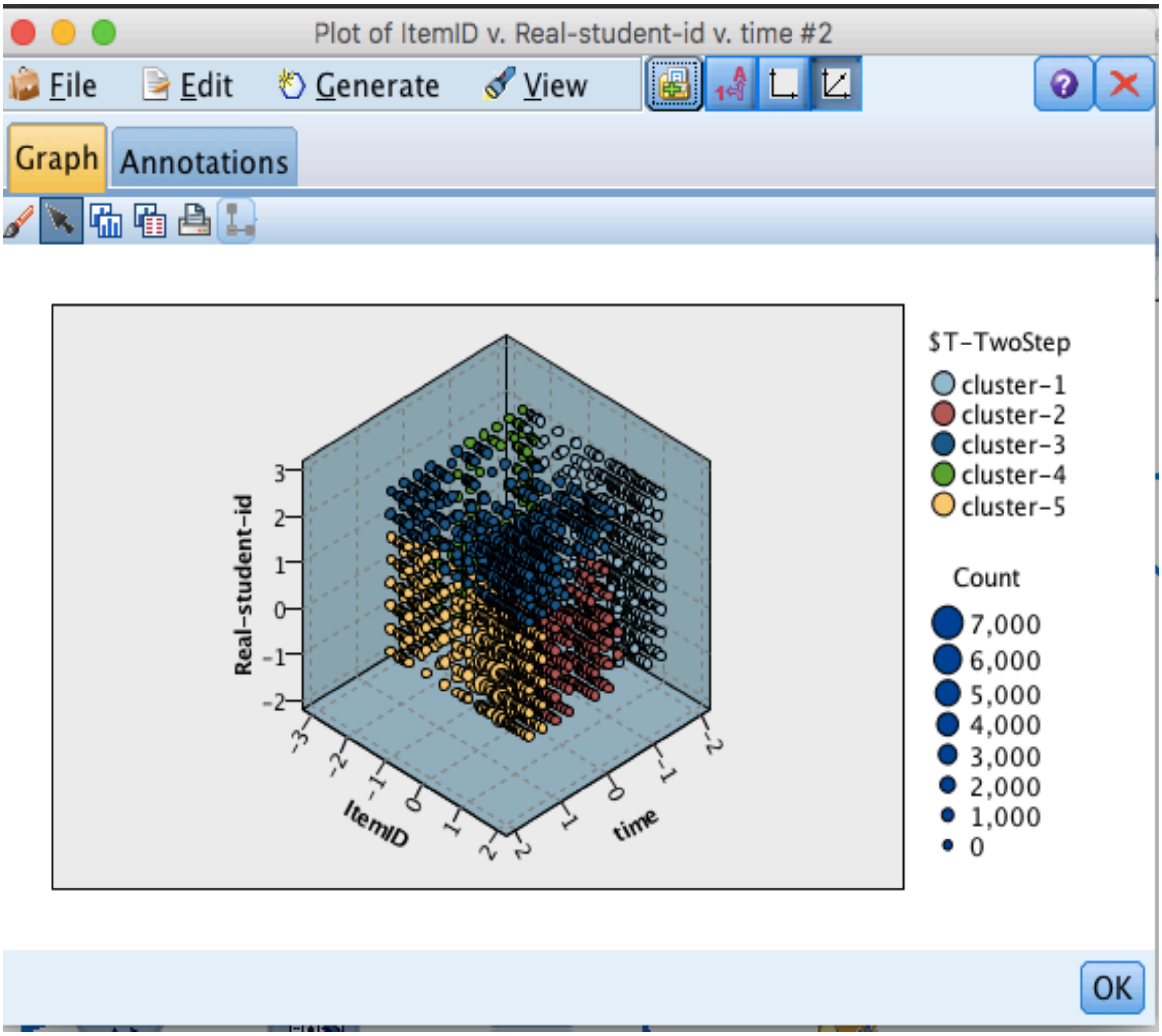




UNSUPERVISED LEARNING

There were 2 options available to us, Association Rule Mining via apriori or Clustering. For this we had read an IJETS paper on 'Comparison of K-Mean Algorithm and Apriori Algorithm analysis. Since our model had multiple variables thus we preferred K-Means for faster computation on small k. Also K-Means promised tighter clusters which were an advantage in problem 1.

The paper gave us the direction on using clustering algorithm so we tried various clustering approaches on out data and found the best clusters in Two-Step. Also it gave chain clusters which were needed since one item-id spread of a big time interval is of importance to us. The Figure 6 below gives us the clusters formed.



OUTPUTS

To analyze the data again we retrieved the data from its normal form using:

((@FIELD) * @GLOBAL_SDEV(@FIELD))+@GLOBAL_MEAN(@FIELD)

and sorted on various attributes to understand it well. Further we aggregated the data on the 3 inputs 'item id', 'student-id' and 'time'. We considered the rules from clusters with large count. Some clusters gave more than 1 rules.

In order to find the penalty 2 separate models were applied.

1st model aggregated time with item-id and student-id as key fields. This was to find the total time in hours for each item for all id's to find the hour segment weight.
2nd model aggregated on item-id and student-id and mapped the student ids to the id weights.
Both the models were merged to find the final total weights.
To maximize the revenue we had to look at the percentage affect of an item id of a student-if at different hours on total revenue. The items with less penalty weights and large percentage affect were selected to increased prices.
Eg: Plain Maggie for id 'F5' in hour 23 only had a small weight of 20 and large reader count 114 having large affect on revenue But the same for id 'H1' in time 23 had maximum weight of 288.

Table (10 fields, 586 records)

File Edit Generate

Table Annotations

	ItemID	ItemNa...	Real-student-id	time	percentage	$T-TwoStep	Record_Count	Weights
267	134	Plain M...	4	20	0.024	cluster-2	122	144
268	134	Plain M...	8	23	0.050	cluster-3	251	288
269	134	Plain M...	3	23	0.165	cluster-5	829	108
270	134	Plain M...	8	22	0.036	cluster-3	181	288
271	134	Plain M...	1	21	0.030	cluster-2	149	36
272	134	Plain M...	2	22	0.106	cluster-5	531	72
273	134	Plain M...	1	20	0.023	cluster-2	114	36
274	134	Plain M...	3	20	0.041	cluster-2	207	108
275	134	Plain M...	4	21	0.023	cluster-2	116	144
276	134	Plain M...	8	21	0.031	cluster-3	154	288
277	134	Plain M...	4	23	0.089	cluster-5	445	144
278	134	Plain M...	4	22	0.046	cluster-5	230	144
279	134	Plain M...	5	23	0.023	cluster-5	117	20
280	134	Plain M...	8	20	0.031	cluster-3	156	288
281	134	Plain M...	3	22	0.091	cluster-5	457	108
282	134	Plain M...	2	23	0.204	cluster-5	1021	72
283	134	Plain M...	1	23	0.120	cluster-5	604	36
284	134	Plain M...	3	21	0.046	cluster-2	232	108
285	135	Fried M...	2	20	0.039	cluster-2	138	50
286	135	Fried M...	8	21	0.042	cluster-3	150	72

OK

Analyzing the table above and keeping the limit of min of 10% and Rs 10 of increase allowed we devised the new priced of various items at specific hours for specific target IDs.
A snapshot of some changed prices is given below.
Finally after applying various mathematical computations the final total increase in revenue and total penalty is calculated on the testing data for Dec Sales.

Table Annotations

	ItemID	ItemName	max-new-price
22	129	Plain Dosa	17.600
23	130	Masala Dosa	23.100
24	132	Onion Uttapam	23.100
25	134	Plain Maggi	16.500
26	135	Fried Maggi	23.100
27	138	Paneer Franky	24.200
28	139	Veg Rice	23.100
29	140	Egg Rice	28.600
30	141	Chicken Rice	44.000
31	142	Chicken Sandwich	27.500
32	146	Butter Chicken	93.500
33	148	Tandury Chicke...	99.000
34	151	Butter Naan	13.200
35	154	Ice Cream Shake	24.200
36	155	PEPSI 600ML	29.700
37	156	DEW MIX	33.000
38	157	SLICE 600ML	35.200
39	159	MYCAN	22.000
40	160	BHELPURI	27.500
41	161	SINGLE SCOOP	13.200

OK

Statistics of [pricelist]

File Edit Generate

Statistics Annotations

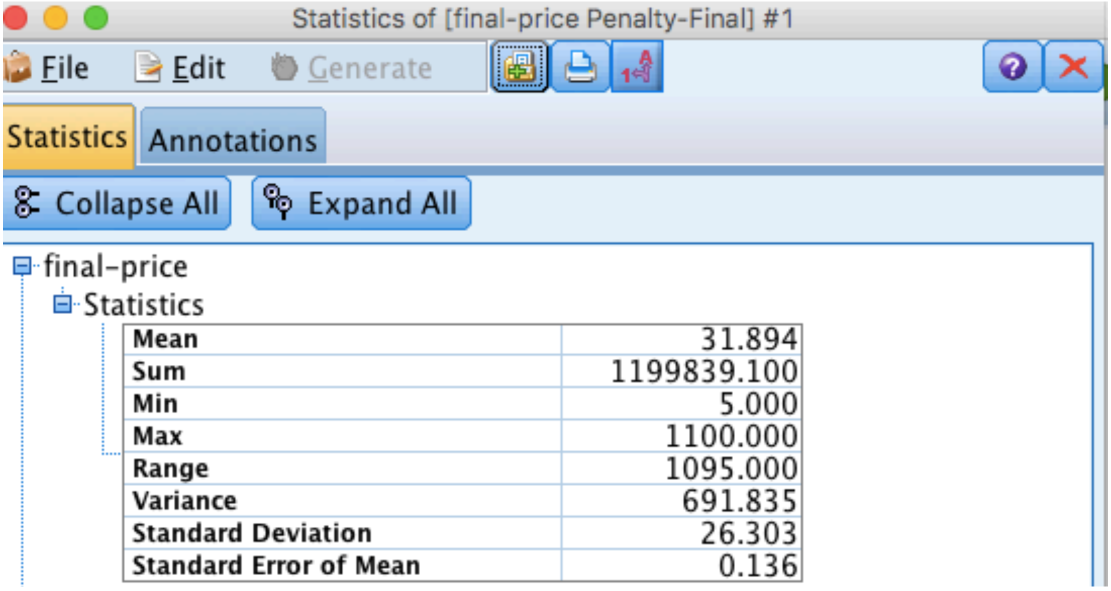
Collapse All Expand All

pricelist

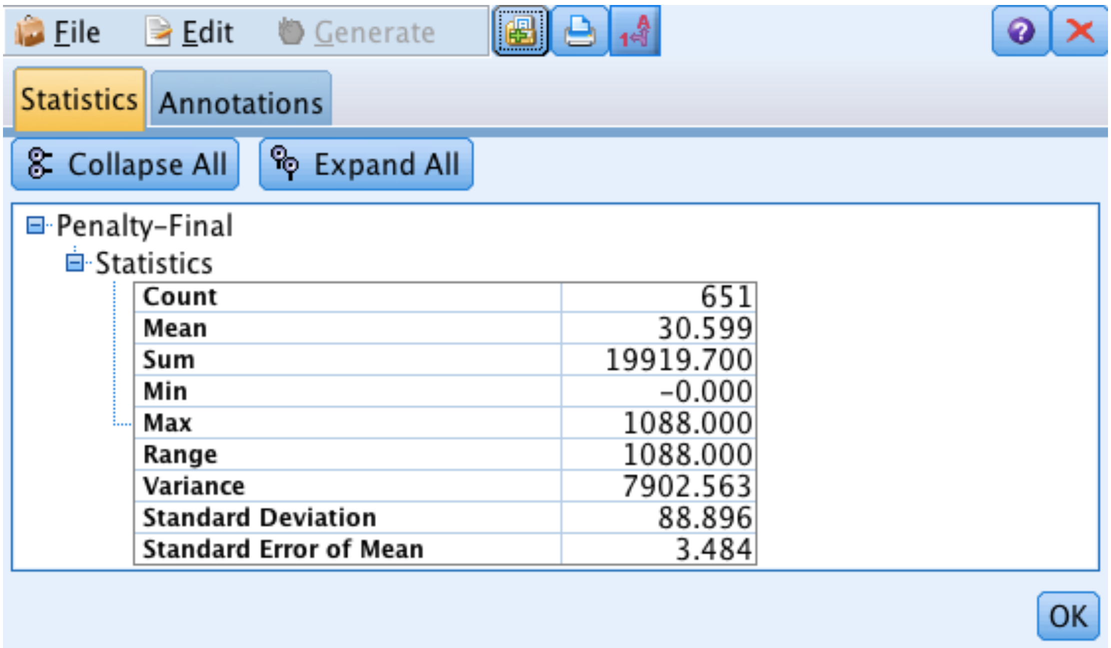
Statistics

Sum	1140943

OK



Mean	31.894
Sum	1199839.100
Min	5.000
Max	1100.000
Range	1095.000
Variance	691.835
Standard Deviation	26.303
Standard Error of Mean	0.136



Count	651
Mean	30.599
Sum	19919.700
Min	-0.000
Max	1088.000
Range	1088.000
Variance	7902.563
Standard Deviation	88.896
Standard Error of Mean	3.484

From the statistic node output we can see that the final price for Dec sales is increased to Rs 1199839.1 from Rs 1140943. Thus making a total increase of **5.1%** and penalty 19,919.7.

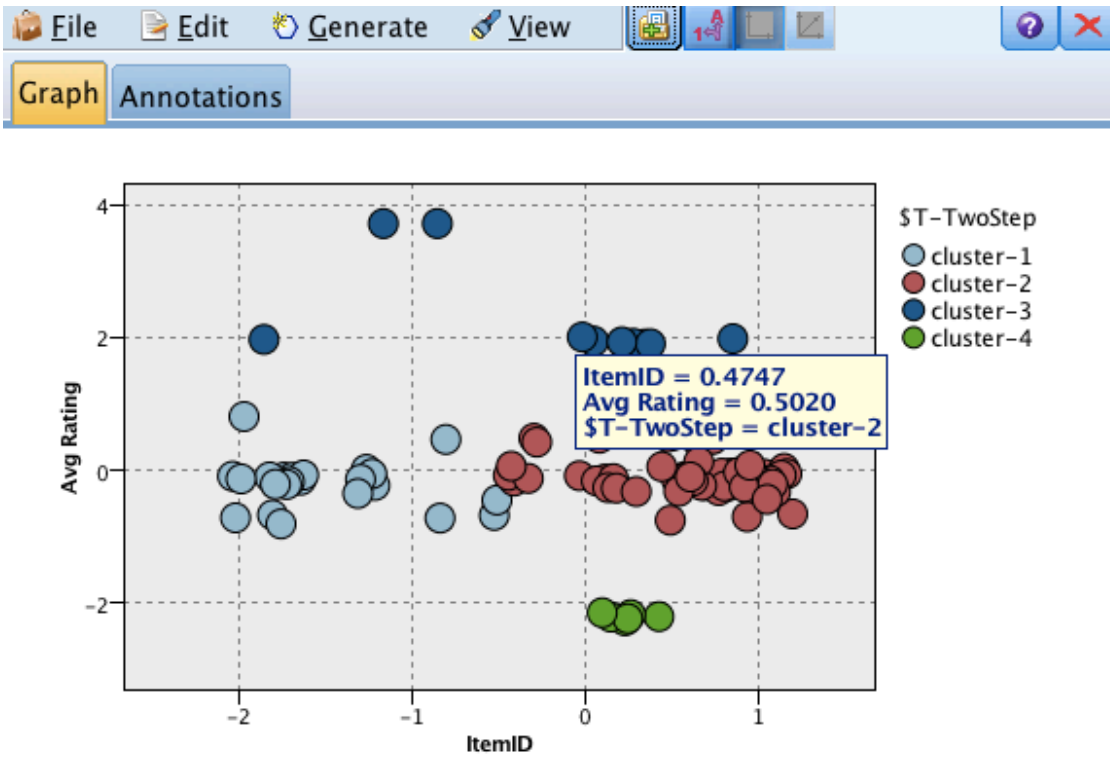
2. PROBLEM 2

In the second problem, we have been asked to form combo meals between items having higher and lower average rating, and also between items having lower average rating. It is clear from the problem that **Apriori** has to be used. We also use clustering to identify items having high average rating and low average rating.
The following steps have been followed:

1. Read the input of August, September, October and November sales that were given to us as the training data and then append the data into a single table.
2. Remove the transactions having ItemID as 900 and 901 because it shows Reimbursement and the fees paid.
3. We aggregate the records based on ItemID in order to find the sum of the final_rating.
4. Average rating is found using the formula given below in the Derive Node:

final_rating_Sum / Record_Count

5. We then normalize the ItemId and average rating values and use TwoStep Clustering on it. We get four clusters as shown below:



Cluster 3- Items having high average rating

Cluster 4- Items having low average rating

Model Summary

Algorithm	TwoStep
Inputs	2
Clusters	4

Cluster Quality

We get a good value of the Silhouette coefficient for our clusters.

6. We restructure the data using the Restructure node so that the unique values of the ItemID becomes the new attributes.

7. Then using the combination of **Date attribute formed and the Bill No. as the Primary Key** we aggregate the values of different Item nos. so that the Items purchased together can be known and Apriori can be applied on it.

8. Then using the items having high average rating and low average rating (found using clustering in step 5) as the possible consequents and antecedents we get the following 22 rules:

Consequent	Antecedent	Support %	Confidence %
ItemID_151_Sum	ItemID_146_Sum	3.506	83.27
ItemID_146_Sum	ItemID_151_Sum	7.796	37.447
ItemID_151_Sum	ItemID_148_Sum	2.537	19.161
ItemID_128_Sum	ItemID_12_Sum	2.764	11.388
ItemID_128_Sum	ItemID_138_Sum	3.002	10.925
ItemID_128_Sum	ItemID_134_Sum	7.372	9.867
ItemID_128_Sum	ItemID_139_Sum	1.98	9.653
ItemID_151_Sum	ItemID_139_Sum	1.98	7.381
ItemID_128_Sum	ItemID_141_Sum	2.115	7.002
ItemID_146_Sum	ItemID_148_Sum	2.537	6.648
ItemID_148_Sum	ItemID_151_Sum	7.796	6.234
ItemID_142_Sum	ItemID_141_Sum	2.115	6.127
ItemID_134_Sum	ItemID_128_Sum	13.655	5.327
ItemID_151_Sum	ItemID_141_Sum	2.115	5.189
ItemID_128_Sum	ItemID_148_Sum	2.537	5.057
ItemID_134_Sum	ItemID_12_Sum	2.764	5.048
ItemID_148_Sum	ItemID_146_Sum ItemID_151_Sum	2.919	4.983
ItemID_134_Sum	ItemID_138_Sum	3.002	4.912
ItemID_134_Sum	ItemID_139_Sum	1.98	4.876
ItemID_146_Sum	ItemID_141_Sum	2.115	4.814
ItemID_148_Sum	ItemID_146_Sum	3.506	4.81

9. Out of these 22 rules, we eliminate the redundant rules and the rules which are formed between items having high rating. Then we select the best 10 rules.

The rules selected are:

1) Item128 + Item 141
2) Item 151 + Item 148
3) Item 128 + Item 134
4) Item 139 + Item 141
5) Item 139 + Item 128
6) Item 151 + Item 141
7) Item 139 + Item 134
8) Item 142 + Item 141
9) Item 146 + Item 141
10) Item 138 + Item 134

10. Then we calculate the decrease in price of the combo and multiply it with the quantity, in those bills where these items occur together.

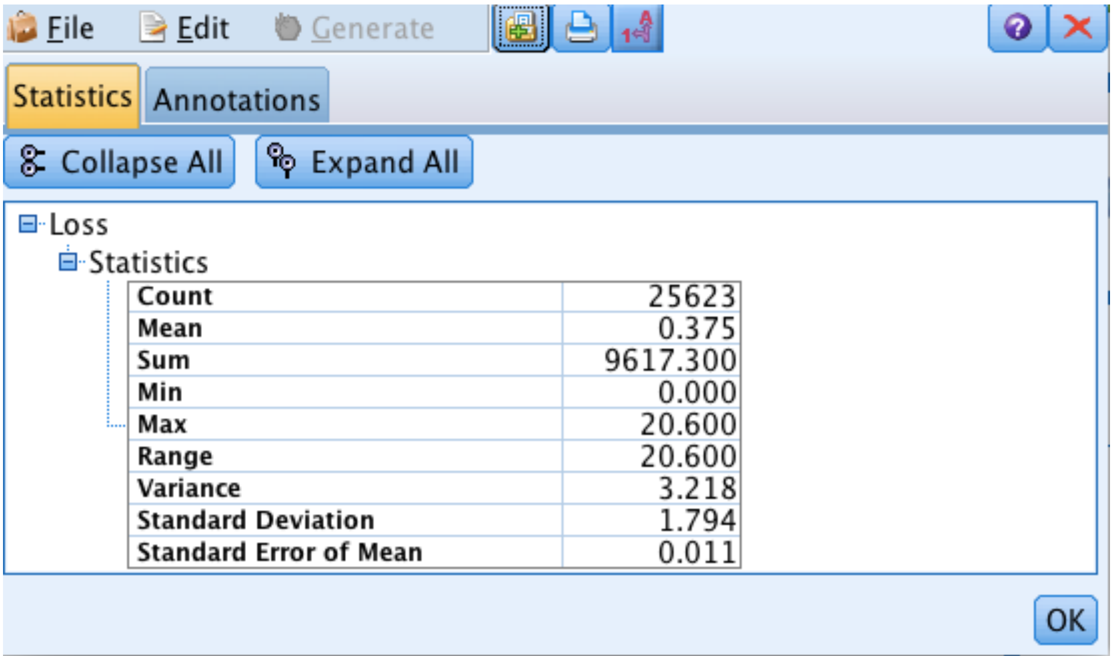
Adding up the decrease in each bill having the combo, gives us the decrease in total revenue.

We used the following condition

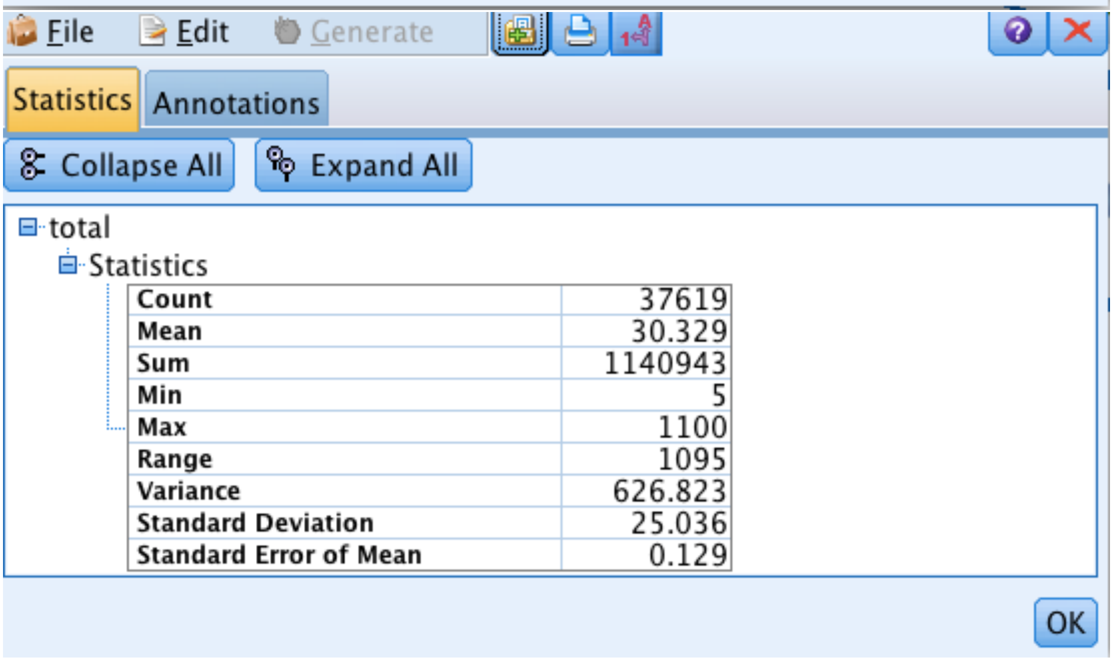
if (ItemID_128_Sum > 0 and ItemID_141_Sum > 0) then (0.1*(ItemID_128_Sum + ItemID_141_Sum))
elseif (ItemID_146_Sum > 0 and ItemID_151_Sum > 0) then (0.1*(ItemID_146_Sum + ItemID_151_Sum))
elseif (ItemID_128_Sum > 0 and ItemID_134_Sum > 0) then (0.1*(ItemID_128_Sum + ItemID_134_Sum))
elseif (ItemID_139_Sum > 0 and ItemID_151_Sum > 0) then (0.1*(ItemID_139_Sum + ItemID_151_Sum))
elseif (ItemID_139_Sum > 0 and ItemID_128_Sum > 0) then (0.1*(ItemID_139_Sum + ItemID_128_Sum))
elseif (ItemID_141_Sum > 0 and ItemID_151_Sum > 0) then (0.1*(ItemID_141_Sum + ItemID_151_Sum))
elseif (ItemID_139_Sum > 0 and ItemID_134_Sum > 0) then (0.1*(ItemID_139_Sum + ItemID_134_Sum))
elseif (ItemID_142_Sum > 0 and ItemID_141_Sum > 0) then (0.1*(ItemID_142_Sum + ItemID_141_Sum))
elseif (ItemID_146_Sum > 0 and ItemID_141_Sum > 0) then (0.1*(ItemID_146_Sum + ItemID_141_Sum))

elseif (ItemID_138_Sum > 0 and ItemID_134_Sum > 0) then (0.1*(ItemID_134_Sum + ItemID_138_Sum))
else 0 endif

It gave us the following output



Count	25623
Mean	0.375
Sum	9617.300
Min	0.000
Max	20.600
Range	20.600
Variance	3.218
Standard Deviation	1.794
Standard Error of Mean	0.011



Count	37619
Mean	30.329
Sum	1140943
Min	5
Max	1100
Range	1095
Variance	626.823
Standard Deviation	25.036
Standard Error of Mean	0.129

Thus total loss is 9617.3/1140943 which is **0.84%.**

3. PROBLEM 3

Q1. Classification in preference of Vegetarian and Non- Vegetarian food item depending on the class of students and the hour they come to ANC. This will help to build preference and discounts and diff hours.

Solution:

Similar to Ques1, data preprocessing of this problem includes appending the data of 4 months, classifying the student-id into 8 classes, converting the timestamp in hours.

We then classify the food items as veg and nonveg. For the food items for which data for non-veg is not available is taken as NULL, eg Veg burger. Also items available at pit-shop like biscuits, chips, coke etc are removed.

Eventually we are left with the following food items mentioned below in the table.

Table (8 fields, 36 records)

File Edit Generate

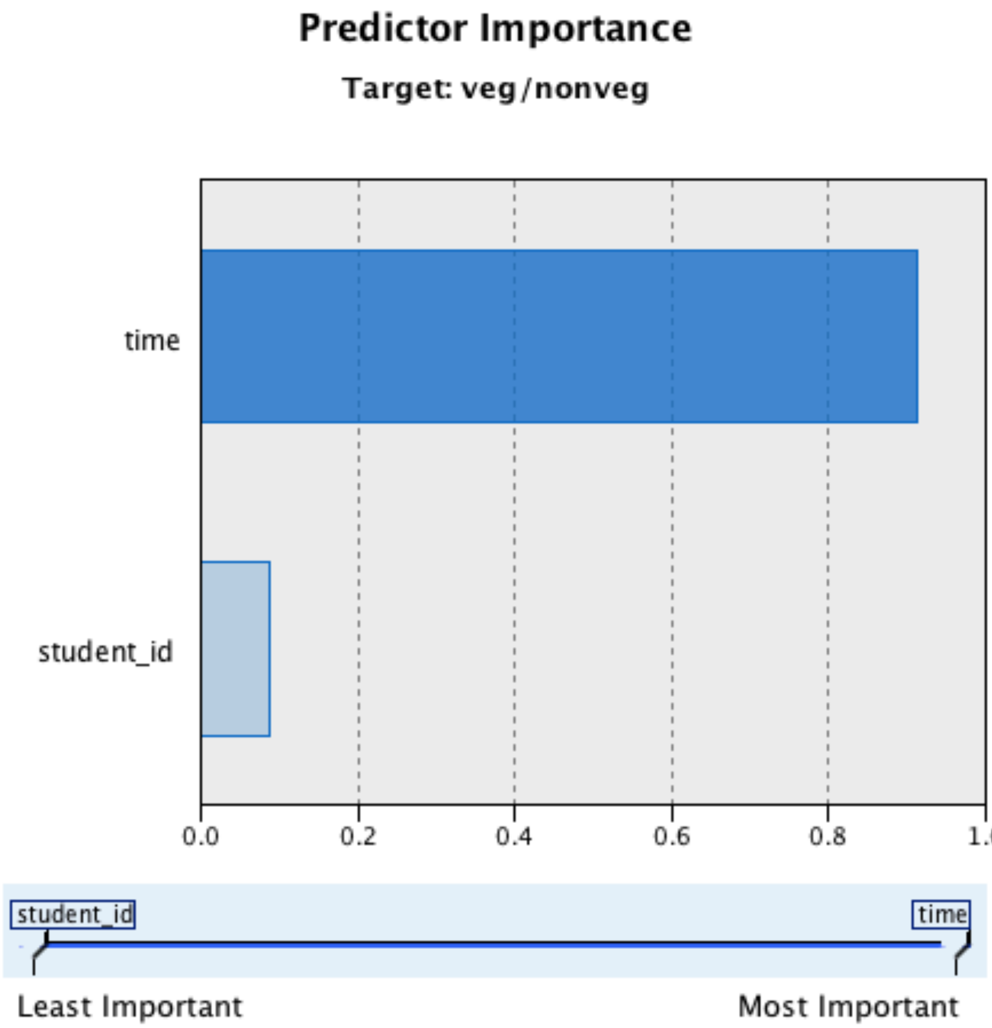
Table Annotations

	ItemID	ItemName	SellingPriceAug	SellingPriceSep	SellingPriceOct	SellingPriceNov	SellingPriceDec	veg/nonveg
1	1	Samosa	5	5	7	7	7	0
2	2	Veg Petty	8	8	8	8	8	0
3	3	Paneer Petty	12	12	12	12	12	0
4	4	Veg Burger	21	21	21	21	21	null
5	12	Pasta	30	30	30	30	30	0
6	124	Chese Burger	25	25	25	25	25	null
7	125	Veg Pizza	35	35	35	35	35	0
8	126	Chese Pizza	40	40	40	40	40	null
9	127	Chicken Pizza	55	55	55	55	55	1
10	128	Chese Toast	25	25	25	25	25	0
11	129	Plain Dosa	16	16	16	16	16	0
12	130	Masala Dosa	21	21	21	21	21	0
13	131	Onion Masal...	21	21	21	21	21	0
14	132	Onion Uttap...	21	21	21	21	21	0
15	133	Mix Uttapam	21	21	21	21	21	0
16	134	Plain Maggi	15	15	15	15	15	0
17	135	Fried Maggi	21	21	21	21	21	0
18	136	Paneer Maggi	25	25	25	25	25	0
19	137	Veg Chomin	30	30	30	30	30	0
20	138	Paneer Franky	22	22	22	22	22	0
21	139	Veg Rice	21	21	21	21	21	0
22	140	Egg Rice	26	26	26	26	26	1
23	141	Chicken Rice	40	40	40	40	40	1
24	142	Chicken San...	25	25	25	25	25	1
25	143	OMLETE	16	16	16	16	16	1
26	144	Dum Aloo	20	20	20	20	20	0
27	145	Mutter Paneer	40	40	40	40	40	0
28	146	Butter Chicken	85	85	85	85	85	1
29	147	Chicken Curry	75	75	75	75	75	1
30	148	Tandury Chi...	90	90	90	90	90	1
31	164	KADHAI PAN...	81	81	81	81	81	0
32	166	CHILLI PANEER	85	85	85	85	85	0
33	177	CHILLY CHIC...	95	95	95	95	95	1
34	178	CHI.PASTA	40	40	40	40	40	1
35	180	TANDOORI F...	170	170	170	170	170	1
36	902	PANEER SAN...	25	25	25	25	25	0

Now we remove the items with NULL value and merge the result with our training data with inner join.

Specifying the type with the help of type node we select time and mapped student id as input and veg and non-veg as output.

We now apply the Bayes Net Classifier to classify the training data.
It gave us the following results.
The Predictor Importance tells us that we can majorly classify on the hour students come in.



Following we found the conditional probabilities for all student ids

Conditional Probabilities of student_id

Parents	Probability							
veg/nonveg	1	2	3	4	5	6	7	8
1	0.09	0.23	0.27	0.17	0.03	0.03	0.03	0.15
0	0.13	0.29	0.25	0.16	0.02	0.02	0.01	0.12

Similarly we can study the conditional probabilities of each hour as well.

Conditional Probabilities of time

Parents		Probability								
veg/nonveg	student_id	13	14	15	16	17	18	19	20	21
1	1	0.18	0.03	0.00	0.00	0.00	0.00	0.00	0.13	0.14
1	2	0.15	0.03	0.00	0.00	0.00	0.00	0.00	0.22	0.14
1	3	0.15	0.04	0.00	0.00	0.00	0.00	0.00	0.23	0.16
1	4	0.17	0.04	0.00	0.00	0.00	0.00	0.00	0.26	0.15
1	5	0.17	0.05	0.00	0.00	0.00	0.00	0.00	0.25	0.16
1	6	0.20	0.04	0.00	0.00	0.00	0.00	0.00	0.26	0.15
1	7	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.54	0.22
1	8	0.06	0.03	0.00	0.00	0.00	0.00	0.00	0.44	0.23
0	1	0.24	0.08	0.00	0.00	0.10	0.02	0.00	0.07	0.08
0	2	0.15	0.04	0.00	0.00	0.23	0.02	0.00	0.08	0.08
0	3	0.19	0.05	0.00	0.00	0.12	0.02	0.00	0.10	0.10

The output of the training model can be analyzed with the analysis node as following.

Results for output field veg/nonveg

Individual Models

Comparing $B-veg/nonveg with veg/nonveg

Correct	97,106	77.95%
Wrong	27,469	22.05%
Total	124,575	

Coincidence Matrix for $B-veg/nonveg (rows show actuals)

	0	1
0	96,504	523
1	26,946	602

Performance Evaluation

0	0.004
1	0.884

Evaluation Metrics

Model	AUC	Gini
$B-veg/nonveg	0.675	0.349

Now the model is ready and we can test it on the testing data given to us as Dec sales. The following was the output observed.

- Results for output field veg/nonveg
  - Individual Models
    - Comparing $B-veg/nonveg with veg/nonveg

Correct	1,675,536	73.65%
Wrong	599,586	26.35%
Total	2,275,122	

    - Performance Evaluation

0	0.004
1	0.813

  - Evaluation Metrics

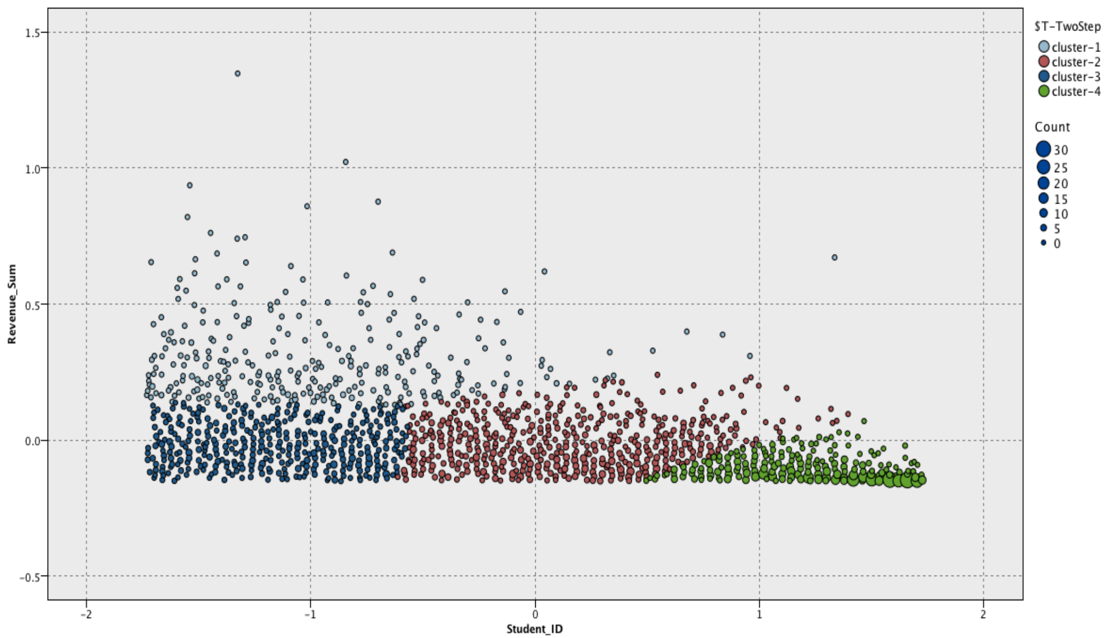
Model	AUC	Gini
$B-veg/nonveg	0.668	0.335

Q2. Market Segmentation. We try to group customers into related sets, in our case based upon revenue generated in the given period. It is an important tool for applied Marketing.

Solution:

In the first step, we read the data for each month from August to November into the Variable node. Then using the Append node, this data is appended into a single table.

The revenue generated by each transaction is found using Price * Quantity. Then we aggregate the data based upon StudentID, such that, we get the revenue generated by each student. We need to remove the revenue generated by the cash transactions. Then after normalizing the attributes StudentID and Revenue we apply Two Step Clustering to it. The result is as shown below:

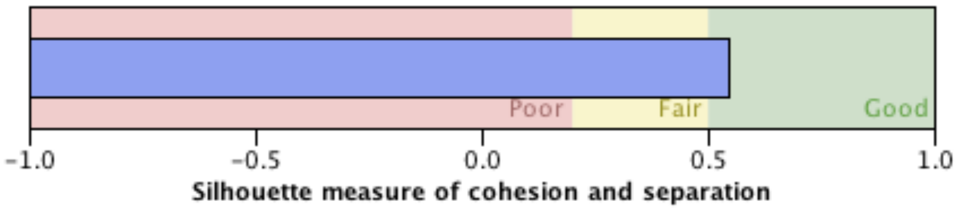


We get 4 clusters. The high revenue customers have been put in one cluster. They can be offered additional discounts and offers which would help increase the items sold and increase revenue.

Model Summary

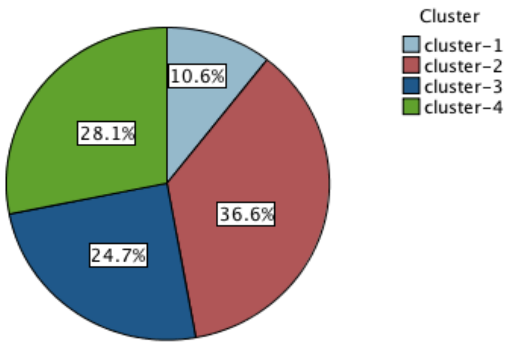
Algorithm	TwoStep
Inputs	2
Clusters	4

Cluster Quality



From the summary, we can see that we get a good measure of the Silhouette coefficient for our clusters.

Cluster Sizes



Size of Smallest Cluster	355 (10.6%)
Size of Largest Cluster	1229 (36.6%)
Ratio of Sizes: Largest Cluster to Smallest Cluster	3.46

From the above result figure, we can see that the points are also well distributed among the clusters. The high revenue customers correspond to cluster number 1.