Module – 01 – Stan-Phase-01.py

0 Id 23223 non-null int64

1 Age 23223 non-null object

2 Geo 23223 non-null object

3 Gender 23223 non-null object

4 Language\_Spoken 23223 non-null object

5 Language\_Used 23223 non-null object

6 Education 23223 non-null object

7 Employed 23223 non-null object

8 Occupation 23223 non-null object

9 HH\_Income 23223 non-null object

10 Ethnic\_Background 23223 non-null object

11 Labels 23223 non-null object

We have 23.223 customers from which 10 parameters were identified and a label indicating what this customer bought (Type of Popeye´s dish consumed by the person). We have 12 types of dishes; the dishes are associated to a number between 0 and 11:

* 0 - Burgers\_Burger Past 30 1-2
* 1 - Burgers\_Burger past 30 3-4
* 2 - Burgers\_Burger Past 30 5+
* 3 - Chicken\_Chicken Past 30 1-2
* 4 - Chicken\_Chicken past 30 3-4
* 5 - Chicken\_Chicken Past 30 5+
* 6 - Popeye (Less) 1-2
* 7 - Popeye (Less) 3-4
* 8 - Popeye (Less) 5+
* 9 - Popeyes 1-2
* 10 - Popeyes 3-4
* 11 - Popeyes 5+

Each parameter encompasses a given number of classes, and each class is associated to a number.

Parameter 01 - Age 15 classes:

70+ 2654 0

40-44 2187 1

30-34 2166 2

55-59 2044 3

45-49 2033 4

50-54 2033 5

65-69 1984 6

60-64 1925 7

25-29 1595 8

35-39 1536 9

22-24 879 10

15-17 784 11

20-21 552 12

19 Years 393 13

18 Years 315 14

14 Years 143 15

Parameter 02 - Geo 10 classes:

Ontario 9115 0

Quebec 5198 1

Alberta 2895 2

British Columbia 2617 3

Manitoba 1005 4

Saskatchewan 797 5

Nova Scotia 672 6

New Brunswick 485 7

Newfoundland and Labrador 361 8

Prince Edward Island 78 9

Parameter 03 - Sex 2 classes

Male 11656 0

Female 11567 1

Parameter 04 - Language spoken 19 classes

English 13468 0

French 5035 1

Other 1294 2

Spanish 567 3

German 463 4

Indian-Hindi 377 5

Italian 279 6

Chinese - Mandarin 243 7

Indian-Punjabi 235 8

Arabic 216 9

Indian-Other 198 10

Chinese-Cantonese 177 11

Tagalog 156 12

Portuguese 138 13

Polish 114 14

Greek 85 15

Ukrainian 84 16

Chinese-Other 50 17

Vietnamese 44 18

Parameter 05 - Language used at home 19 classes

English 16435 0

French 4691 1

Other 730 2

Chinese-Cantonese 182 3

Arabic 179 4

Chinese - Mandarin 169 5

Spanish 135 6

Indian-Punjabi 118 7

Tagalog 109 8

Indian-Hindi 106 9

Indian-Other 93 10

Polish 63 11

Portuguese 44 12

Vietnamese 37 13

German 35 14

Ukrainian 33 15

Greek 26 16

Italian 25 17

Chinese-Other 13 18

Parameter 06 - Education 12 classes

High School Graduation Certificate 6549 0

Elementary School Graduation Cert 4604 1

Other Certificate/Diploma Not Univ 3325 2

Bachelors Degrees 3174 3

Trade Certificate/Diploma 2239 4

Masters Degrees 932 5

Univ Cert/Dipl Below Bachelor Level 913 6

None 663 7

Univ Cert/Dipl Above Bachelor Level 545 8

Earned Doctorate 178 9

Degree In Medicine/Dentistry/ Veterinary Medic 92 10

Not Stated 9 11

Parameter 07 - Employment 3 classes

Full-Time 11025 0

Not Employed 9704 1

Part-Time 2494 2

Parameter 08 - Type of job 9 classes

Other 9322 0

Other Managers 3530 1

Technical/Sales/Teaching/Other White Collar 2359 2

Clerical/Secretarial 1973 3

Unskilled 1762 4

Professionals 1659 5

Skilled 1315 6

Senior Managers/Owners 1105 7

Primary 198 8

Parameter 09 - Income range 11 classes

$75000-$99999 3403 0

$100000-$124999 3077 1

Under $25000 2910 2

$60000-$74999 2569 3

$50000-$59999 2011 4

$25000-$34999 1877 5

$40000-$49999 1808 6

$150000-$199999 1638 7

$125000-$149999 1568 8

$200000 Or Over 1368 9

$35000-$39999 994 10

Parameter 10 Ethnic Background 14 classes

White 16653 0

South Asian (East Indian Pakistani) 1066 1

Not Stated 969 2

Black 911 3

Other 812 4

Aboriginal 748 5

Chinese 574 6

Arab 377 7

Filipino 324 8

Latin/Central and South American 288 9

East/South Asian (Cambodian Indonesian) 223 10

West Asian (Afghan Iranian etc) 147 11

Japanese 71 12

Korean 60 13

All associated parameters were normalized and brought back to an interval between 0-1.

Module 02 – Stan-phase-02.py

Once we have normalized the parameters we group the customers into 120 nodes (Using k-means).

Note that the choice of 120 nodes is arbitrary and the objective is to see if grouping these customers around these nodes we manage to see enough consistence in terms of predominance of a specific label in specific nodes.If we do not see this predominance we may increase the number of nodes. If we do we may reduce it until we find a point where it stops happening.

Here it is important to mention that we do not use the 23.223 customers as our training data directly. The reason for that is the fact that the labels are unbalanced. To be able to obtain a good result we have to balance it:

0 9450

3 7035

1 2630

4 1450

2 840

5 579

6 571

9 303

7 177

11 65

10 62

8 61

We do that creating a code that oversample some of the labels generating a file with 1.000 samples of each one of the 12 labels (Total of 12.000 training samples).

Now that we have our 120 nodes we calculated the percentage of label the customers associate to the node have. For instance, if we have 1.000 customers associated to a node we may have the following distribution by label:



* That means, a new customer whose 10 parameters make him to be associated to this node will be more likely to buy label 12 (11- Popeyes 5+ 21,85%, followed by label 11(10 - Popeyes 3-4 15,97%) and label 2 (1 - Burgers\_Burger past 30 3-4 15,13%).

To be able to predict which dish people will prefer, or to analyze the likelihood of selling each dish in a given neighborhood we assume that the preferences will follow the patterns identified.

In this module we identify the best number of nodes which should be used to represent the aggregation of the samples. It is defined stablishing the minimum number of samples associated to each node. We vary the number of nodes and check the number of samples associated to each node (The percentage of the total sample). If there are nodes with a number of associated samples bellow a given percentage we reduce the number of nodes and run the process again.

Note that the variable “Threshold1” is the minimum percentage required to be a node.

Nodes\_num -> is the initial number of nodes to be tested.

He objective here is to cluster the samples in the most uniform way as possible. That means a bigger percentage concentrate in lees labels as possible.

But the How to define the initial number of nodes ? Here we have to follow the logic of the process:

Age – 15 classes

Geo – 10 Classes

Sex – 2 Classes

Language 01 – 19 Classes

Language 02 – 19 Classes

Education – 12 Classes

Employment – 3 Classes

Type of Job – 9 Classes

Income – 11 Classes

Ethnicity – 10 Classes

Here we would have 3.859.812.000 possible combinations of these 10 parameters.

Looking into our data we have 23.223 samples is clear that we will have less combinations than the ones possible. Verifying the combinations in these 23.223 samples we found that we have  19.756  different combinations there. That means if we had 19.756  nodes each one of them would have 100% of the samples associated to a specific label.

So the challenge is to go reducing these 19.756 combinations (nodes) without dispersing too much the % of the labels associated to the nodes.

Reducing the number of nodes to a point where the percentage of samples of one specific label  goes further than X % (Ideally above 50%), So de the objective would be to have enough nodes to allow a situation such as when a sample is associated to a node the chance that it belongs to a specific label is above X%.

We may consider combining the % of more than one label, given the fact that a client may order more than one dish and knowing that a given profile (Combination of factors –Node) can have different % of chance to order different dishes makes sense.

Therefore the threshold (criteria defining the best number of nodes) has to start in xxx e go down to a number of nodes where a given criteria is achieved.

Module 03 – stan-phase-03.py

To test how effective the process is we separated a subset of our customers 4.688 and associated them to the nodes and check if the actual label matches the more likely one