Milestone 3

Team Members: Rajat Aghi, Pal Doshi, Khushal Navani, Akash Udani

Data Collection and Cleaning Efforts:

a. The Consumer Finance Protection Bureau (CFPB) updates the Consumer Complaints data set fortnightly. We took the latest available data set for the purpose of this milestone.
b. Some of the variables in the new data set were blank (ex. ‘State’,‘Consumer.disputed.’). These rows were removed from the data set.
c. Just like in the previous milestone, we created a new variable called region which mapped all the states into 6 regions.

Research Question 1:
Predicting the response to a particular consumer complaint

Predictors: Product, Issue, Region Submitted.via,, Timely.response.,Consumer.disputed.
Outcome:Company.response.to.consumer (8 levels)
Sampling rate: (Training:Testing)=75:25

Question 1: Support Vector Machine
To apply this technique in RStudio, we used the kernlab library. Moreover, we had to reduce the size of our dataset from around 600,000 records to 200,000 records so that RStudio could handle it, else we ran into memory allocation problems. We used two different kernels to build our svm classifier and tested them out. Following are the results:

- **Kernel: Linear**
  The 'vanilladot' kernel in ksvm was used to create the classifier.

Confusion Matrix:

Predicted/Actual	**Closed**	**Closed with explanation**	**Closed with monetary relief**	**Closed with non-monetary relief**	**Closed with relief**	**Closed without relief**	**Untimely response**
Closed	0	0	0	0	0	0	0
Closed with explanation	1142	35623	3065	6107	369	1292	0
Closed with monetary relief	3	135	184	18	16	15	0
Closed with non-monetary relief	0	0	0	0	0	0	0
Closed with relief	0	0	0	0	0	0	0

Closed without relief	0	0	0	0	0	0	0
Untimely response	0	0	0	0	0	0	0

Classifier Measures:

Class-wise Measures	**Closed**	**Closed with explanation**	**Closed with monetary relief**	**Closed with non-monetary relief**	**Closed with relief**	**Closed without relief**	**Untimely response**
Recall	0	0.99622	0.056633	0	0	0	NA
Specificity	1	0.01933	0.995818	1	1	1	1
Precision	NaN	0.74841	0.495957	NaN	NaN	NaN	NA
Accuracy	0.5	0.50778	0.526226	0.5	0.5	0.5	NA
F1 Measure	NaN	0.854715338	0.101657767	NaN	NaN	NaN	NA

Overall Accuracy: 0.7465

- **Kernel: Non-linear (Gaussian)**
  The 'rbfdot' kernel in ksvm was used to create the classifier. This classifier uses the "one versus one" approach.

Confusion Matrix:

Predicted/Actual	**Closed**	**Closed with explanation**	**Closed with monetary relief**	**Closed with non-monetary relief**	**Closed with relief**	**Closed without relief**	**Untimely response**
Closed	0	0	0	0	0	0	0

Closed with explanation	1133	35475	2965	6000	365	1291	0
Closed with monetary relief	11	181	279	23	20	16	0
Closed with non-monetary relief	1	102	5	102	0	0	0
Closed with relief	0	0	0	0	0	0	0
Closed without relief	0	0	0	0	0	0	0
Untimely response	0	0	0	0	0	0	0

Classifier Measures:

Class-wise Measures	**Closed**	**Closed with explanation**	**Closed with monetary relief**	**Closed with non-monetary relief**	**Closed with relief**	**Closed without relief**	**Untimely response**
Recall	0	0.99209	0.085873	0.016653	0	0	NA
Specificity	1	0.03743	0.994387	0.997419	1	1	1
Precision	NaN	0.75113	0.526415	0.485714	NaN	NaN	NA

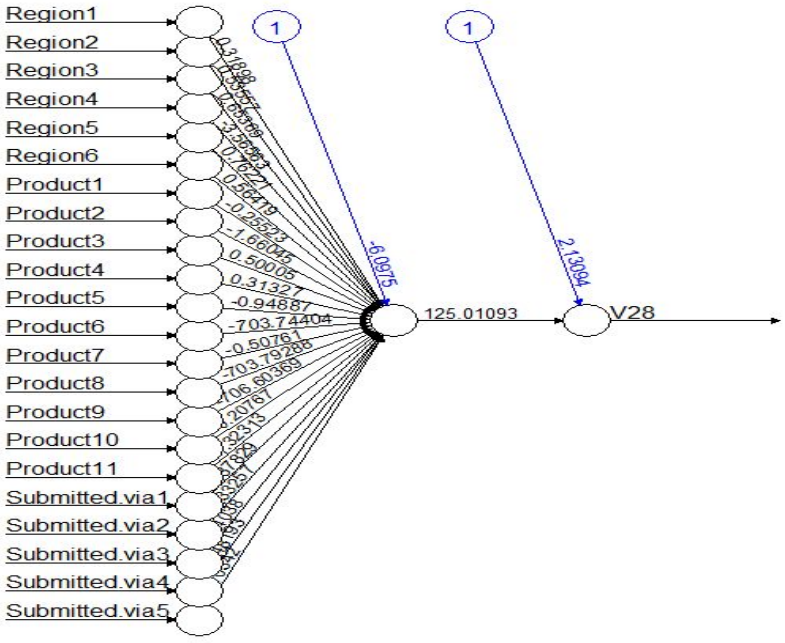
Accuracy	0.5	0.51476	0.54013	0.507036	0.5	0.5	NA
F1 Measure	NaN	0.854956416	0.147658733	0.032201937	NaN	NaN	NA

Overall Accuracy: 0.7475

Question 2: Neural Networks

Since all our variables (dependent and independent) are categorical, we could not apply neural networks directly. We used effect coding for all the dependent variables and numerical coding (one number for each class) for the independent variable. We had to reduce the size of our data set to 10,000, else RStudio we ran into memory allocation problems. We applied neural networks with 3 different number of hidden layers:

Model 1:
Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)
Hidden: 1
Activation Function: Logistic

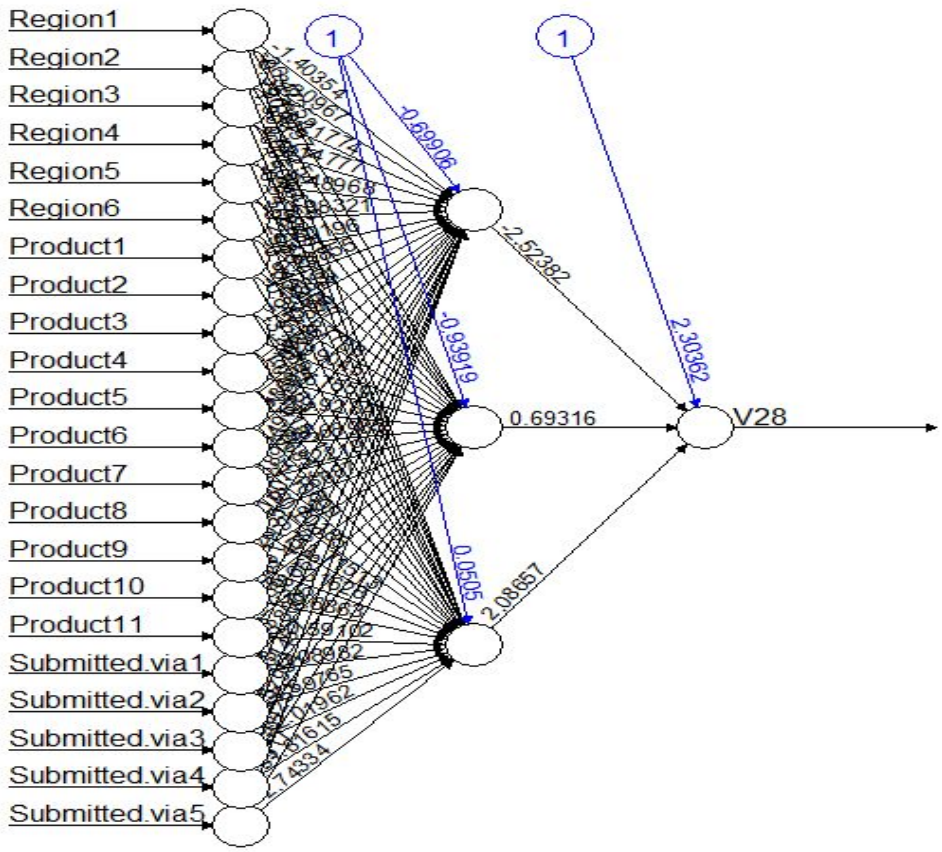


Correlation (b/w predicted and actual response): 0.003

We think that the correlation is low because the actual response is a categorical variable while the predicted value is a continuous variable.

Model 2:

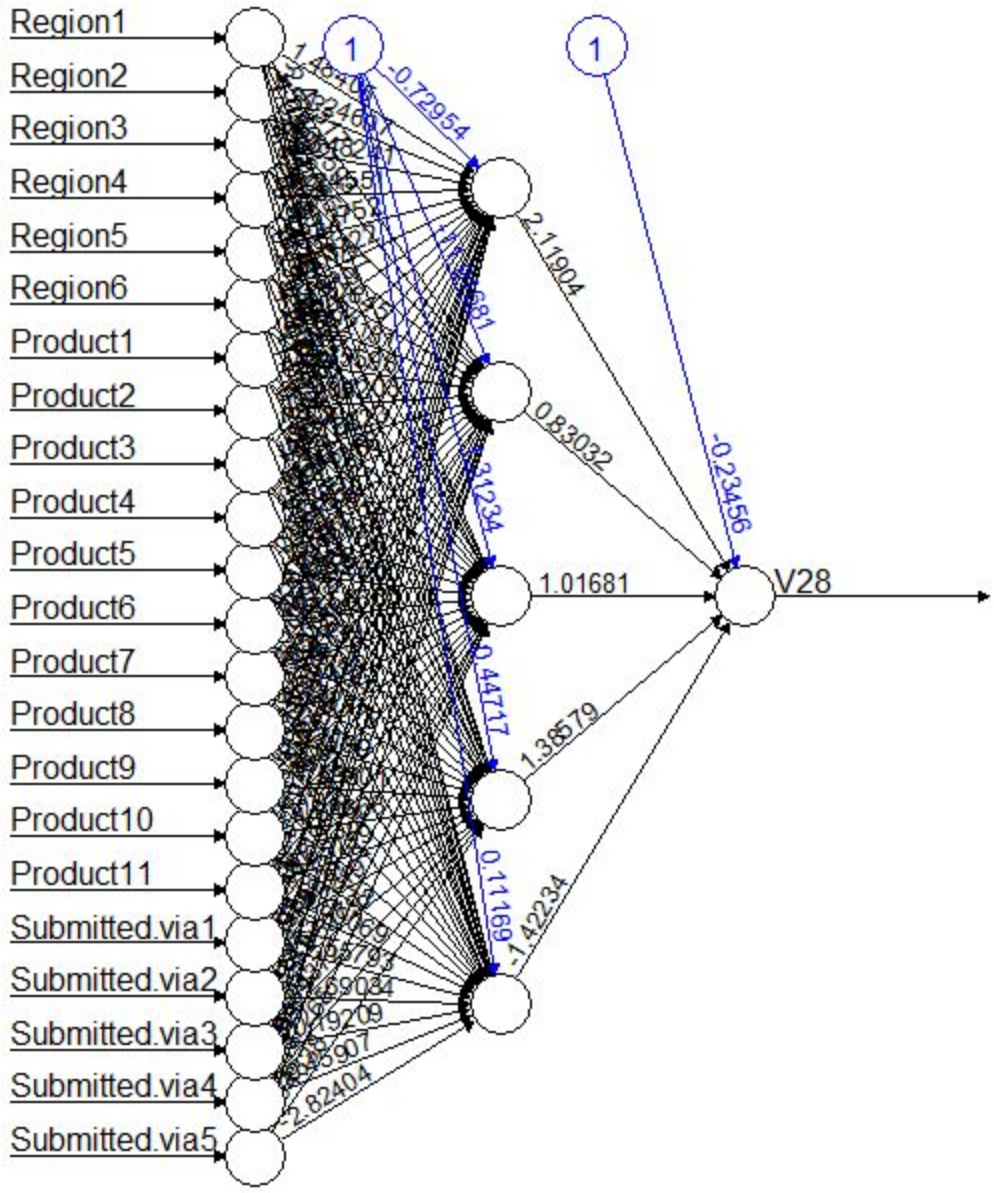
Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)
Hidden: 3
Activation Function: Logistic



Correlation (b/w predicted and actual response): 0.00009

Model 3:

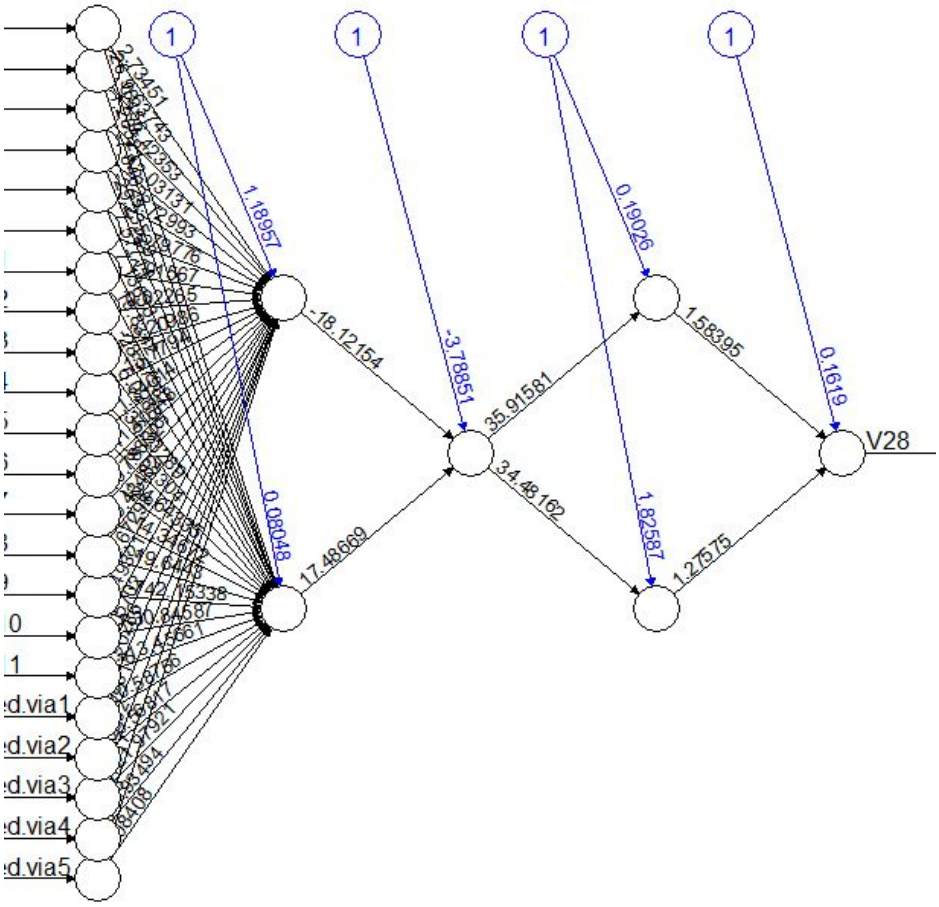
Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)
Hidden: 5
Activation Function: Logistic



Correlation (b/w predicted and actual response): 0.05

Model 4:

Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)
Hidden: 3 layers (2,1,2)
Activation Function: Logistic



Correlation (b/w predicted and actual response): 0.04

Question 3: Clustering

We applied different types of clustering techniques as follows

Technique 1: Hierarchical Clustering

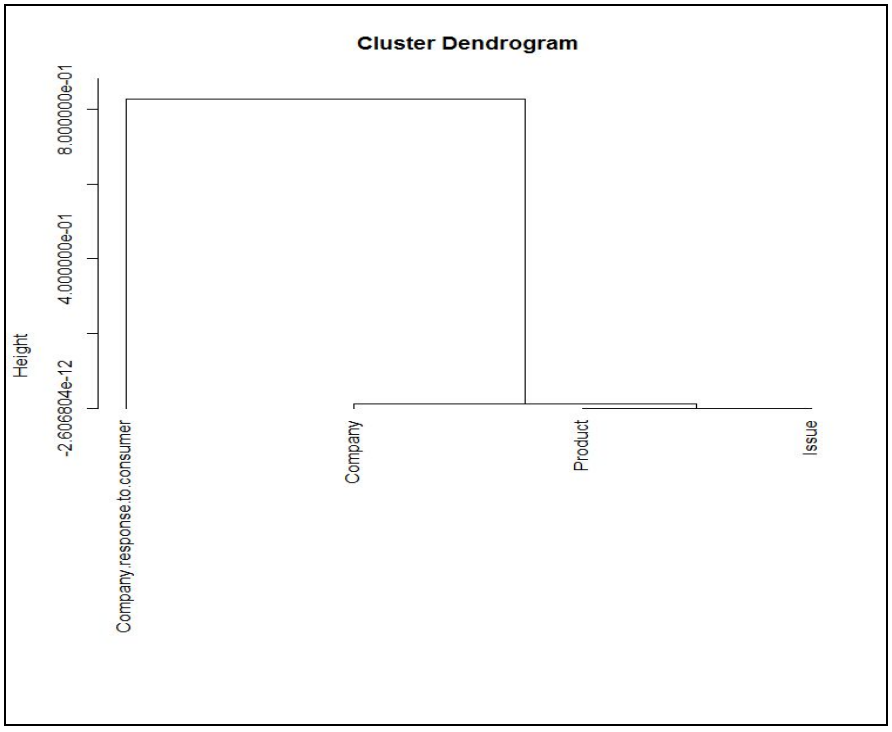
Since our entire dataset consists of categorical variables, using the normal 'hclust' method in R for performing hierarchical clustering is not possible. We found a package called **'ClustOfVar'** which provides methods specifically devoted to the clustering of variables with no restriction on the type (quantitative or qualitative) of the variables. The clustering methods developed in the package work with a mixture of quantitative and qualitative variables and also work for a set exclusively containing quantitative or qualitative variables. Also, missing data are allowed: they

are replaced by means for quantitative variables and by zeros in the indicator matrix for qualitative variables.

Variables used for generating the dendrogram:

- Company.response.to.consumer (8 levels)
- Product (12 levels)
- Issue (97 levels)
- Company (To reduce the number of levels, we took only the top 10 companies which account for more than 50% of the entire dataset)

We used a sample of approximately 300000 records for this test.
We used the **'hclustvar'** method in this package to generate a hierarchical tree structure. The dendrogram generated is as follows



```
Call:
cutreevar(obj = tree, k = 2, matsim = TRUE)



Data:
   number of observations:  314739
   number of variables:  4
   number of clusters:  2

Cluster  1 :
        squared loading
Product            1.00
Issue              1.00
Company            0.99


Cluster  2 :
                              squared loading
Company.response.to.consumer                1


Gain in cohesion (in %):  98.47
```

Similarity Matrix

```
$cluster1
          Product     Issue   Company
Product 1.0000000 1.0000000 0.9808154
Issue   1.0000000 1.0000000 0.9809108
Company 0.9808154 0.9809108 1.0000000

$cluster2
                            Company.response.to.consumer
Company.response.to.consumer                            1
```

The similarity matrix shows the similarities between variables for each cluster. For qualitative variables, the similarity between two variables is defined as the square of the canonical relation between two sets of dummy variables. In this case, the similarity is very high between ‘Product’ and ‘Company’ and between ‘Issue’ and ‘Company’. Hence they are grouped in one cluster.

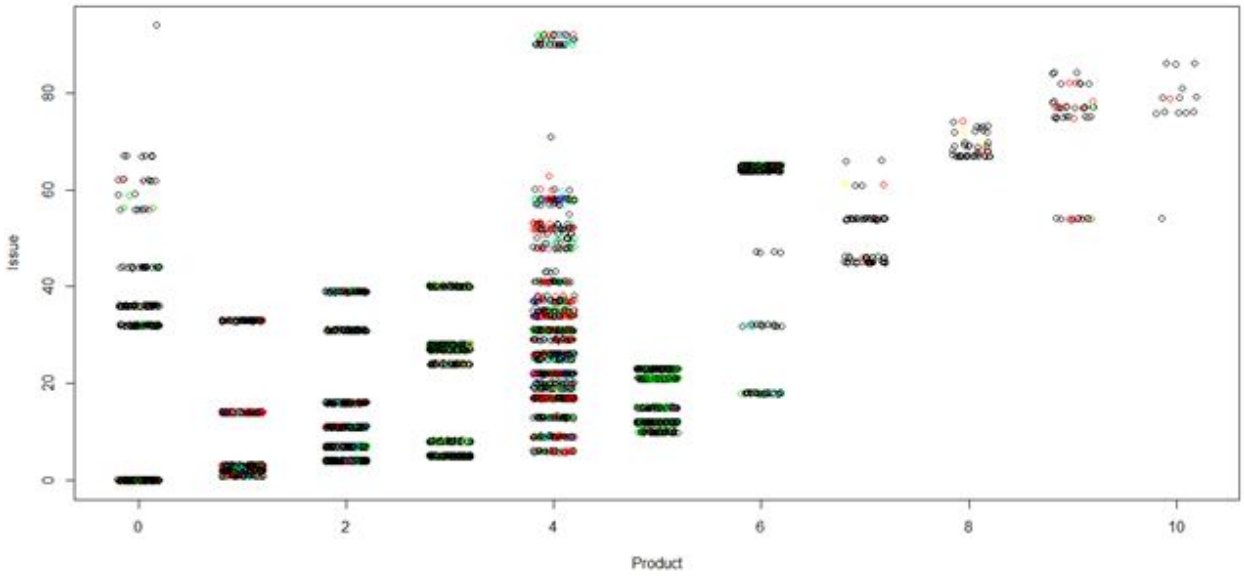
Technique 2: K-Modes clustering

To apply this technique, we have to make use of klaR library. Since our data has only categorical variables, we cannot use k-means and k-medoids as it is, nor we can use a distance formula based clustering method, like Euclidean or Gower distance, we have to make use of k-modes package. For ease of use, we have mapped the textual data to numerical factors. We reduced the data to 10,000 rows and ran the code for 3-4 iterations, as more data resulted in memory allocation issues in R Studio.

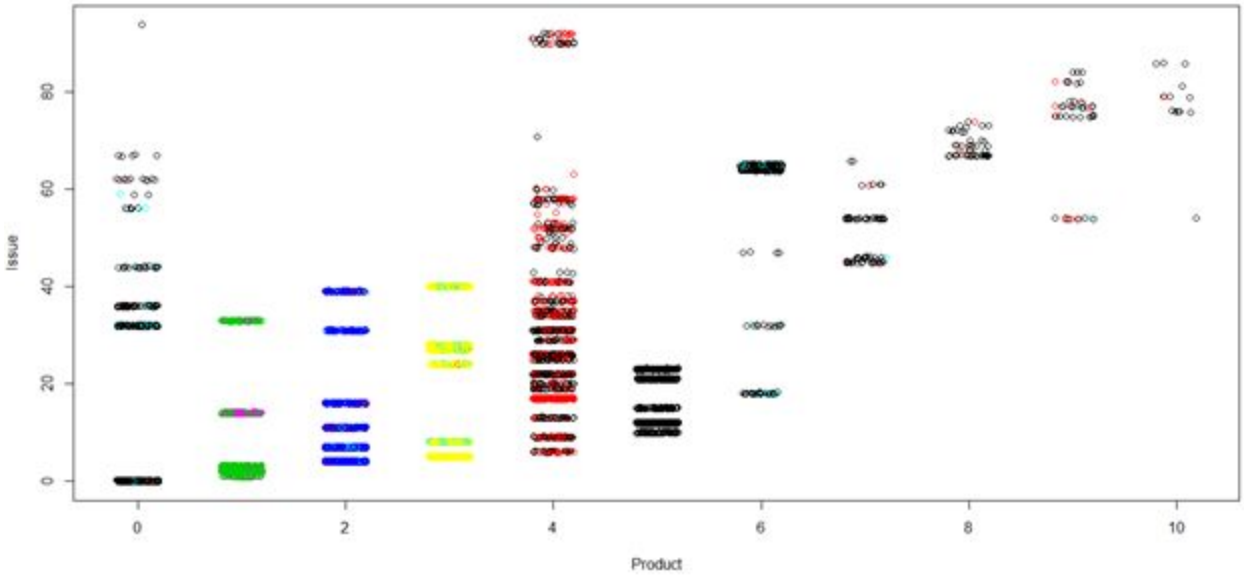
Columns used: Product, Issue, Company.Response.To.Consumer

```
> cl
K-modes clustering with 7 clusters of sizes 3168, 716, 949, 2871, 439, 158, 1699

> cl$modes
  Product Issue Company.response.to.consumer
1       5    12                            1
2       4    17                            2
3       1     2                            1
4       2     7                            1
5       2     7                            3
6       1    14                            2
7       3     5                            1
> cl$withindiff
[1] 3808  997  647 1652  571   94 1212
```

Scatter plot of Product vs Issue, colored on basis of Company response



Scatter plot of Product vs Issue, colored on basis of clusters so formed



Technique 3: Mixture Model Clustering

```
*****************************************************************
* Number of samples    =  50000
* Problem dimension    =  4
*****************************************************************
*       Number of cluster =  8
*             Model Type =  Binary_pk_Ekjh
*              Criterion =  BIC(503320.9708)
*             Parameters =  list by cluster
*                 Cluster  1 :
                      Proportion =  0.3588
                          Center =  7.0000 52.0000 1.0000 2.0000
                         Scatter = |     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0001     0.0000     0.0000     0.0000     0.0000
|
                                   |     0.0000     0.0000     0.0000     0.0000     0.0000     0.0598     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0
000     0.0206     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
 0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.4450     0.3044     0.0000     0.0000
     0.0000     0.0000     0.0000     0.0000     0.0000     0.0266     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.
0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0335     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
  0.0000     0.0000     0.0000 |
                                   |     0.6586     0.0078     0.0731     0.0002     0.0004     0.1312     0.1940     0.0001     0.2518 |
                                   |     0.0202     0.1878     0.0195     0.0746     0.0040     0.0652     0.0025     0.0019 |
*                 Cluster  2 :
                      Proportion =  0.0890
                          Center =  3.0000 13.0000 3.0000 2.0000
                         Scatter = |     0.0000     0.0000     0.0002     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
|
                                   |     0.0000     0.0000     0.0311     0.0000     0.0072     0.0000     0.0000     0.0671     0.0042     0.0115     0.0025
    0.0063     0.8569     0.0316     0.0000     0.0000     0.0035     0.0018     0.0000     0.0799     0.0145     0.0178     0.0000     0.0000     0.0017     0.0
291     0.0000     0.0386     0.0348     0.0000     0.0341     0.0000     0.0416     0.0000     0.0000     0.0463     0.0000     0.0000     0.0000     0.0000
 0.0000     0.0085     0.0000     0.0000     0.0924     0.0000     0.0000     0.0000     0.0000     0.0000     0.0198     0.0000     0.0000     0.0000     0.0000
     0.0000     0.0000     0.0000     0.0000     0.0000     0.1055     0.0175     0.0000     0.0000     0.0000     0.0038     0.0000     0.0305     0.0065     0.
0000     0.0000     0.0000     0.0000     0.0256     0.0051     0.0000     0.0000     0.0000     0.0000     0.0000     0.0216     0.0000     0.0000     0.0000
  0.0148     0.0000     0.0000 |
                                   |     0.2499     0.3204     0.6699     0.0009     0.0009     0.0000     0.0000     0.0002     0.0974 |
                                   |     0.0102     0.2592     0.0010     0.1271     0.0260     0.0905     0.0031     0.0013 |
*                 Cluster  3 :
                      Proportion =  0.1000
                          Center =  1.0000 1.0000 9.0000 2.0000
                         Scatter = |     0.0502     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0501
|
                                   |     0.5710     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0096     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0
000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0194     0.0000     0.2564     0.0000     0.0000     0.0000
 0.0000     0.0000     0.0000     0.0040     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
     0.0840     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.
1138     0.0003     0.0000     0.0168     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
  0.0000     0.0666     0.0000 |
                                   |     0.3944     0.0620     0.1300     0.0006     0.0008     0.0010     0.0024     0.0008     0.5922 |
                                   |     0.0118     0.3501     0.1946     0.0846     0.0209     0.0329     0.0048     0.0004 |
```

```
*                Cluster  4 :
                       Proportion =  0.2351
                            Center =  4.0000 49.0000 4.0000 2.0000
                           Scatter = |     0.0000     0.0000     0.0000     0.0001     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
|
                                     |     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0
000     0.0000     0.0000     0.0000     0.0000     0.0298     0.0000     0.1343     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
 0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0428     0.0000     0.2617     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.
0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0546     0.0000     0.0000
  0.0000     0.0000     0.0000 |
                                     |     0.0009     0.0028     0.0014     0.5724     0.2446     0.0001     0.0003     0.3205     0.0018 |
                                     |     0.0002     0.0070     0.0000     0.0065     0.0000     0.0000     0.0003     0.0000 |
*                Cluster  5 :
                       Proportion =  0.0415
                            Center =  5.0000 24.0000 9.0000 2.0000
                           Scatter = |     0.0000     0.3682     0.0000     0.0000     0.3820     0.0000     0.0000     0.0000     0.0000     0.0000     0.0134
|
                                     |     0.0000     0.0140     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0010     0.0000     0.0000     0.0000     0.1194     0.7791     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0
000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.1449     0.0000     0.0366
 0.0000     0.0000     0.0000     0.0000     0.0000     0.0375     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0217     0.1858     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0014     0.0000     0.0000     0.
0000     0.0903     0.0000     0.0000     0.0000     0.0000     0.0000     0.0063     0.0226     0.0385     0.0587     0.0000     0.0000     0.0000     0.0000
  0.0000     0.0000     0.0000 |
                                     |     0.1742     0.2282     0.2295     0.0049     0.0082     0.0313     0.0487     0.0053     0.7304 |
                                     |     0.0145     0.2133     0.0573     0.1204     0.0043     0.0118     0.0044     0.0005 |
*                Cluster  6 :
                       Proportion =  0.1406
                            Center =  4.0000 49.0000 5.0000 4.0000
                           Scatter = |     0.0000     0.0000     0.0000     0.0001     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
|
                                     |     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0
000     0.0000     0.0000     0.0000     0.0000     0.0366     0.0000     0.1078     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
 0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0259     0.0000     0.2991     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.
0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.1286     0.0000     0.0000
  0.0000     0.0000     0.0000 |
                                     |     0.0009     0.0024     0.0024     0.2484     0.5002     0.0000     0.0000     0.2453     0.0008 |
                                     |     0.0062     0.1841     0.0156     0.2320     0.0000     0.0000     0.0259     0.0000 |
*                Cluster  7 :
                       Proportion =  0.0316
                            Center =  3.0000 13.0000 3.0000 3.0000
                           Scatter = |     0.0001     0.0001     0.0006     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001
|
                                     |     0.0000     0.0000     0.0173     0.0000     0.0000     0.0000     0.0000     0.0000     0.0386     0.0015     0.0086     0.0000
    0.0014     0.7490     0.0355     0.0000     0.0000     0.0055     0.0057     0.0000     0.0500     0.0055     0.0057     0.0000     0.0000     0.0010     0.0000     0.0
465     0.0000     0.0021     0.0045     0.0000     0.0115     0.0000     0.0423     0.0000     0.0000     0.0259     0.0000     0.0000     0.0000     0.0000     0.0000
 0.0000     0.0032     0.0000     0.0000     0.0939     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.1219     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0591     0.0645     0.0000     0.0000     0.0000     0.0000     0.0057     0.0000     0.0331     0.0000     0.
0000     0.0000     0.0000     0.0000     0.0000     0.0177     0.0014     0.0000     0.0000     0.0000     0.0000     0.0000     0.0328     0.0000     0.0000     0.0000
  0.0064     0.0000     0.0000 |
                                     |     0.1306     0.2621     0.4571     0.0001     0.0007     0.0001     0.0001     0.0001     0.0634 |
                                     |     0.0005     0.0884     0.1502     0.0007     0.0571     0.0008     0.0010     0.0016 |

*                Cluster  8 :
                       Proportion =  0.0033
                            Center =  6.0000 43.0000 1.0000 2.0000
                           Scatter = |     0.0005     0.0006     0.0005     0.0005     0.0005     0.5443     0.0005     0.1921     0.0544     0.2940     0.0005
|
                                     |     0.0001     0.0001     0.0060     0.0120     0.0001     0.0001     0.0240     0.0001     0.0001     0.0001     0.0001
    0.0001     0.0001     0.0001     0.0061     0.0001     0.0001     0.0001     0.0181     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0
001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0480     0.0001     0.0001     0.0001     0.0001     0.0120     0.0001
 0.0420     0.0001     0.8083     0.0001     0.0001     0.0001     0.0001     0.0180     0.0001     0.0240     0.0001     0.0001     0.0001     0.0180     0.0120
    0.0001     0.0001     0.0001     0.1258     0.1019     0.0001     0.0001     0.0180     0.1438     0.0180     0.0001     0.0001     0.0001     0.0001     0.
0001     0.0001     0.0061     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0001     0.0779     0.0300
  0.0001     0.0001     0.0420 |
                                     |     0.6581     0.0725     0.2762     0.0067     0.0126     0.0007     0.0007     0.0007     0.2881 |
                                     |     0.0127     0.2747     0.2163     0.0427     0.0007     0.0007     0.0007     0.0007 |
*          Log-likelihood =  -246818.6346
*****************************************************************
```

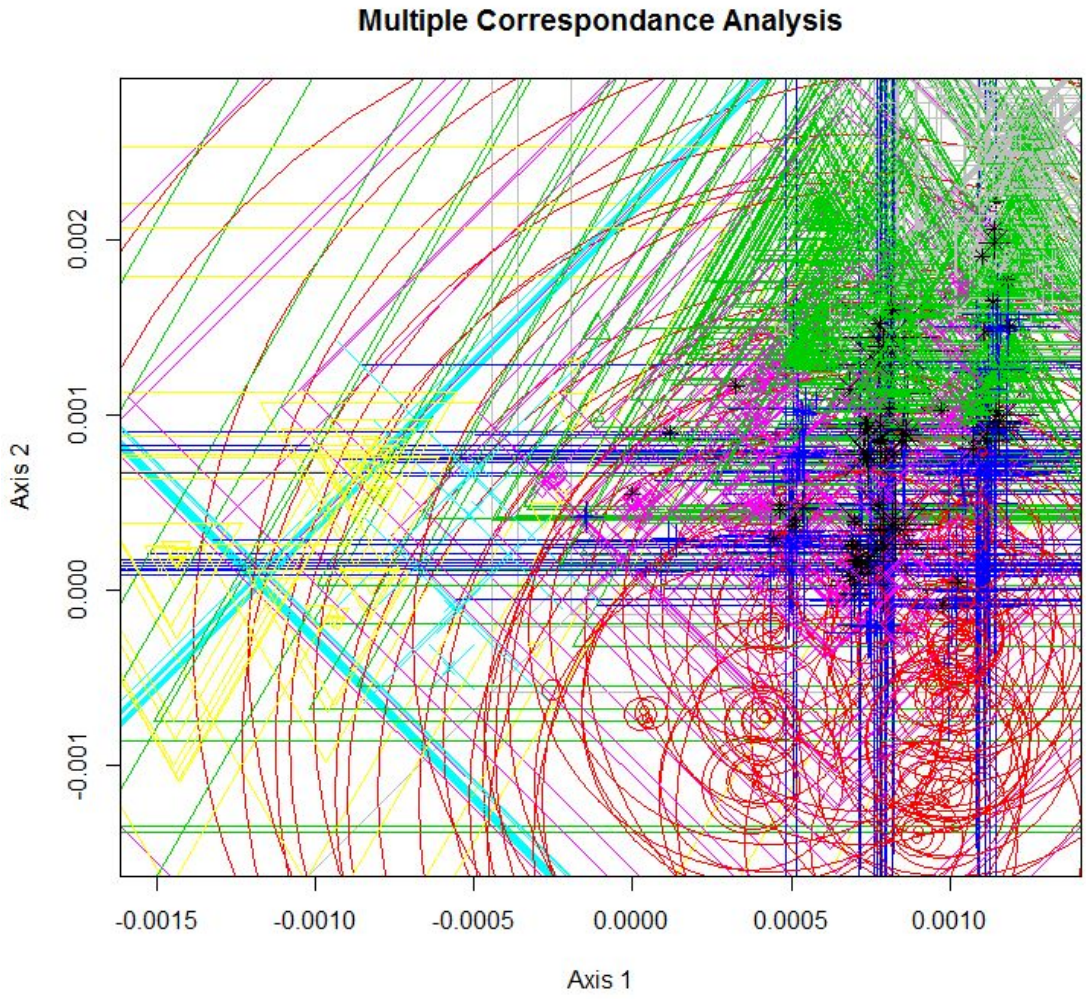
Proportion of records in Cluster1: 0.3588
Proportion of records in Cluster2: 0.0890
Proportion of records in Cluster3: 0.10
Proportion of records in Cluster4: 0.2351
Proportion of records in Cluster5: 0.0415
Proportion of records in Cluster6: 0.1406
Proportion of records in Cluster7: 0.0316
Proportion of records in Cluster8: 0.0033

Plot of the result

The plot() function gives the following output.

A multiple correspondence analysis is performed to get a 2-dimensional representation of the dataset and a bigger symbol is used when observations are similar.

Each shape/color represents a different cluster.



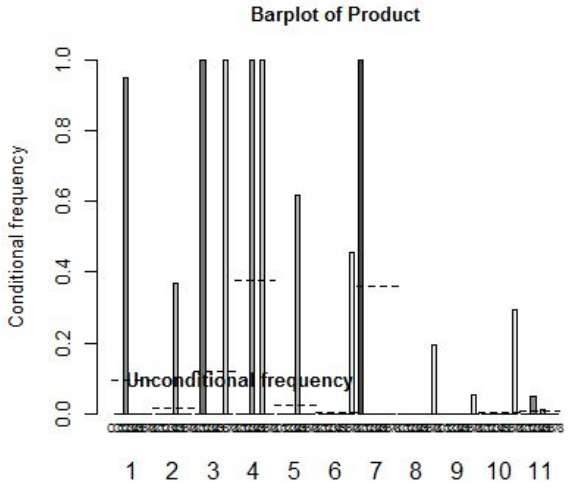
Barplot of the result

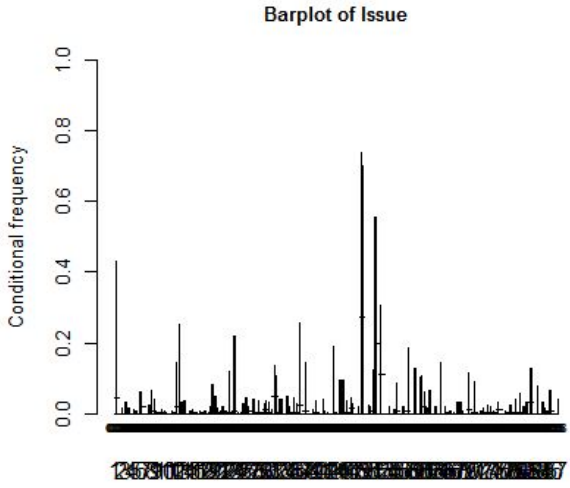
The barplot() function gives the following output.

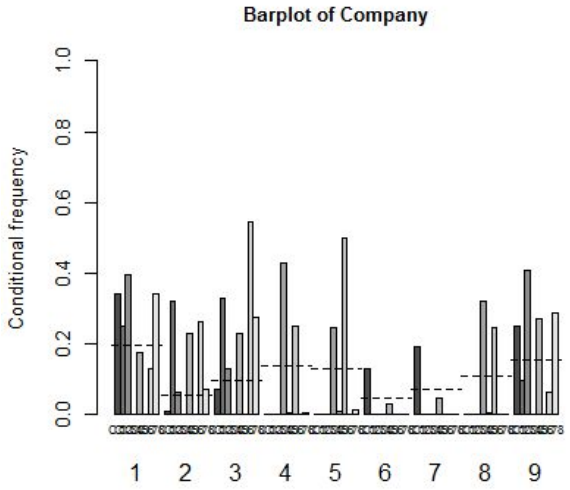
For each qualitative variable, we obtain:

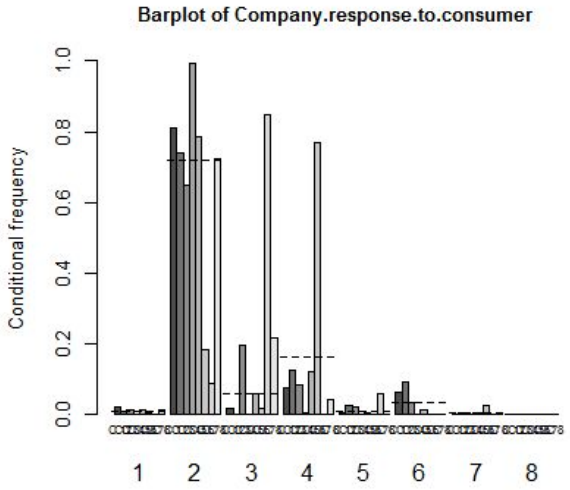
• a barplot with the frequencies of the modalities;

• for each cluster a barplot with the probabilities for each modality to be in that cluster.


Barplot of Product
Conditional frequency
Unconditional frequency
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10 11


Barplot of Issue
Conditional frequency
0.0
0.2
0.4
0.6
0.8
1.0


Barplot of Company
Conditional frequency
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9


Barplot of Company.response.to.consumer
Conditional frequency
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8

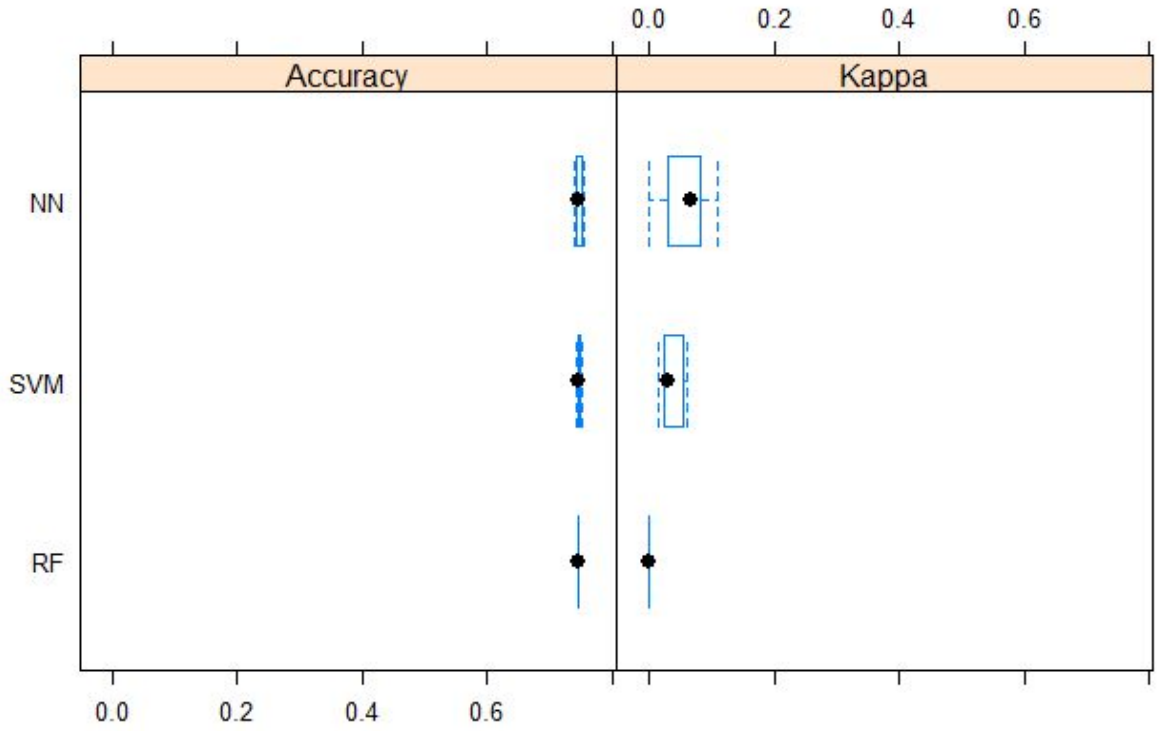
Question 4: Comparative Analysis

We compared three models: Support Vector Machine, Neural Networks and Random Forests using the caret library for three different CV methods: K-Fold, Bootstrap and Repeated K-Fold. Following are the results:

- **K-Fold with 10 folds**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.7383	0.7443	0.7472	0.7474	0.751	0.7542
RF	0.7449	0.745	0.7457	0.7456	0.746	0.7466
SVM	0.7439	0.7451	0.7466	0.7474	0.7496	0.7526

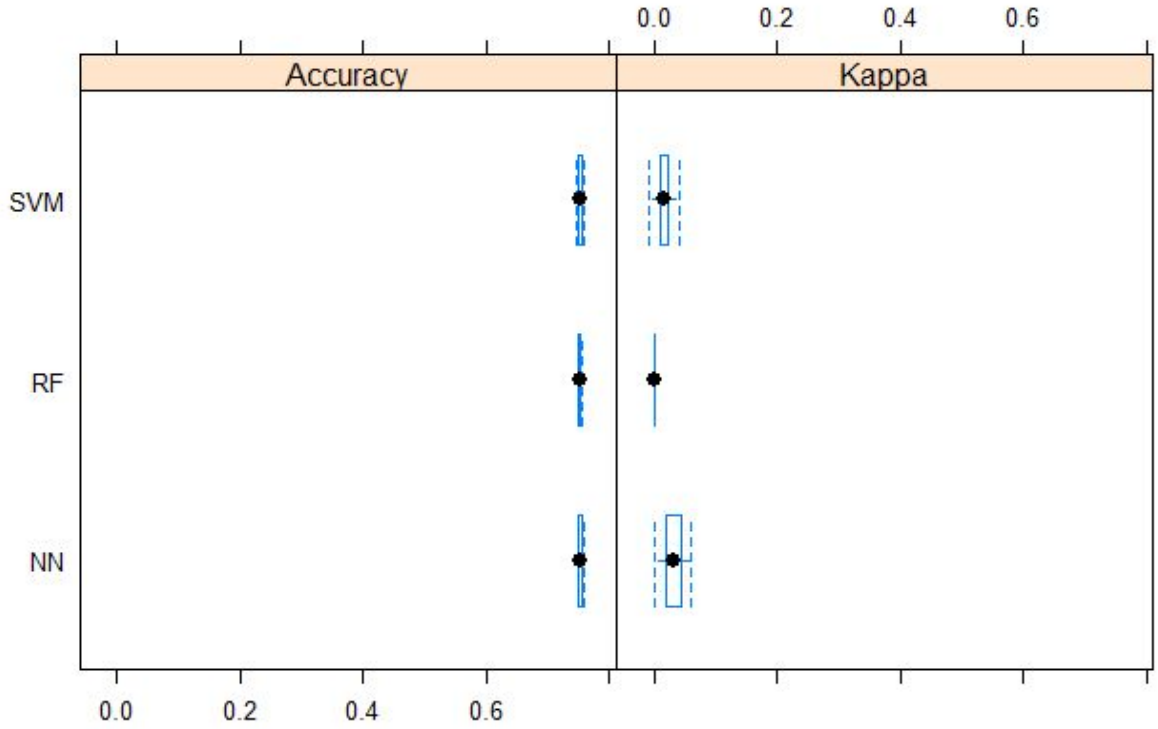


From the above plot, it can be observed that the accuracy for the three techniques are in the same range. But, out of the three, Neural networks is the most accurate.

- **K-Fold with 10 folds and 3 repeats**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.748	0.7505	0.752	0.7525	0.7544	0.7579
RF	0.7492	0.7508	0.752	0.7519	0.7528	0.7544
SVM	0.7468	0.7508	0.752	0.7522	0.7542	0.7575



Rather surprising results are observed in this case. All three models show the exact same (median) accuracy. In this case, we observe the Kappa values.

Kappa:

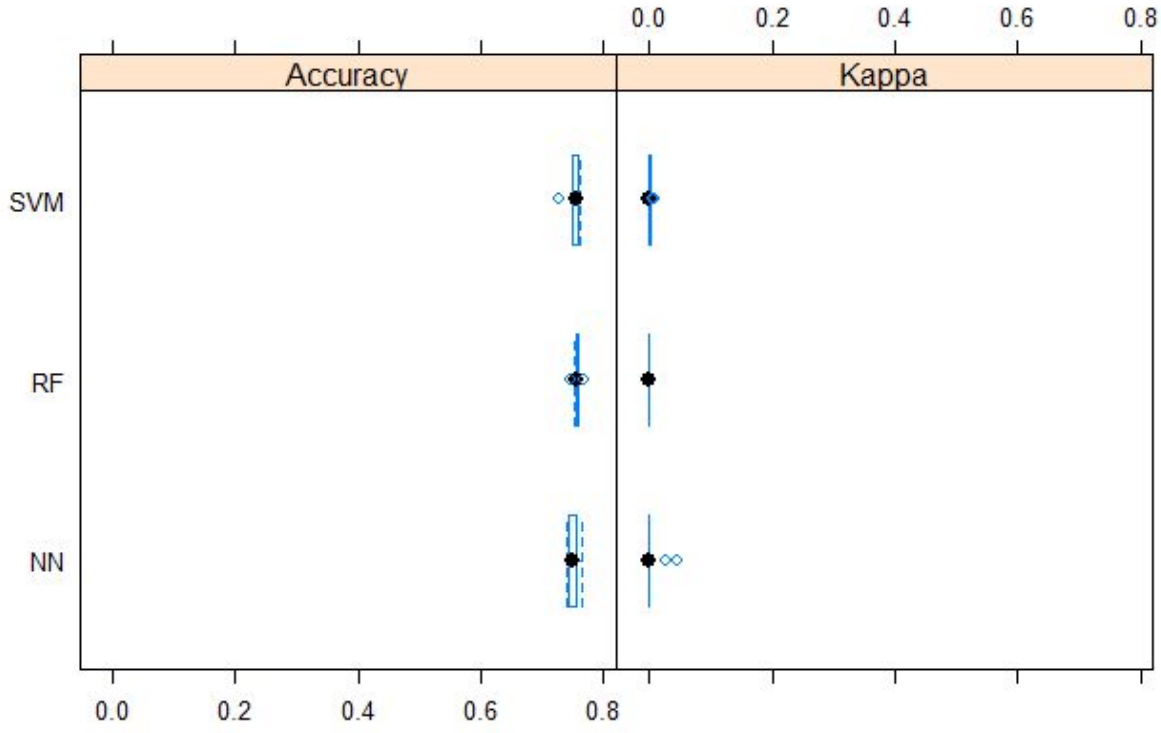
Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0	0.01965	0.0318	0.03111	0.04378	0.06049
RF	0	0	0	0	0	0
SVM	-0.007934	0.008702	0.01492	0.0143	0.02126	0.04005

Looking at the Kappa values, Neural networks seems to be the best model for this question.

- **Bootstrap**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.7387	0.7442	0.7492	0.7503	0.7544	0.7651
RF	0.7473	0.7541	0.755	0.7558	0.7574	0.7669
SVM	0.7274	0.7504	0.7556	0.7528	0.7593	0.7619



From the above plot, it can be observed that the accuracy for SVM is slightly better than Random Forests.

Research Question 2:
Predict if a consumer would dispute the company's feedback or not

Predictors: Product, Issue, state, Submitted.via, Company.response.to.consumer, Timely.response.
Outcome: Consumer.disputed.
Sampling rate: (Training:Testing)=75:25

Question 1: Support Vector Machine
To apply this technique in RStudio, we used the kernlab library. Moreover, we had to reduce the size of our dataset from around 600,000 records to 100,000 records so that RStudio could handle it, else we ran into memory allocation problems. We used two different kernels to build our svm classifier and tested them out. Following are the results:

- **Kernel: Linear**
  The 'vanilladot' kernel in ksvm was used to create the classifier.

Confusion Matrix:

Predicted/Actual	**No**	**Yes**
No	6238	1665
Yes	0	0

Classifier Measures:

Class-wise Measures	
Recall	1
Specificity	0
Precision	0.7893
Accuracy	0.5

F1 Measure	0.882244453

Overall Accuracy: 0.7893

- **Kernel: Non-linear (Gaussian)**
  The 'rbfdot' kernel in ksvm was used to create the classifier. This classifier uses the "one versus one" approach.

Confusion Matrix:

Predicted/Actual	**No**	**Yes**
No	6238	1665
Yes	0	0

Classifier Measures:

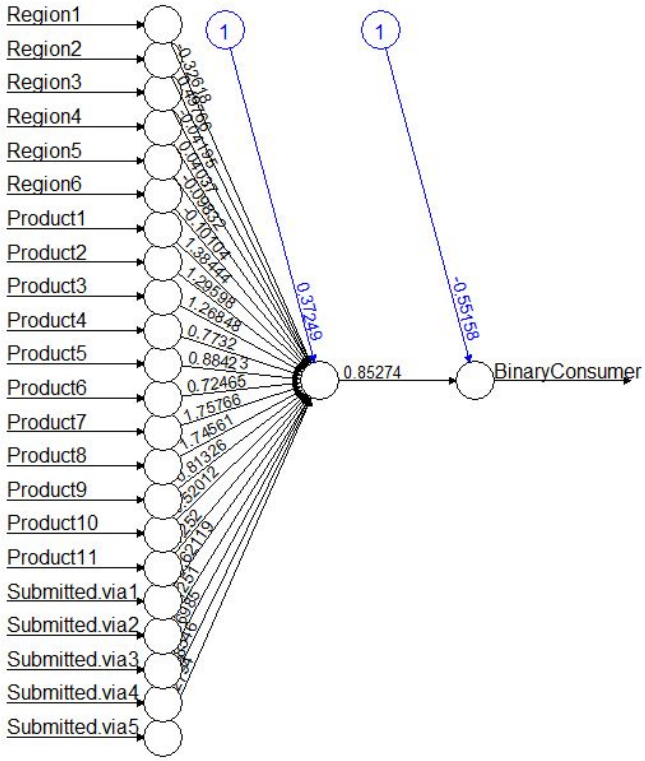
Class-wise Measures	
Recall	1
Specificity	0
Precision	0.7893
Accuracy	0.5
F1 Measure	0.882244453

Overall Accuracy: 0.7893

Question 2: Neural Networks

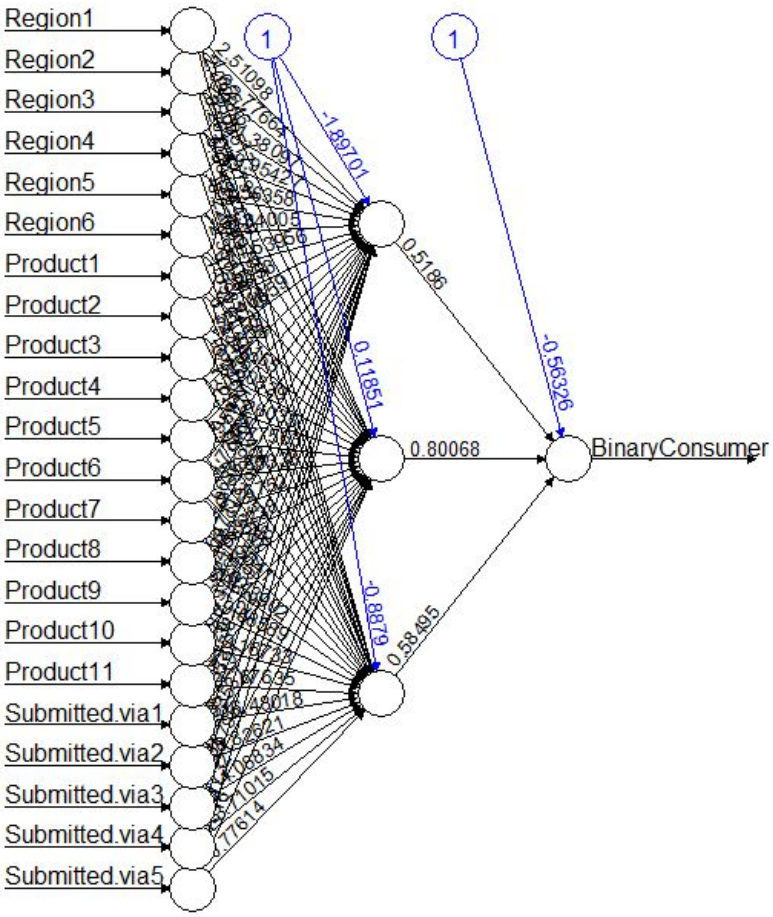
Since all our variables (dependent and independent) are categorical, we could not apply neural networks directly. We used effect coding for all the dependent variables and numerical coding (one number for each class) for the independent variable. We had to reduce the size of our data set to 10,000, else RStudio we ran into memory allocation problems. We applied neural networks with 3 different number of hidden layers:

Model 1:
Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)
Hidden: 1
Activation Function: Logistic



Correlation (between predicted and actual response): -0.04

Model 2:
Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)
Hidden: 3
Activation Function: Logistic



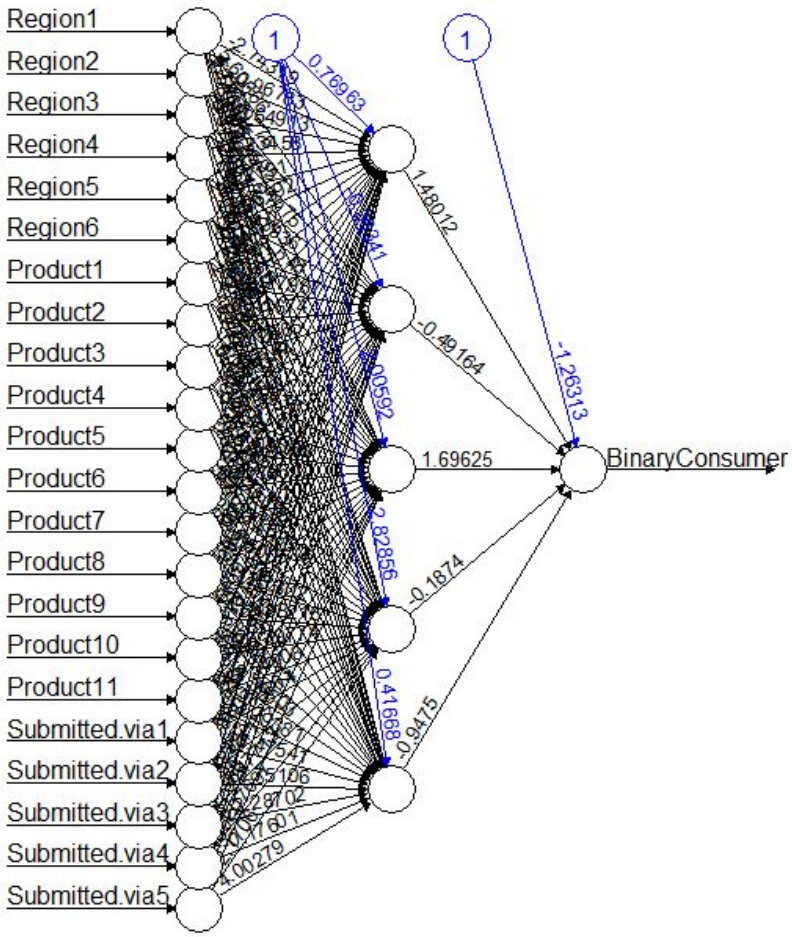
Correlation (between predicted and actual response): 0.04

Model 3:

Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)

Hidden: 5

Activation Function: Logistic



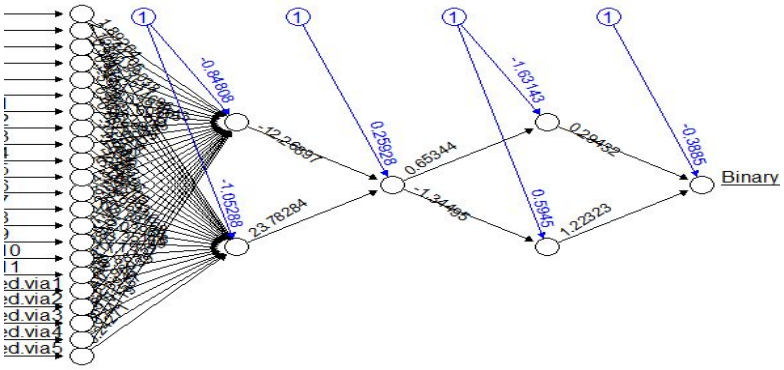
Correlation (between predicted and actual response): -0.04

<u>Model 4:</u>

Dependent Variables: Region (6 levels), Product (11 levels), Submitted.via (5 levels)

Hidden: 3 layers (2,1,2)

Activation Function: Logistic



Correlation: 0.02

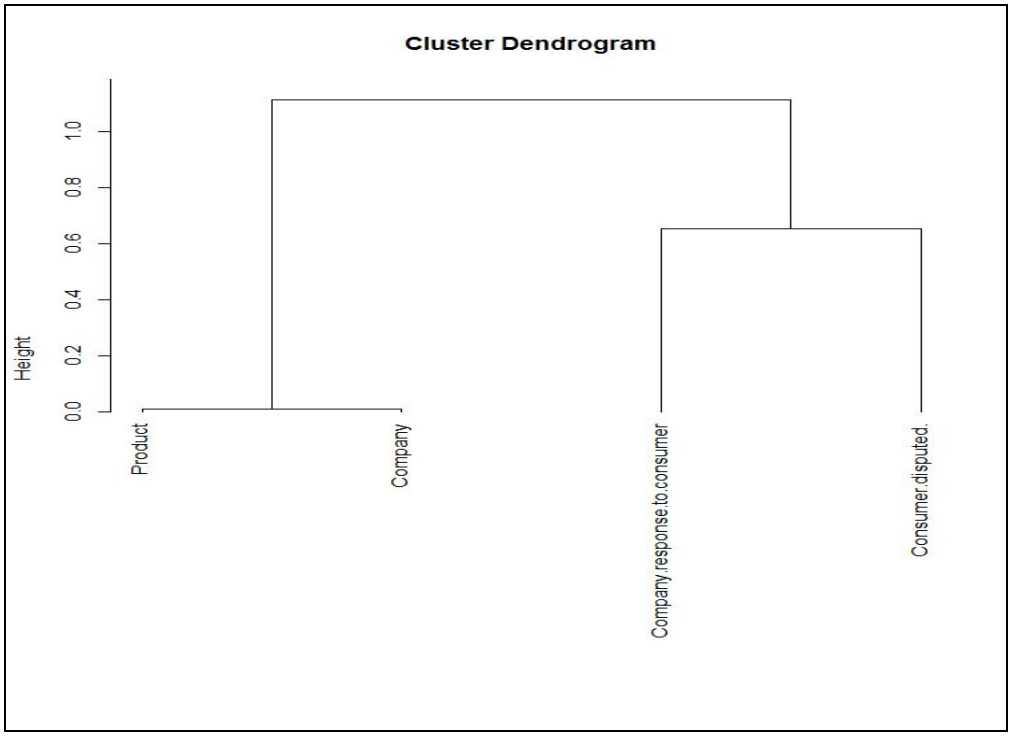
Question 3: Clustering
We applied different types of clustering as follows

Technique 1: Hierarchical Clustering
Since our entire dataset consists of categorical variables, using the normal 'hclust' method in R for performing hierarchical clustering is not possible. We found a package called **'ClustOfVar'** which provides methods specifically devoted to the clustering of variables with no restriction on the type (quantitative or qualitative) of the variables. The clustering methods developed in the package work with a mixture of quantitative and qualitative variables and also work for a set exclusively containing quantitative or qualitative variables. Also, missing data are allowed: they are replaced by means for quantitative variables and by zeros in the indicator matrix for qualitative variables.

Variables used for generating the dendrogram:

- Company.response.to.consumer (8 levels)
- Product (12 levels)
- Company (To reduce the number of levels, we took only the top 10 companies which account for more than 50% of the entire dataset)
- Consumer.disputed (2 levels)

We used a sample of approximately 300000 records for this test.
We used the **'hclustvar'** method in this package to generate a hierarchical tree structure. The dendrogram generated is as follows




```
Call:
cutreevar(obj = tree, k = 2, matsim = TRUE)



Data:
   number of observations:  314739
   number of variables:  4
   number of clusters:  2

Cluster  1 :
        squared loading
Product               1
Company               1


Cluster  2 :
                              squared loading
Company.response.to.consumer             0.67
Consumer.disputed.                       0.67


Gain in cohesion (in %):  62.68
```

Similarity matrix

```
$cluster1
          Product   Company
Product 1.0000000 0.9808154
Company 0.9808154 1.0000000

$cluster2
                                Company.response.to.consumer Consumer.disputed.
Company.response.to.consumer                       1.0000000          0.1207653
Consumer.disputed.                                 0.1207653          1.0000000
```

The similarity matrix shows the similarities between variables for each cluster. For qualitative variables, the similarity between two variables is defined as the square of the canonical relation between two sets of dummy variables. In this case, the similarity is very high between 'Product' and 'Company'. Hence they are grouped in Cluster 1. The ratio value is less between 'Company.response.to.consumer' and 'Consumer.disputed' and they are grouped together in Cluster 2.

Technique 2: K-Modes clustering

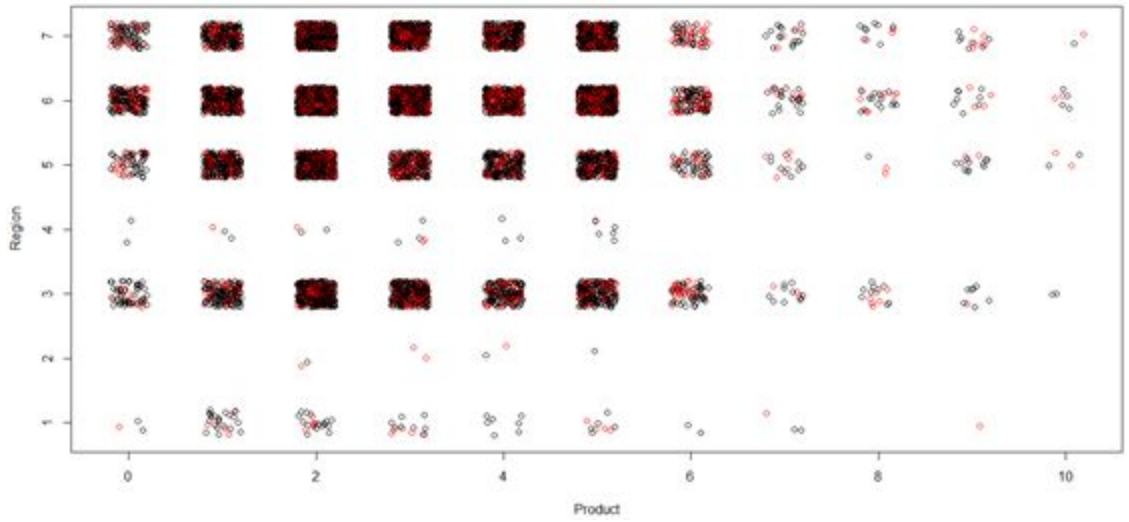
To apply this technique, we have to make use of klaR library. Since our data has only categorical variables, we cannot use k-means and k-medoids as it is, nor we can use a distance formula based clustering method, like Euclidean or Gower distance, we have to make use of k-modes package. For ease of use, we have mapped the textual data to numerical factors. We reduced the data to 10,000 rows and ran the code for 3-4 iterations, as more data resulted in memory allocation issues in R Studio.

Columns used: Product, Region, Submitted.via, Consumer.disputed

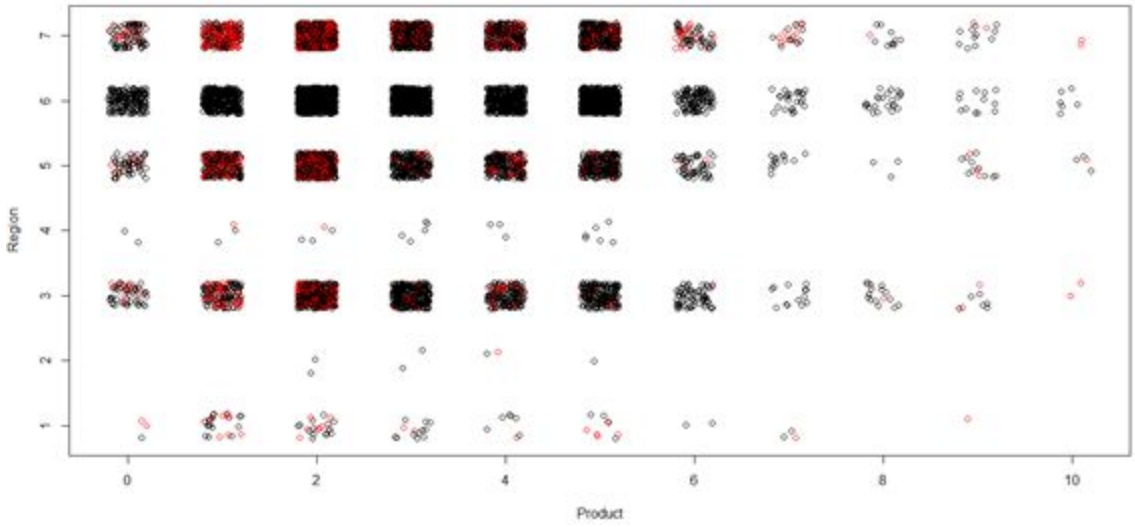
```
> cl
K-modes clustering with 2 clusters of sizes 8487, 1513

Cluster modes:
  Product Region Submitted.via Consumer.disputed.
1       2      4             5                  0
2       2      5             4                  0
```

```
> cl$withindiff
[1] 14585  2222
```

Scatter Plot of Product vs Region, colored on basis of Consumer Disputed



Scatter Plot of Product vs Region, colored on basis of clusters formed



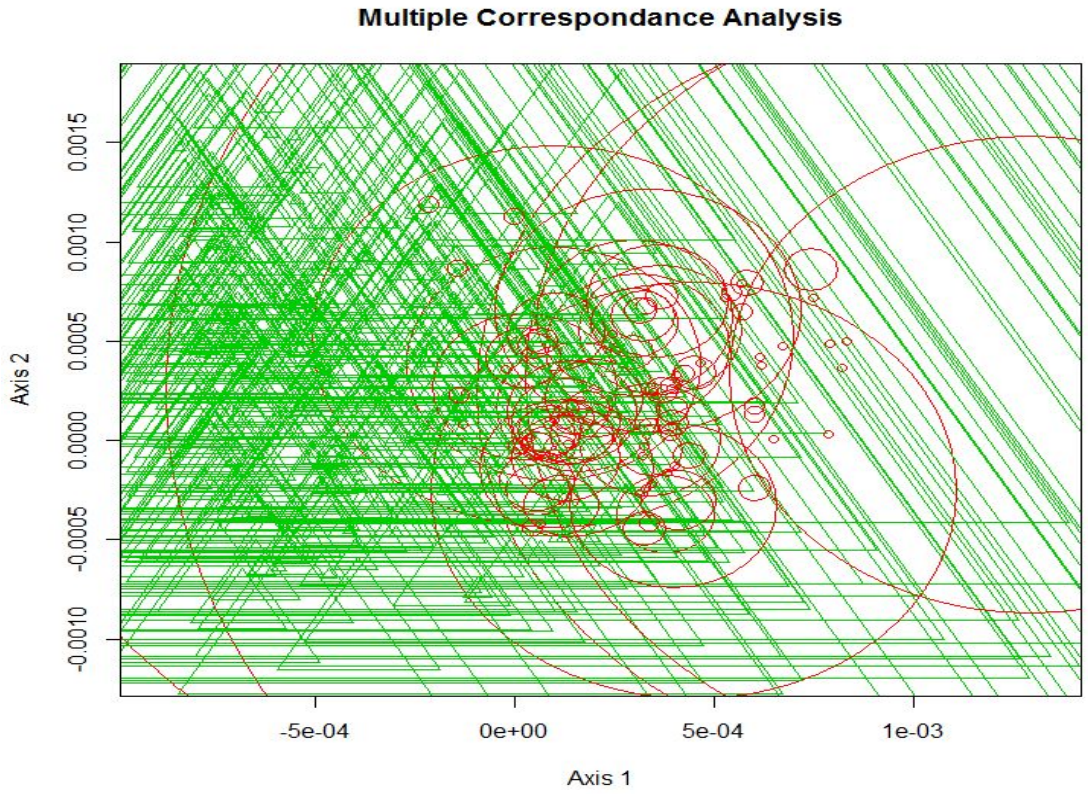
Technique 3: Mixture Model Clustering

```
************************************************************
* Number of samples    =  100000
* Problem dimension    =  4
************************************************************
*       Number of cluster =  2
*             Model Type =  Binary_pk_Ekjh
*              Criterion =  BIC(892570.1563)
*             Parameters =  list by cluster
*                 Cluster  1 :
                        Proportion =  0.3737
                            Center =  4.0000 4.0000 2.0000 2.0000
                           Scatter = |     0.0003     0.0008     0.0006     0.0044     0.0016     0.0000     0.0005     0.0001     0.0000     0.0000     0.0005
    0.0000 |
                                     |     0.0007     0.0024     0.0022     0.6390     0.3414     0.0000     0.0001     0.2916     0.0006 |
                                     |     0.0025     0.3062     0.0049     0.2880     0.0000     0.0000     0.0108     0.0000 |
                                     |     0.0823     0.2442     0.1620 |
*                 Cluster  2 :
                        Proportion =  0.6263
                            Center =  7.0000 1.0000 2.0000 2.0000
                           Scatter = |     0.1535     0.0241     0.1941     0.0008     0.0414     0.0026     0.4280     0.0007     0.0002     0.0013     0.0093
    0.0000 |
                                     |     0.6865     0.0907     0.1577     0.0001     0.0002     0.0769     0.1136     0.0000     0.2472 |
                                     |     0.0165     0.2654     0.0907     0.0870     0.0121     0.0545     0.0032     0.0013 |
                                     |     0.0359     0.2541     0.2182 |
*          Log-likelihood =  -445956.9598
************************************************************
```

Proportion of records in Cluster1: 0.3737
Proportion of records in Cluster2: 0.6263

Plot of the result

The plot() function gives the following output.
A multiple correspondence analysis is performed to get a 2-dimensional representation of the dataset and a bigger symbol is used when observations are similar.



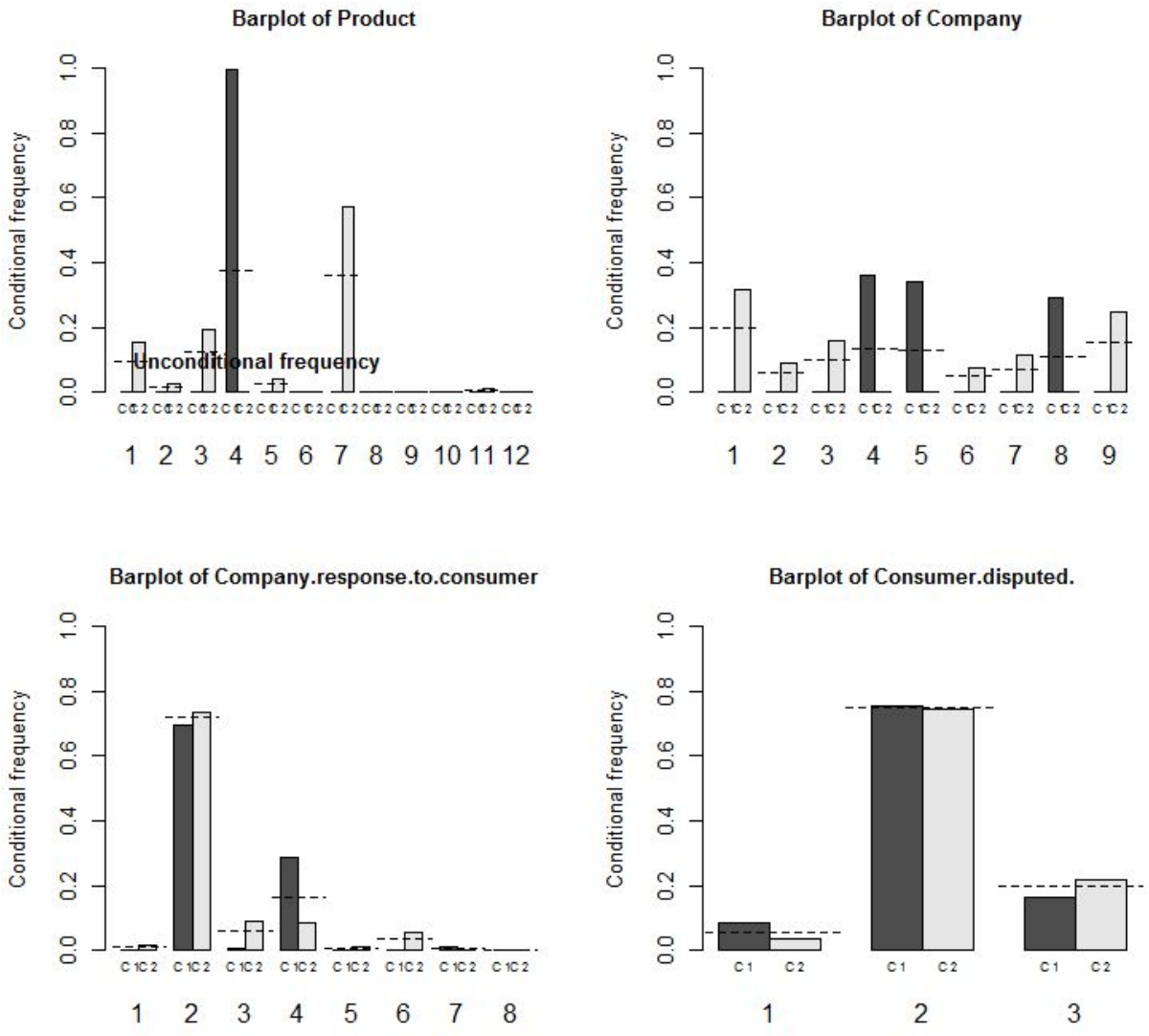
Barplot of the result

The barplot() function gives the following output.

For each qualitative variable, we obtain:

• a barplot with the frequencies of the modalities;

• for each cluster a barplot with the probabilities for each modality to be in that cluster.



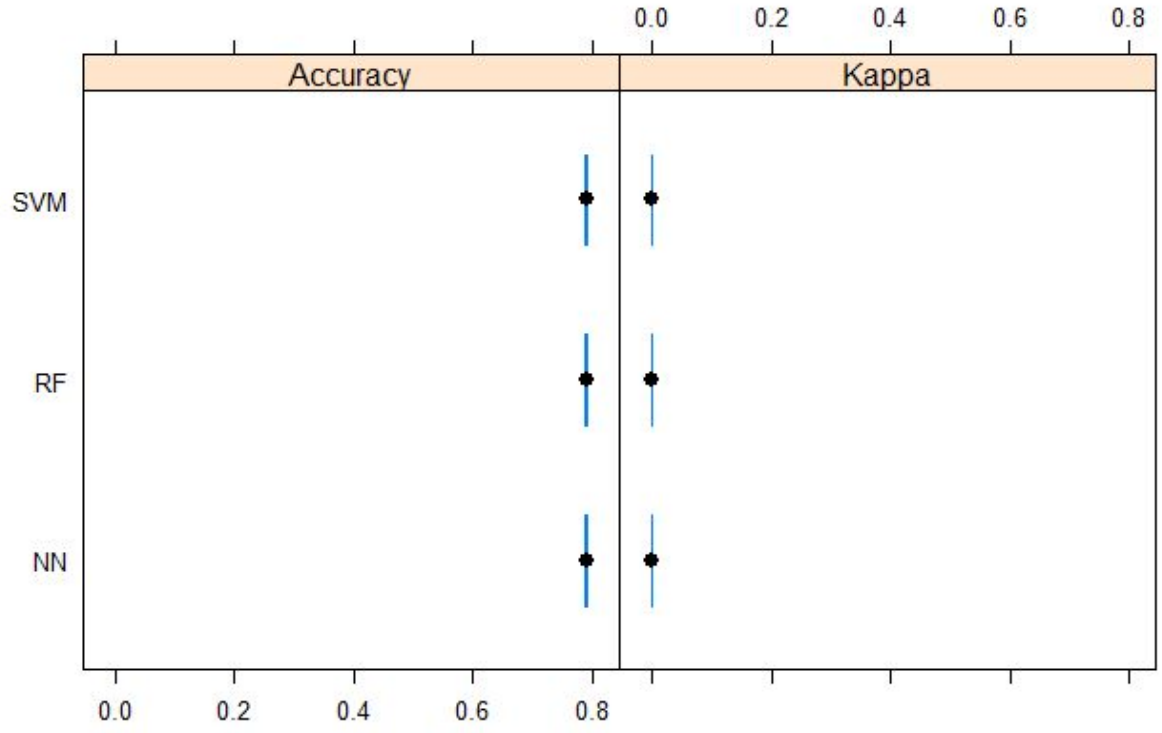
Question 4: Comparative Analysis

We compared three models: Support Vector Machine, Neural Networks and Random Forests using the caret library for three different CV methods: K-Fold, Bootstrap and Repeated K-Fold. Following are the results:

- **K-Fold with 10 folds**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.7883	0.7889	0.7896	0.7892	0.7896	0.7896
RF	0.7883	0.7886	0.7896	0.7892	0.7896	0.7899
SVM	0.7883	0.7886	0.7896	0.7892	0.7896	0.7899



All three models have almost similar accuracy values. But, out of the three, neural networks has the best accuracy.

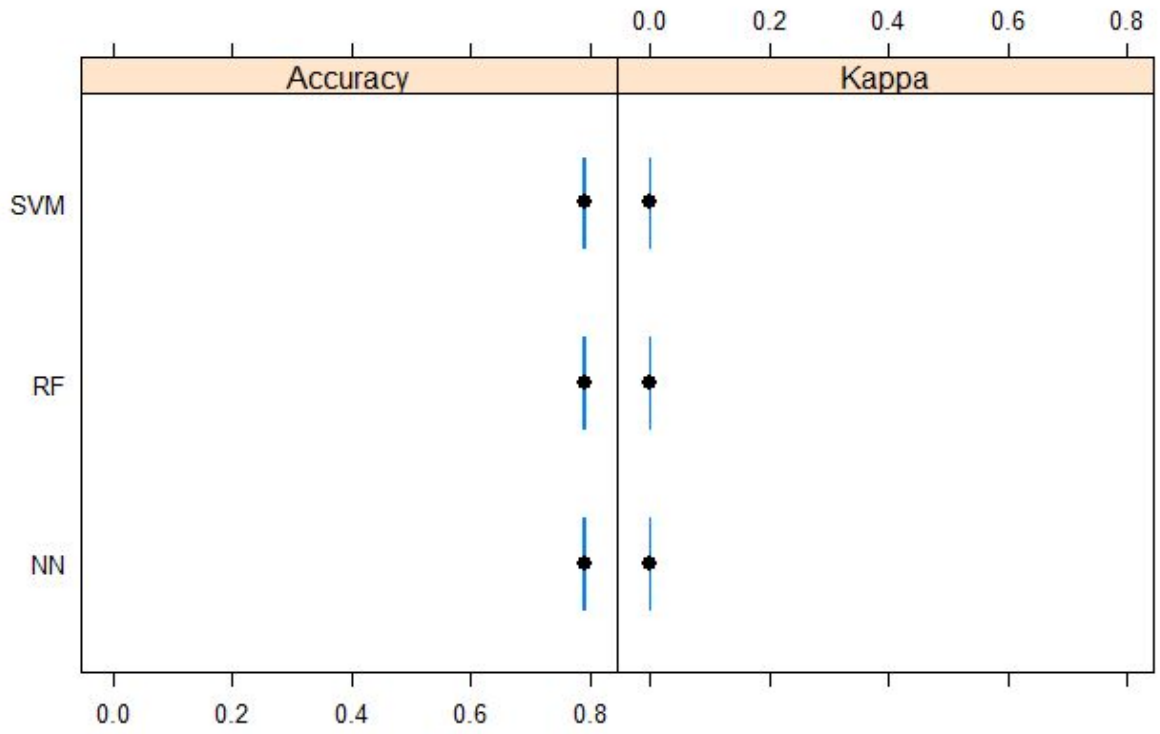
- **Repeated K-Fold with 10 folds and 3 repeats**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.7883	0.7883	0.7896	0.7892	0.7896	0.7899
RF	0.7883	0.7883	0.7896	0.7892	0.7896	0.7899
SVM	0.7864	0.7883	0.7896	0.7891	0.7896	0.7899

Kappa:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0	0	0	0	0	0
RF	0	0	0	0	0	0
SVM	-0.006274	0	0	-0.0002091	0	0

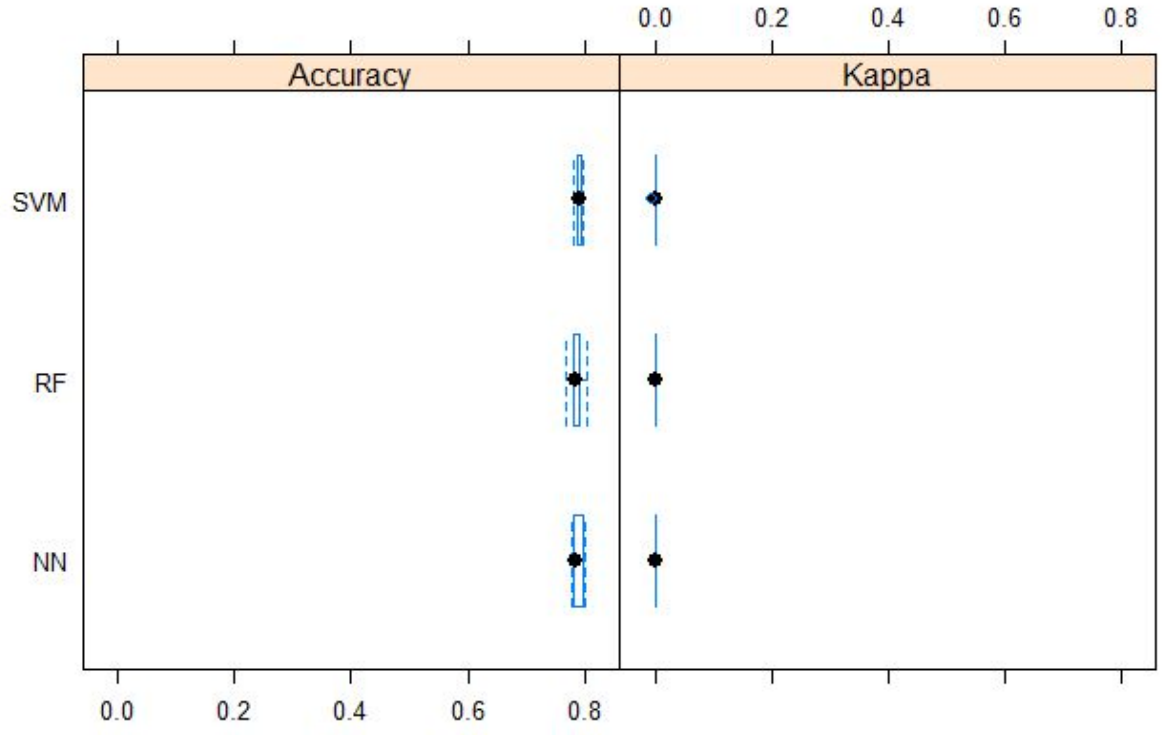


In this case, we can't say much in regards to the best methodology since the accuracy and kappa values for all three models are same.

- **Bootstrap**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.7769	0.7819	0.7859	0.7883	0.7961	0.8002
RF	0.7687	0.7821	0.7861	0.787	0.7899	0.8033
SVM	0.7818	0.7871	0.7911	0.7902	0.7932	0.7991



From the table and plot above, Support Vector Machine gives the most accurate results.

Research Question 3:
Predict the medium through which a complaint will be received

Predictors: Product, Company, Region(derived from the State), Timely.response.
Outcome: Submitted.via
Sampling rate: (Training:Testing)=75:25

Question 1: Support Vector Machine
To apply this technique in RStudio, we used the kernlab library. Moreover, we had to reduce the size of our dataset from around 600,000 records to 100,000 records so that RStudio could handle it, else we ran into memory allocation problems. We used two different kernels to build our svm classifier and tested them out. Following are the results:

- **Kernel: Linear**
  The 'vanilladot' kernel in ksvm was used to create the classifier.

Confusion Matrix:

Predicted/Actual	**Email**	**Fax**	**Phone**	**Postal mail**	**Referral**	**Web**
Email	0	0	0	0	0	0
Fax	0	0	0	0	0	0
Phone	0	0	0	0	0	0
Postal mail	0	0	0	0	0	0
Referral	0	0	0	0	0	0
Web	3	112	545	511	1451	5280

Classifier Measures:

Class-wise Measures	**Email**	**Fax**	**Phone**	**Postal mail**	**Referral**	**Web**

Recall	0	0	0	0	0	1
Specificity	1	1	1	1	1	0
Precision	NaN	NaN	NaN	NaN	NaN	0.6682
Accuracy	0.5	0.5	0.5	0.5	0.5	0.5
F1 Measure	NaN	NaN	NaN	NaN	NaN	0.801102985

Overall Accuracy: 0.6682

- **Kernel: Non-linear (Gaussian)**
  The 'rbfdot' kernel in ksvm was used to create the classifier. This classifier uses the "one versus one" approach.

Confusion Matrix:

Predicted/Actual	**Email**	**Fax**	**Phone**	**Postal mail**	**Referral**	**Web**
Email	0	0	0	0	0	0
Fax	0	0	0	0	0	0
Phone	0	0	0	0	0	0
Postal mail	0	0	0	0	0	0
Referral	0	0	9	1	26	26
Web	3	112	536	510	1425	5254

Classifier Measures:

Class-wise Measures	**Email**	**Fax**	**Phone**	**Postal mail**	**Referral**	**Web**

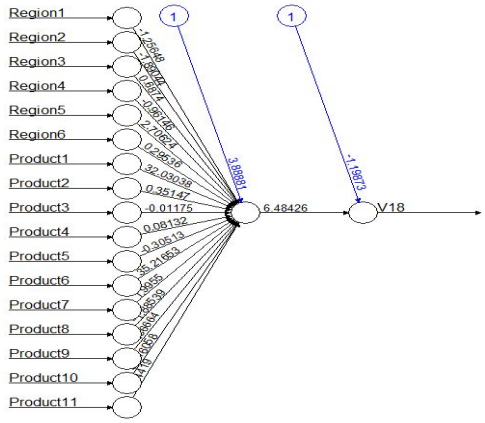
Recall	0	0	0	0	0.017919	0.99508
Specificity	1	1	1	1	0.994419	0.01373
Precision	NaN	NaN	NaN	NaN	0.419355	0.67015
Accuracy	0.5	0.5	0.5	0.5	0.506169	0.5044
F1 Measure	NaN	NaN	NaN	NaN	0.034369399	0.800913822

Overall Accuracy: 0.6682

Question 2: Neural Networks

Since all our variables (dependent and independent) are categorical, we could not apply neural networks directly. We used effect coding for all the dependent variables and numerical coding (one number for each class) for the independent variable. We had to reduce the size of our data set to 10,000, else RStudio we ran into memory allocation problems. We applied neural networks with 3 different number of hidden layers:

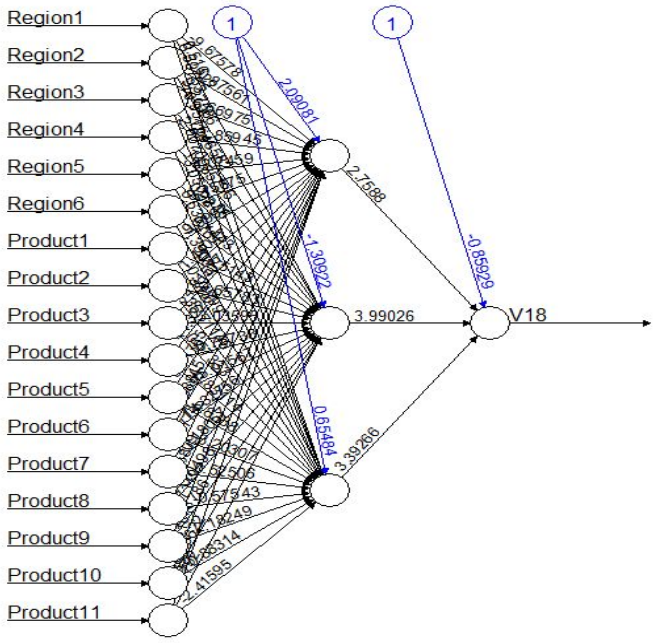
<u>Model 1:</u>
Dependent Variables: Region (6 levels), Product (11 levels)
Hidden: 1
Activation Function: Logistic



Correlation: -0.02

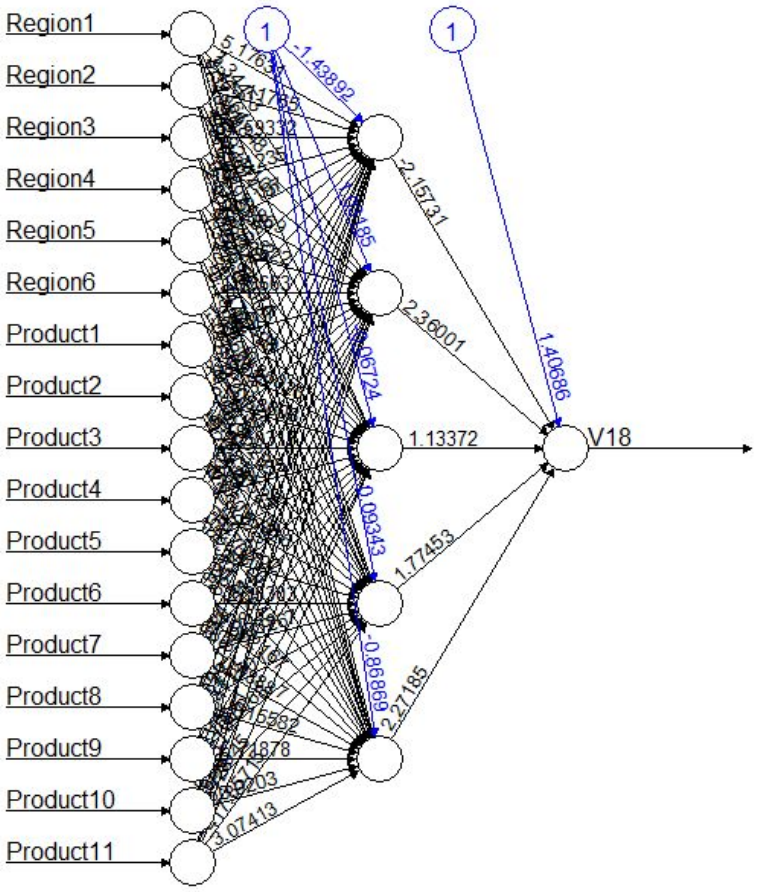
Model 2:
Dependent Variables: Region (6 levels), Product (11 levels)
Hidden: 3
Activation Function: Logistic

For the model with 3 hidden layers, RStudio was not able to converge the weights while training the model. We even tried to increase the converge time via the parameter 'stepmax', but weights were not being calculated with the given data set. So, we reduced the size of the data set to 1,000 in order to get the result. Following is a snapshot of the model:



Correlation: -0.00011

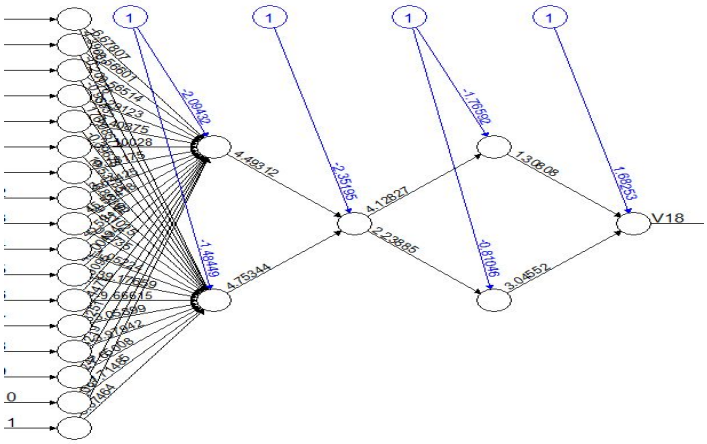
Model 3:
Dependent Variables: Region (6 levels), Product (11 levels)
Hidden: 5
Activation Function: Logistic



Correlation: 0.005

Model 4:

Dependent Variables: Region (6 levels), Product (11 levels)
Hidden: 3 layers (2,1,2)
Activation Function: Logistic



Correlation: 0.10

Question 3: Clustering

We applied different types of clustering as follows

Technique 1: Hierarchical Clustering

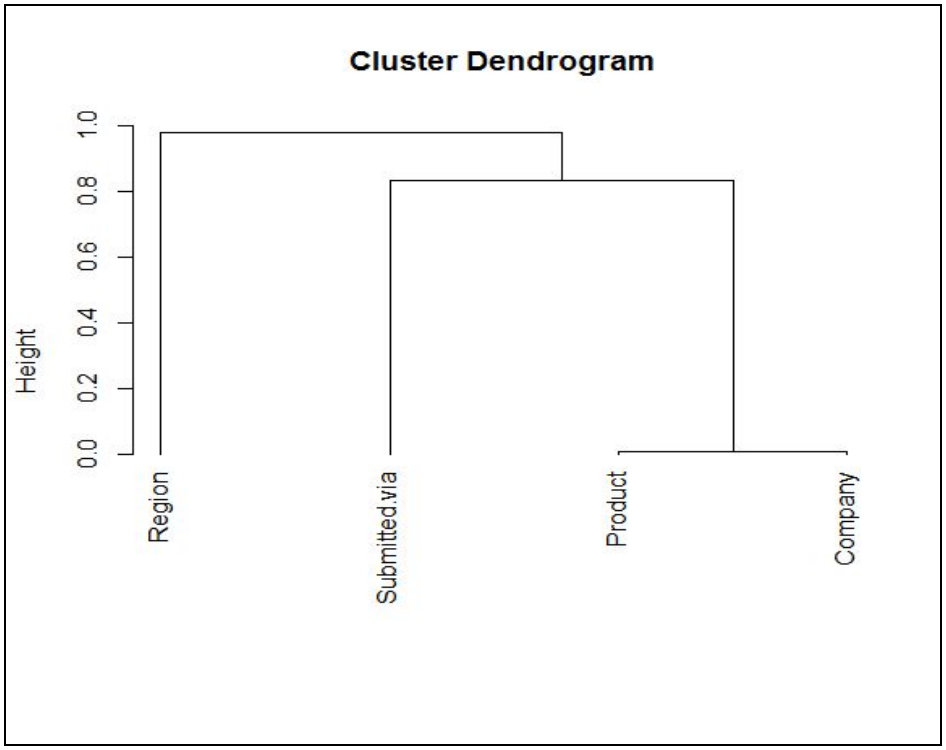
Since our entire dataset consists of categorical variables, using the normal 'hclust' method in R for performing hierarchical clustering is not possible. We found a package called **'ClustOfVar'** which provides methods specifically devoted to the clustering of variables with no restriction on the type (quantitative or qualitative) of the variables. The clustering methods developed in the package work with a mixture of quantitative and qualitative variables and also work for a set exclusively containing quantitative or qualitative variables. Also, missing data are allowed: they are replaced by means for quantitative variables and by zeros in the indicator matrix for qualitative variables.

Variables used for generating the dendrogram:

- Region (4 levels)
- Product (12 levels)
- Company (To reduce the number of levels, we took only the top 10 companies which account for more than 50% of the entire dataset)
- Submitted.via (6 levels)

We used a sample of approximately 300000 records for this test.

We used the **'hclustvar'** method in this package to generate a hierarchical tree structure. The dendrogram generated is as follows




```
Call:
cutreevar(obj = tree, k = 2, matsim = TRUE)



Data:
   number of observations:  314739
   number of variables:  4
   number of clusters:  2

Cluster  1 :
             squared loading
Product                 0.94
Company                 0.94
Submitted.via           0.28


Cluster  2 :
       squared loading
Region               1


Gain in cohesion (in %):  53.76
```

Similarity Matrix

```
$cluster1
                Product   Company Submitted.via
Product       1.0000000 0.9808154     0.1073043
Company       0.9808154 1.0000000     0.1057351
Submitted.via 0.1073043 0.1057351     1.0000000

$cluster2
       Region
Region      1
```

The similarity matrix shows the similarities between variables for each cluster. For qualitative variables, the similarity between two variables is defined as the square of the canonical relation between two sets of dummy variables. In this case, 'Product','Company' and 'Submitted.via' are grouped in Cluster 1, whereas only 'Region' is in Cluster 2.

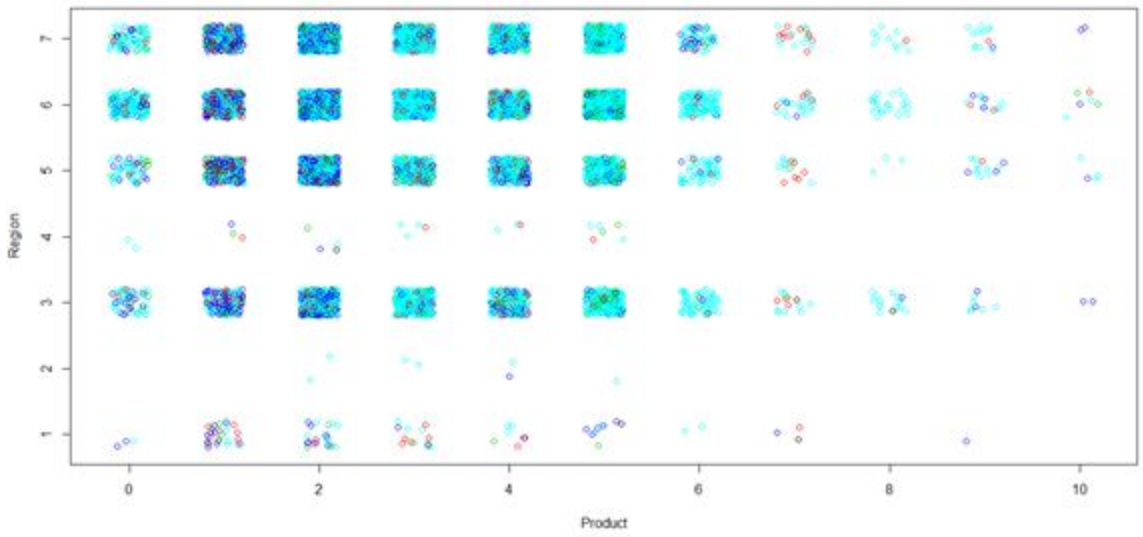
Technique 2: K Modes Clustering

To apply this technique, we have to make use of klaR library. Since our data has only categorical variables, we cannot use k-means and k-medoids as it is, nor we can use a distance formula based clustering method, like Euclidean or Gower distance, we have to make use of k-modes package. For ease of use, we have mapped the textual data to numerical factors. We reduced the data to 10,000 rows and ran the code for 3-4 iterations, as more data resulted in memory allocation issues in R Studio.

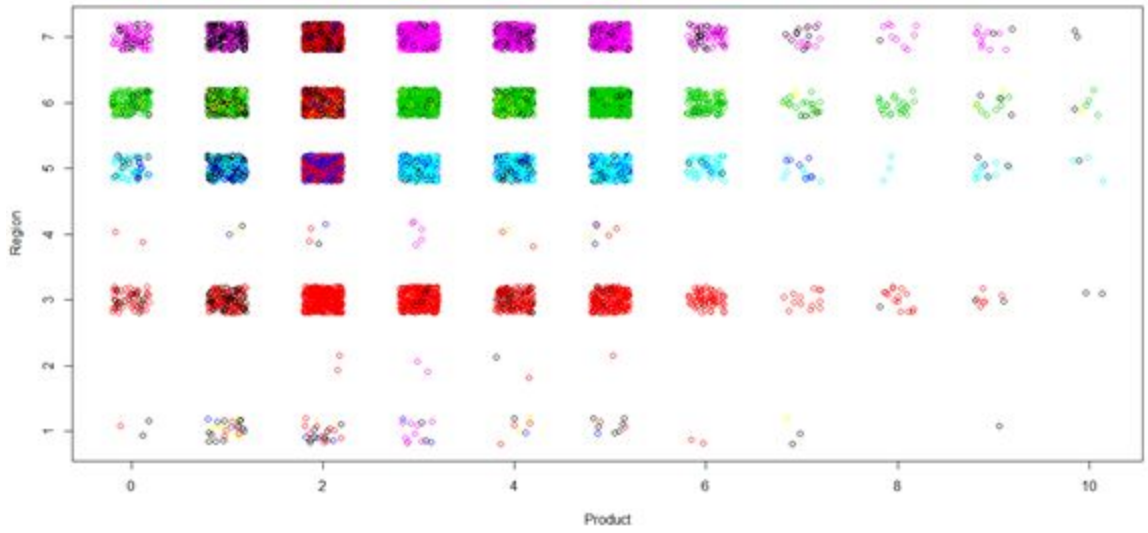
Columns used: Product, Region, Submitted.via

```
> cl
K-modes clustering with 7 clusters of sizes 1737, 2941, 2326, 648, 810, 1260, 278

Cluster modes:
  Product Region Submitted.via
1       2      5             4
2       2      1             5
3       5      4             5
4       2      3             3
5       5      3             5
6       3      5             5
7       2      4             2

> cl$withindiff
[1] 2356 2880 1793  757  591  903  205
```

Scatter plot of Product vs Region, colored on basis of Submitted.via



Scatter plot of Product vs Region, colored on basis of clusters



Technique 3: Mixture Model Clustering

```
************************************************************
* Number of samples    =  314739
* Problem dimension    =  4
************************************************************
*       Number of cluster =  6
*             Model Type =  Binary_pk_Ekjh
*              Criterion =  BIC(3170328.0264)
*             Parameters =  list by cluster
*                 Cluster  1 :
                        Proportion =  0.3739
                            Center =  4.0000 4.0000 3.0000 6.0000
                           Scatter = |     0.0003     0.0008     0.0001     0.0034     0.0014     0.0000     0.0005     0.0001     0.0000     0.0000     0.0003
    0.0000 |
                                     |     0.0009     0.0028     0.0020     0.6405     0.3416     0.0001     0.0003     0.2917     0.0012 |
                                     |     0.1416     0.1609     0.5355     0.2330 |
                                     |     0.0002     0.0151     0.0185     0.1351     0.0695     0.2383 |
*                 Cluster  2 :
                        Proportion =  0.0809
                            Center =  1.0000 1.0000 3.0000 6.0000
                           Scatter = |     0.3586     0.0071     0.2566     0.0000     0.0092     0.0106     0.0705     0.0030     0.0002     0.0013     0.0000
    0.0000 |
                                     |     0.3224     0.1169     0.2042     0.0000     0.0001     0.0000     0.0008     0.0000     0.0004 |
                                     |     0.0985     0.2555     0.6296     0.2756 |
                                     |     0.0003     0.0144     0.1190     0.0432     0.2961     0.4730 |
*                 Cluster  3 :
                        Proportion =  0.1263
                            Center =  3.0000 3.0000 3.0000 6.0000
                           Scatter = |     0.0222     0.0547     0.2437     0.0001     0.1298     0.0007     0.0127     0.0006     0.0007     0.0053     0.0170
    0.0000 |
                                     |     0.1235     0.3659     0.5474     0.0012     0.0013     0.0000     0.0000     0.0005     0.0551 |
                                     |     0.1696     0.2336     0.6406     0.2374 |
                                     |     0.0004     0.0094     0.0738     0.0641     0.1366     0.2843 |
*                 Cluster  4 :
                        Proportion =  0.1129
                            Center =  7.0000 1.0000 3.0000 5.0000
                           Scatter = |     0.0042     0.0002     0.0000     0.0006     0.0001     0.0000     0.0051     0.0000     0.0000     0.0000     0.0000
    0.0000 |
                                     |     0.5514     0.0058     0.0694     0.0000     0.0000     0.0575     0.1233     0.0000     0.2954 |
                                     |     0.1150     0.2251     0.6472     0.3070 |
                                     |     0.0019     0.0061     0.0567     0.0431     0.2965     0.1887 |
*                 Cluster  5 :
                        Proportion =  0.0629
                            Center =  1.0000 9.0000 3.0000 6.0000
                           Scatter = |     0.3587     0.1153     0.1050     0.0001     0.0176     0.0120     0.0474     0.0023     0.0009     0.0007     0.0573
    0.0000 |
                                     |     0.0828     0.0002     0.0001     0.0000     0.0002     0.0000     0.0001     0.0000     0.0835 |
                                     |     0.0962     0.1486     0.5795     0.3348 |
                                     |     0.0004     0.0133     0.1682     0.0498     0.2803     0.5120 |
*                 Cluster  6 :
                        Proportion =  0.2431
                            Center =  7.0000 1.0000 3.0000 6.0000
                           Scatter = |     0.0015     0.0014     0.0000     0.0000     0.0293     0.0001     0.0325     0.0001     0.0001     0.0000     0.0000
    0.0000 |
                                     |     0.7165     0.0044     0.0700     0.0002     0.0002     0.1711     0.2407     0.0001     0.2298 |
                                     |     0.1453     0.1689     0.6146     0.3004 |
                                     |     0.0001     0.0285     0.0585     0.0662     0.1245     0.2779 |
*          Log-likelihood =  -1584106.9450
************************************************************
```

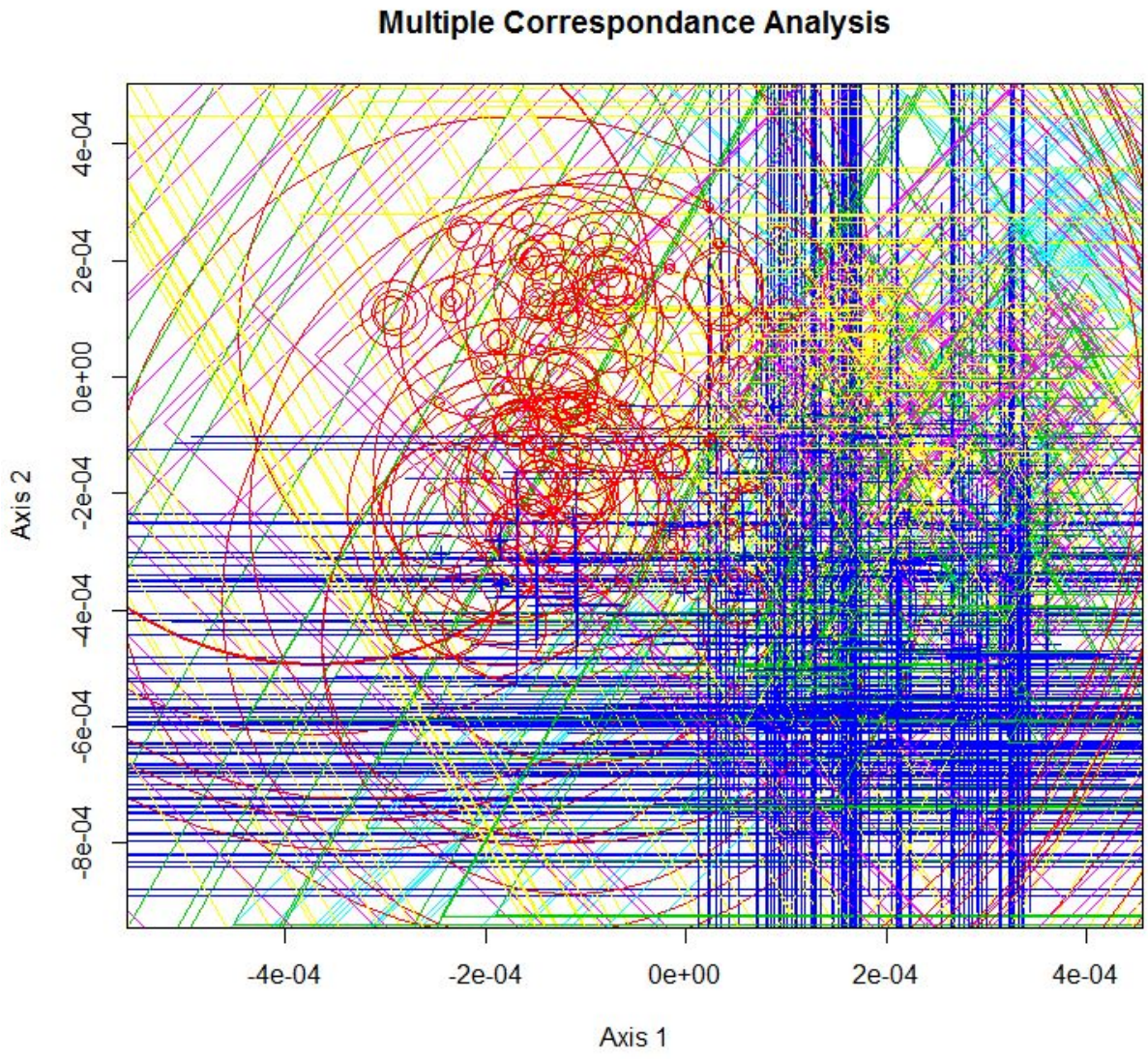
Proportion of records in Cluster 1: 0.3739
Proportion of records in Cluster 2: 0.0809
Proportion of records in Cluster 3: 0.1263
Proportion of records in Cluster 4: 0.1129
Proportion of records in Cluster 5: 0.0629
Proportion of records in Cluster 6: 0.2431

Plot of the result

The plot() function gives the following output.

A multiple correspondence analysis is performed to get a 2-dimensional representation of the dataset and a bigger symbol is used when observations are similar.

Each shape/color represents a cluster



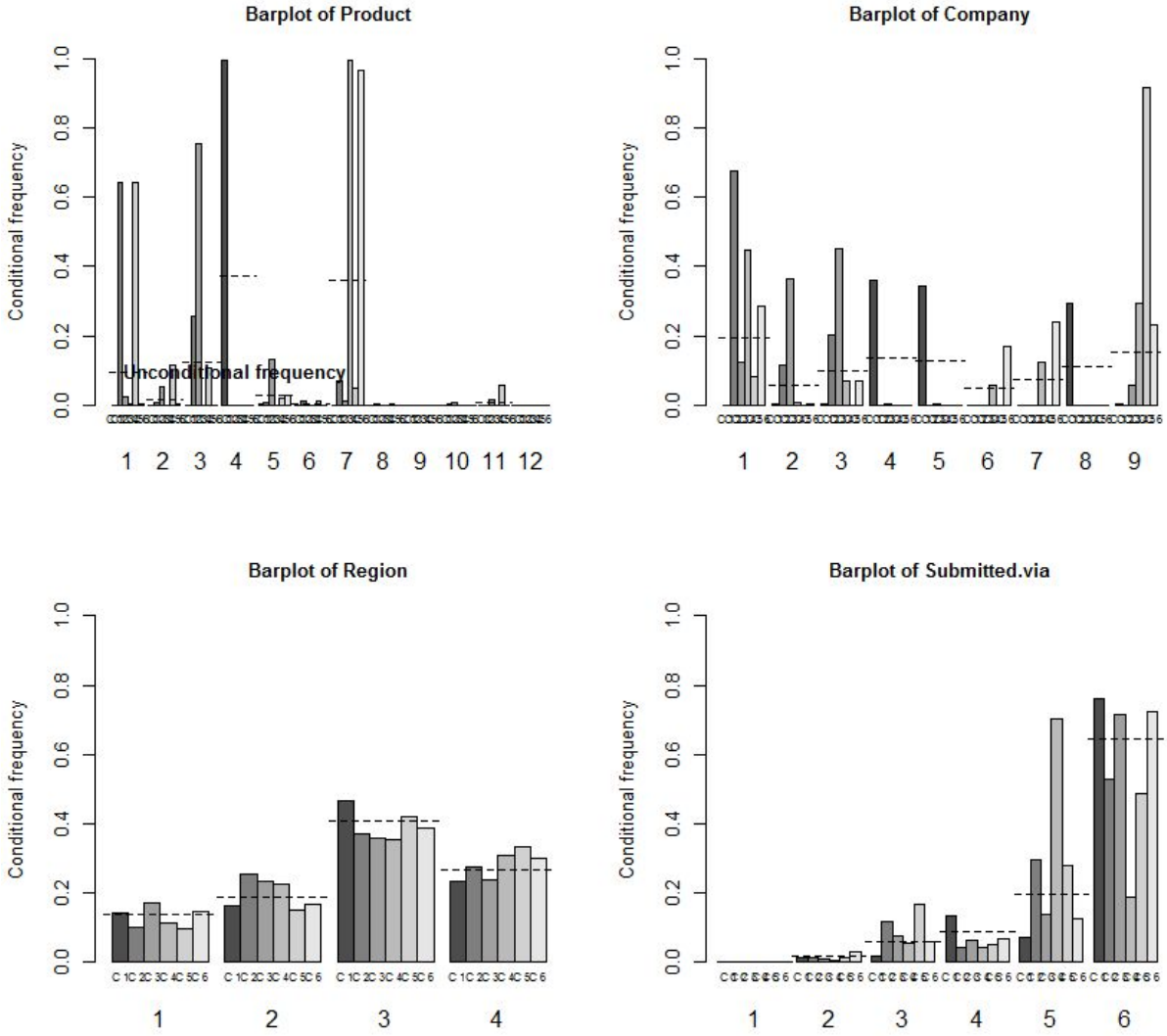
Barplot of the result

The barplot() function gives the following output.

For each qualitative variable, we obtain:

• a barplot with the frequencies of the modalities;

• for each cluster a barplot with the probabilities for each modality to be in that cluster.



Question 4: Comparative Analysis

We compared three models: Support Vector Machine, Neural Networks and Random Forests using the caret library for three different CV methods: K-Fold, Bootstrap and Repeated K-Fold. Following are the results:

- **K-Fold with 10 folds**

Accuracy:

Model	**Minimum**	**1st Quantile**	**Median**	**Mean**	**3rd Quantile**	**Maximum**

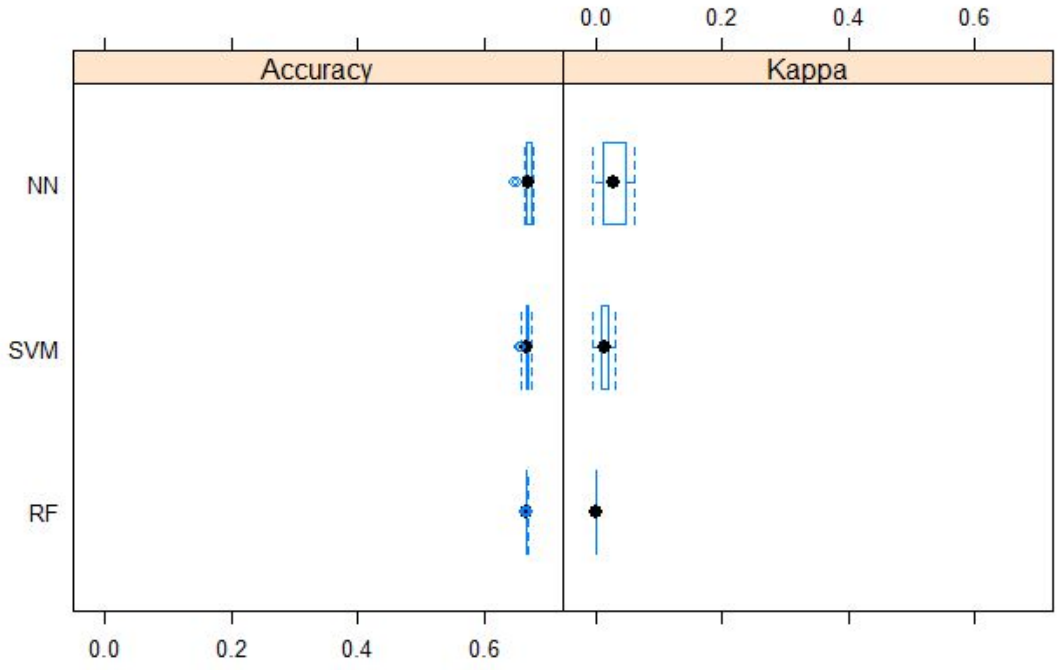
NN	0.6572	0.6663	0.6696	0.6693	0.6742	0.6777
RF	0.6667	0.6677	0.6677	0.6679	0.6682	0.6693
SVM	0.6635	0.6661	0.6682	0.6679	0.6692	0.6714

From the above results, we can say that Neural networks is the best model in terms of accuracy.

- **Repeated K-Fold with 10 folds and 3 repeats**

Accuracy:

Model	**Minimum**	**1st Quantile**	**Median**	**Mean**	**3rd Quantile**	**Maximum**
NN	0.6483	0.6667	0.6688	0.6693	0.6744	0.6772
RF	0.6646	0.6677	0.6682	0.6679	0.6682	0.6693
SVM	0.6566	0.6663	0.6682	0.6676	0.6704	0.6741

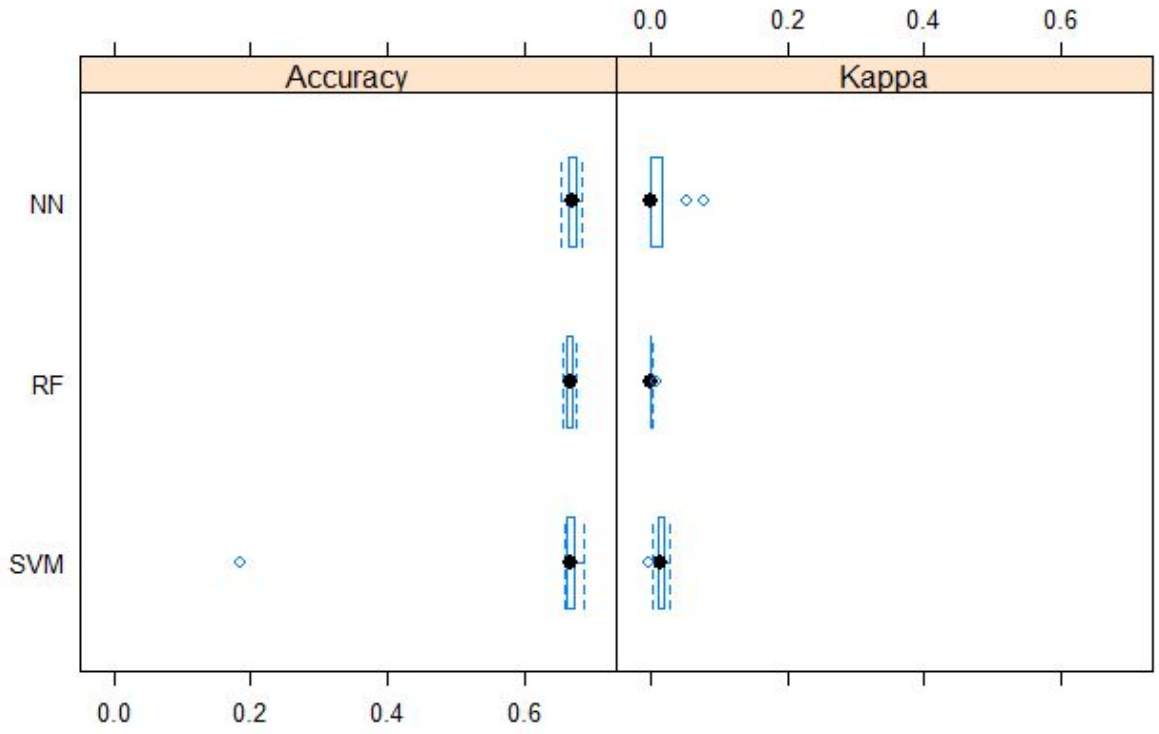


From the above table and plot, Neural network seems to be the best classification method in our case.

- **Bootstrap**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.6556	0.6657	0.671	0.6704	0.6765	0.6843
RF	0.6571	0.6647	0.6687	0.6683	0.6715	0.6776
SVM	0.1859	0.6639	0.6685	0.622	0.6732	0.6872



From the plot and table, it is clear that Neural networks is the best methodology for this question.

Research Question 4:
Predict the geographical region in United States where the complaint originated

Predictors: Product, Submitted.via, Timely.response., Consumer.disputed., Company.response.to.consumer, Company
Outcome: Region (4 levels)
Sampling rate: (Training:Testing)=75:25

Question 1: Support Vector Machine
To apply this technique in RStudio, we used the kernlab library. Moreover, we had to reduce the size of our dataset from around 600,000 records to 100,000 records so that RStudio could handle it, else we ran into memory allocation problems. We used two different kernels to build our svm classifier and tested them out. Following are the results:

- **Kernel: Linear**
  The 'vanilladot' kernel in ksvm was used to create the classifier.

Confusion Matrix:

Predicted/Actual	Midwest	North East	South	West
Midwest	0	0	0	0
North East	0	0	0	0
South	2444	3005	6334	4023
West	0	0	0	0

Classifier Measures:

Class-wise Measures	Midwest	North East	South	West
Recall	0	0	1	0
Specificity	1	1	0	1

Precision	NaN	NaN	0.4007	NaN
Accuracy	0.5	0.5	0.5	0.5
F1 Measure	NaN	NaN	0.5721425	NaN

Overall Accuracy: 0.4007

- **Kernel: Non-linear (Gaussian)**
  The 'rbfdot' kernel in ksvm was used to create the classifier. This classifier uses the "one versus one" approach.

Confusion Matrix:

Predicted/Actual	**Midwest**	**North East**	**South**	**West**
Midwest	0	0	0	0
North East	12	40	38	20
South	2423	2962	6279	3981
West	9	3	17	22

Classifier Measures:

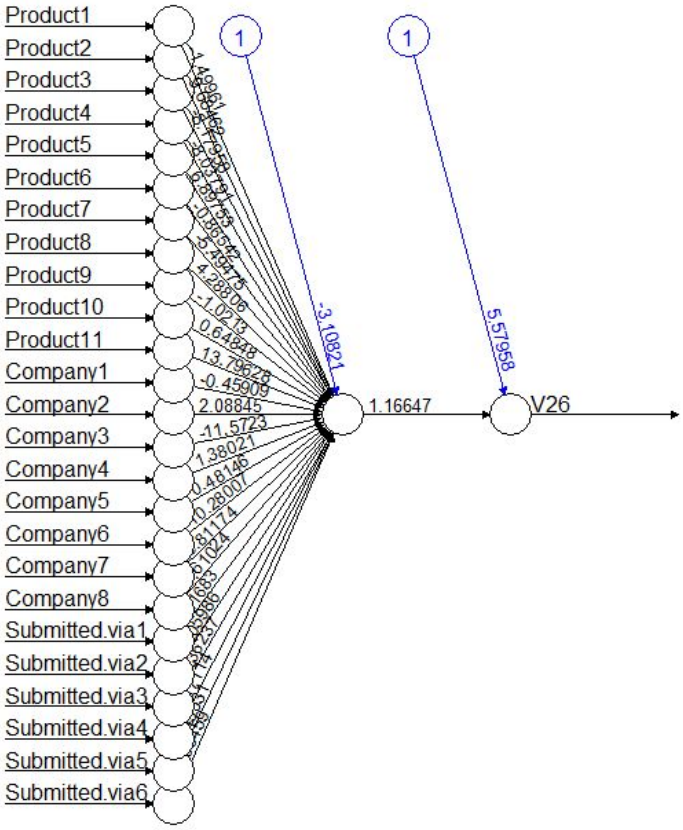
Class-wise Measures	**Midwest**	**North East**	**South**	**West**
Recall	0	0.013311	0.99132	0.005469
Specificity	1	0.994532	0.01119	0.997539
Precision	NaN	0.363636	0.40134	0.431373
Accuracy	0.5	0.503921	0.50125	0.501504
F1 Measure	NaN	0.025681906	0.571361809	0.010801063

Overall Accuracy: 0.4012

Question 2: Neural Networks

Since all our variables (dependent and independent) are categorical, we could not apply neural networks directly. We used effect coding for all the dependent variables and numerical coding (one number for each class) for the independent variable. We had to reduce the size of our data set to 1,000, else RStudio we ran into memory allocation problems. We applied neural networks with 3 different number of hidden layers:

Model 1:
Dependent Variables: Product (11 levels), Company(8 levels), Submitted.via(6 levels)
Hidden: 1
Activation Function: Logistic



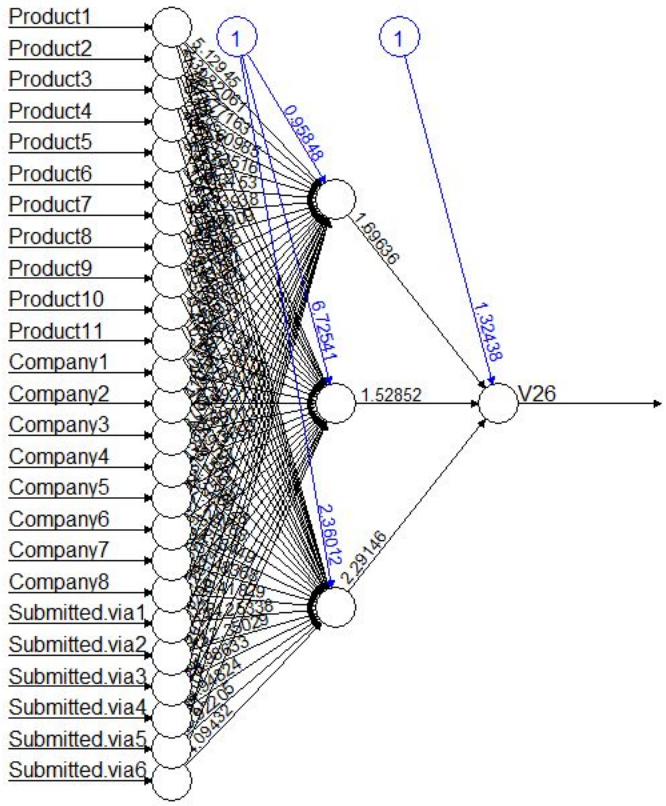
Correlation: 0.33

Model 2:

Dependent Variables: Product (11 levels), Company(8 levels), Submitted.via(6 levels)

Hidden: 3

Activation Function: Logistic



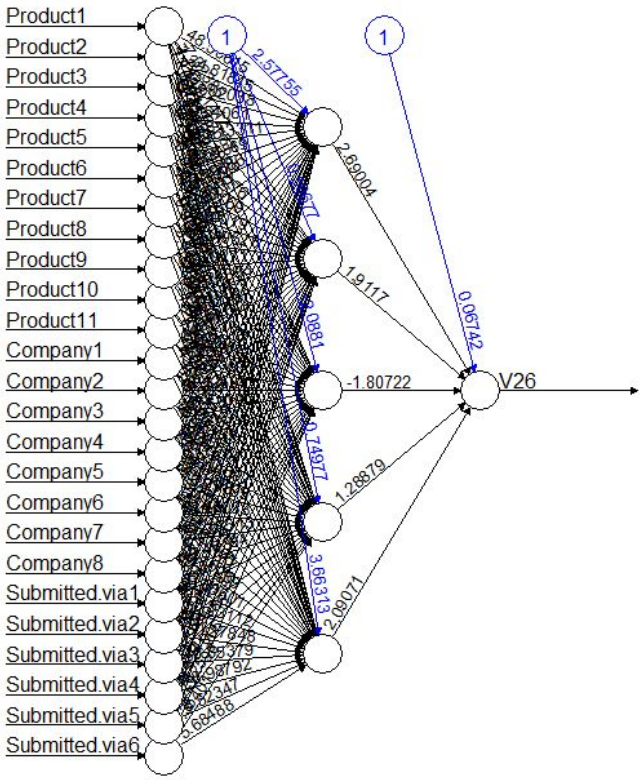
Correlation: 0.097

Model 3:

Dependent Variables: Product (11 levels), Company(8 levels), Submitted.via(6 levels)

Hidden: 5

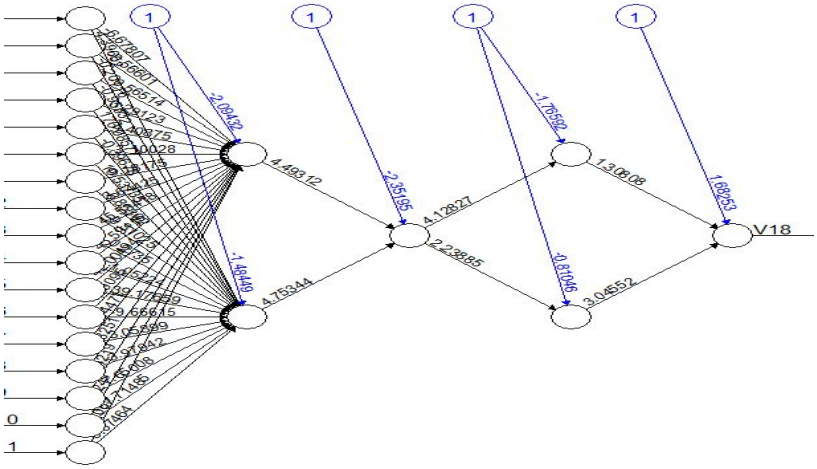
Activation Function: Logistic



Correlation: 0.086

Model 4:

Dependent Variables: Product (11 levels), Company(8 levels), Submitted.via(6 levels)
Hidden: 3 layers (2,1,2)
Activation Function: Logistic



Correlation: 0.0821

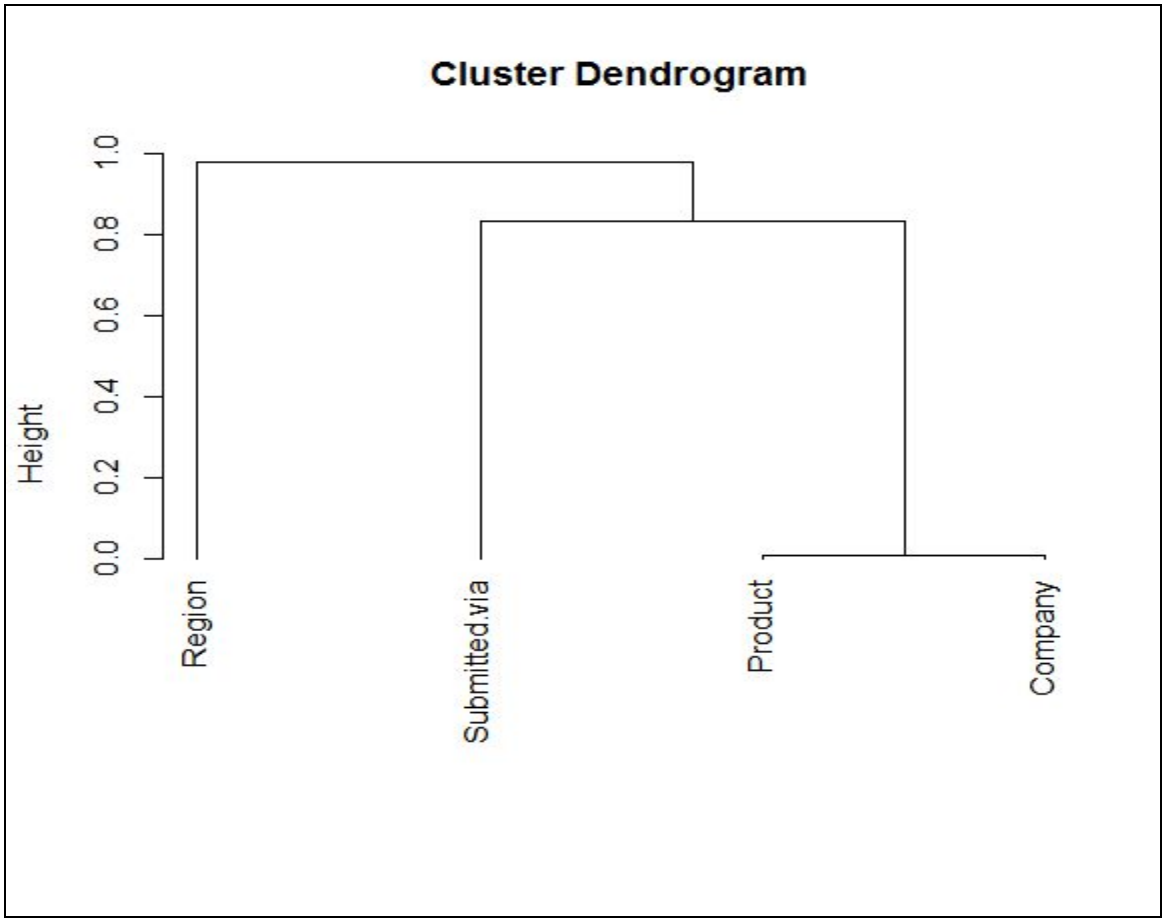
Question 3: Clustering
We applied different types of clustering as follows

Technique 1: Hierarchical Clustering
Since our entire dataset consists of categorical variables, using the normal 'hclust' method in R for performing hierarchical clustering is not possible. We found a package called **'ClustOfVar'** which provides methods specifically devoted to the clustering of variables with no restriction on the type (quantitative or qualitative) of the variables. The clustering methods developed in the package work with a mixture of quantitative and qualitative variables and also work for a set exclusively containing quantitative or qualitative variables. Also, missing data are allowed: they are replaced by means for quantitative variables and by zeros in the indicator matrix for qualitative variables.

Variables used for generating the dendrogram:

- Region (4 levels)
- Product (12 levels)
- Company (To reduce the number of levels, we took only the top 10 companies which account for more than 50% of the entire dataset)
- Submitted.via (6 levels)

We used a sample of approximately 300000 records for this test.
We used the **'hclustvar'** method in this package to generate a hierarchical tree structure. The dendrogram generated is as follows




```
Call:
cutreevar(obj = tree, k = 2, matsim = TRUE)



Data:
   number of observations:  314739
   number of variables:  4
   number of clusters:  2

Cluster  1 :
             squared loading
Product                 0.94
Company                 0.94
Submitted.via           0.28


Cluster  2 :
       squared loading
Region               1


Gain in cohesion (in %):  53.76
```

Similarity Matrix

```
$cluster1
              Product   Company Submitted.via
Product       1.0000000 0.9808154     0.1073043
Company       0.9808154 1.0000000     0.1057351
Submitted.via 0.1073043 0.1057351     1.0000000

$cluster2
       Region
Region      1
```

The similarity matrix shows the similarities between variables for each cluster. For qualitative variables, the similarity between two variables is defined as the square of the canonical relation between two sets of dummy variables.

In this case, 'Product','Company' and 'Submitted.via' are grouped in Cluster 1, whereas only 'Region' is in Cluster 2.

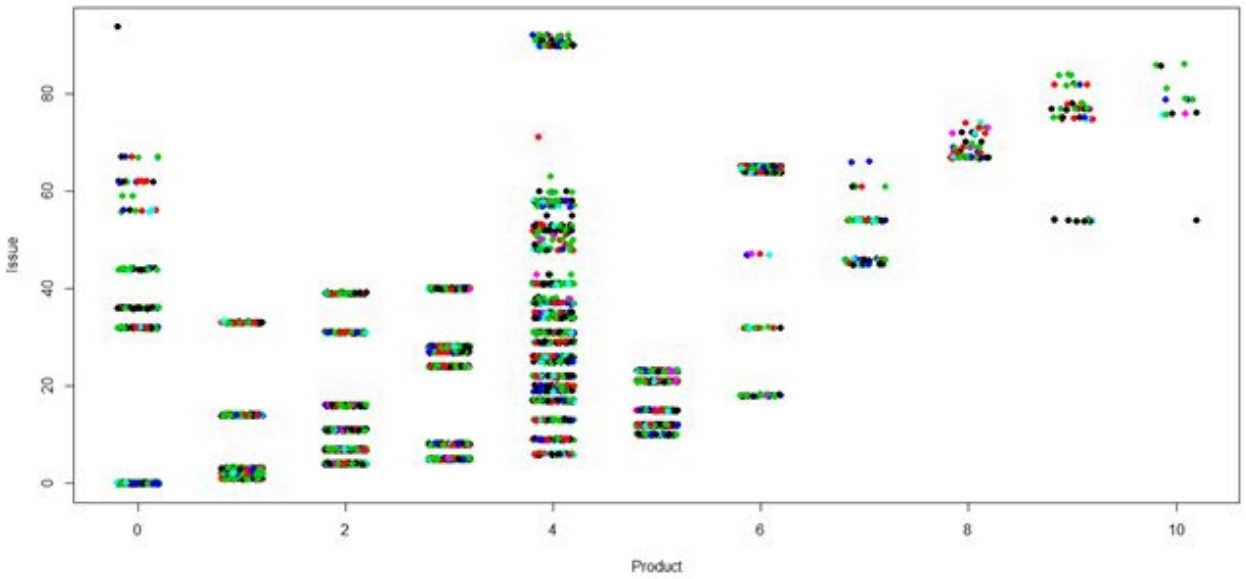
Technique 2: modes Clustering

To apply this technique, we have to make use of klaR library. Since our data has only categorical variables, we cannot use k-means and k-medoids as it is, nor we can use a distance formula based clustering method. We reduced the data to 10,000 rows and ran the code for 3-4 iterations, as more data resulted in memory allocation issues in R Studio.

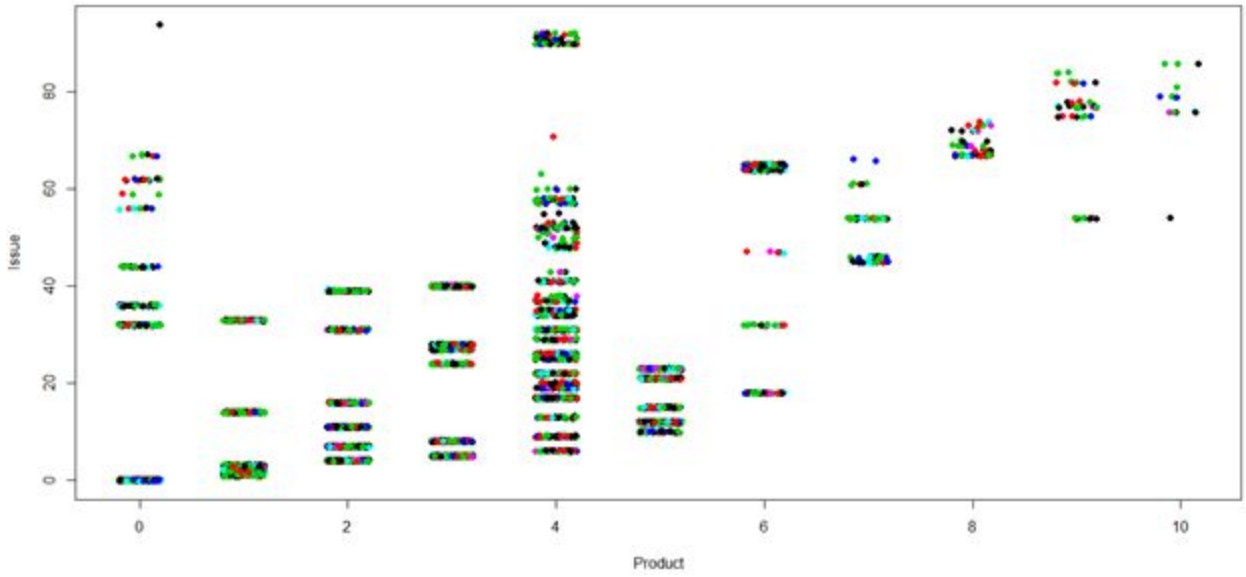
Columns used: Product, Issue, Region

```
> cl
K-modes clustering with 6 clusters of sizes 2491, 1780, 3359, 1043, 885, 357

Cluster modes:
  Product Issue Region
1       2     7      5
2       5    12      3
3       2     7      4
4       2     7      3
5       3     5      1
6       4    26      3

> cl$withindiff
[1] 3633 2276 4912  971 1313  342
```

Scatter plot of Product vs Issue, colored on the basis of Region



Scatter plot of Product vs Issue, colored on the basis of clusters



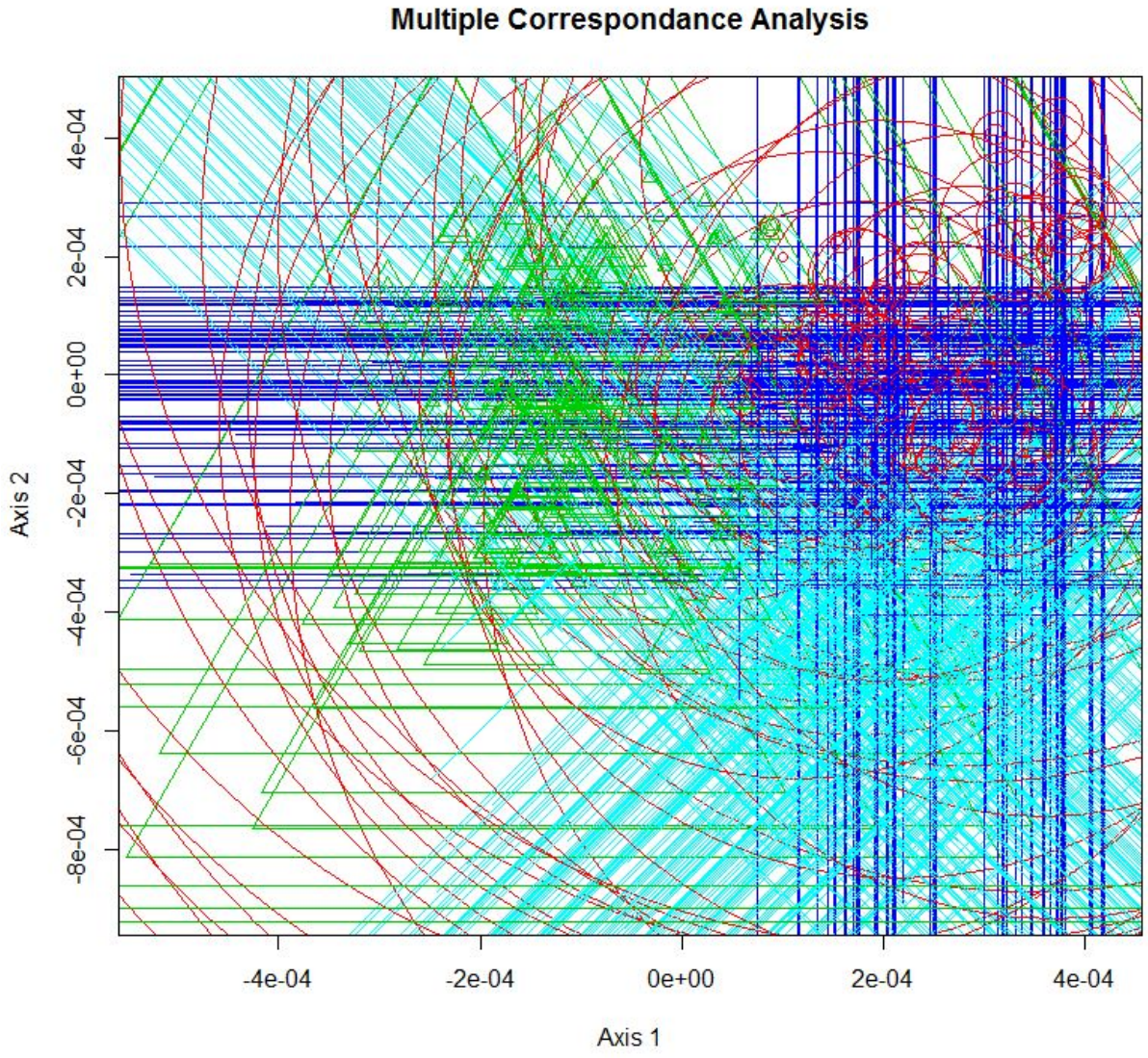
Technique 3: Mixture Model Clustering

```
************************************************************
* Number of samples    =  314739
* Problem dimension    =  4
************************************************************
*       Number of cluster =  4
*              Model Type =  Binary_pk_Ekjh
*               Criterion =  BIC(3175342.0631)
*              Parameters =  list by cluster
*                  Cluster  1 :
                         Proportion =  0.2730
                             Center =  7.0000 1.0000 3.0000 6.0000
                            Scatter = |     0.0017     0.0009     0.0000     0.0001     0.0223     0.0001     0.0252     0.0001     0.0000     0.0000     0.0000
    0.0000 |
                                      |     0.6975     0.0019     0.0808     0.0001     0.0000     0.1761     0.2649     0.0000     0.1736 |
                                      |     0.1478     0.1846     0.6298     0.2975 |
                                      |     0.0007     0.0236     0.0482     0.0628     0.2743     0.4096 |
*                  Cluster  2 :
                         Proportion =  0.3740
                             Center =  4.0000 4.0000 3.0000 6.0000
                            Scatter = |     0.0002     0.0008     0.0002     0.0038     0.0014     0.0000     0.0006     0.0001     0.0000     0.0000     0.0003
    0.0000 |
                                      |     0.0008     0.0028     0.0020     0.6405     0.3416     0.0001     0.0003     0.2917     0.0011 |
                                      |     0.1416     0.1609     0.5355     0.2330 |
                                      |     0.0002     0.0151     0.0185     0.1351     0.0695     0.2383 |
*                  Cluster  3 :
                         Proportion =  0.1940
                             Center =  7.0000 9.0000 3.0000 6.0000
                            Scatter = |     0.3979     0.0429     0.0574     0.0005     0.0081     0.0071     0.5357     0.0016     0.0004     0.0003     0.0197
    0.0000 |
                                      |     0.4344     0.0221     0.0446     0.0000     0.0001     0.0000     0.0008     0.0000     0.5020 |
                                      |     0.0940     0.1890     0.6017     0.3187 |
                                      |     0.0005     0.0142     0.1249     0.0454     0.3596     0.5447 |
*                  Cluster  4 :
                         Proportion =  0.1589
                             Center =  3.0000 3.0000 3.0000 6.0000
                            Scatter = |     0.1148     0.0409     0.2972     0.0000     0.1114     0.0022     0.0094     0.0010     0.0006     0.0048     0.0121
    0.0000 |
                                      |     0.1784     0.3311     0.5734     0.0008     0.0010     0.0000     0.0001     0.0003     0.0618 |
                                      |     0.1583     0.2449     0.6418     0.2386 |
                                      |     0.0004     0.0102     0.0801     0.0611     0.1502     0.3020 |
*          Log-likelihood =  -1586968.4293
************************************************************
```

Proportion of records in Cluster1 = 0.2730
Proportion of records in Cluster2 = 0.3740
Proportion of records in Cluster3 = 0.1940
Proportion of records in Cluster4 = 0.1589

Plot of the Result

The plot() function gives the following output.
A multiple correspondence analysis is performed to get a 2-dimensional representation of the dataset and a bigger symbol is used when observations are similar.



Each shape/color represent a cluster.

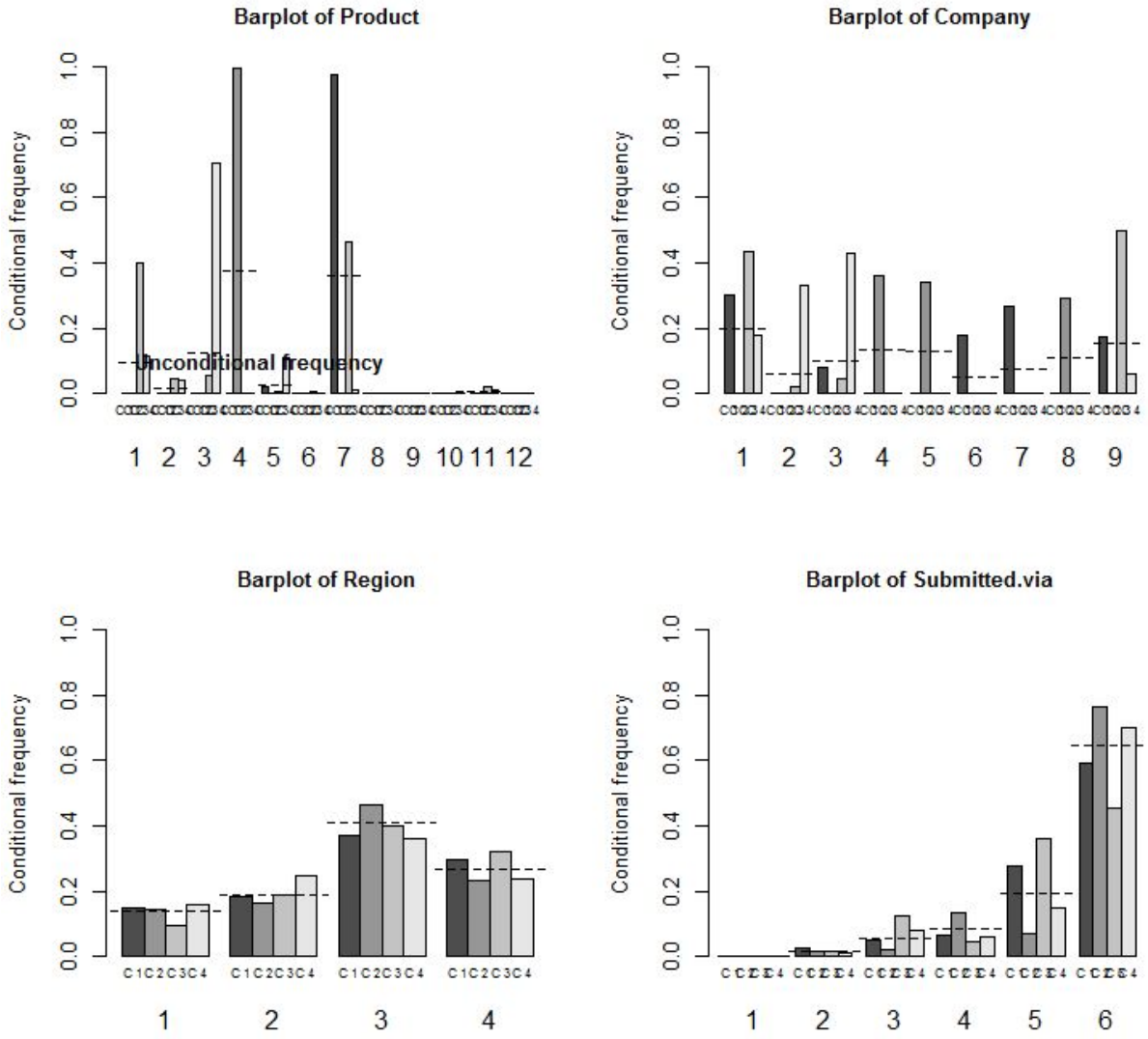
Barplot of the result

The barplot() function gives the following output.

For each qualitative variable, we obtain:

• a barplot with the frequencies of the modalities;

• for each cluster a barplot with the probabilities for each modality to be in that cluster.



For example, it tells us that a record having Region 1(say Midwest) has the probability of approx. 0.13 of belonging to cluster C1
OR
A complaint sent via medium 6(say Web) has the probability of approx. 0.8 of belonging to cluster C2

Question 4: Comparative Analysis
We compared three models: Support Vector Machine, Neural Networks and Random Forests using the caret library for three different CV methods: K-Fold, Bootstrap and Repeated K-Fold. Following are the results:

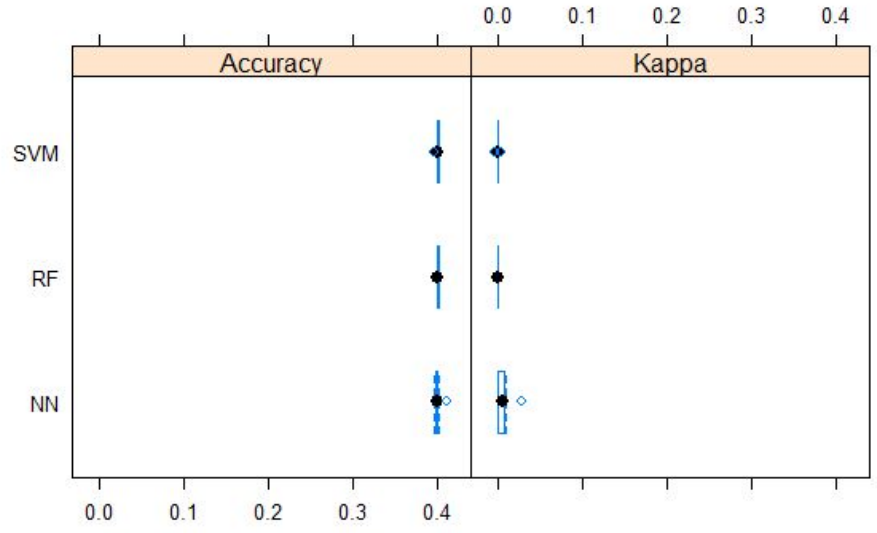
- **K-Fold with 10 folds**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.3965245	0.3987342	0.399684	0.4004771	0.4005523	0.4104596
RF	0.399684	0.3998421	0.4003165	0.4006327	0.4012638	0.4018987
SVM	0.3965245	0.4003165	0.4003165	0.4003165	0.4012638	0.4018987

Kappa:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0	0	0.004567353	0.005955571	0.006755079	0.02833465
RF	0	0	0	0	0	0
SVM	-0.003323582	0	0	-9.84875E-06	0	0.003225095

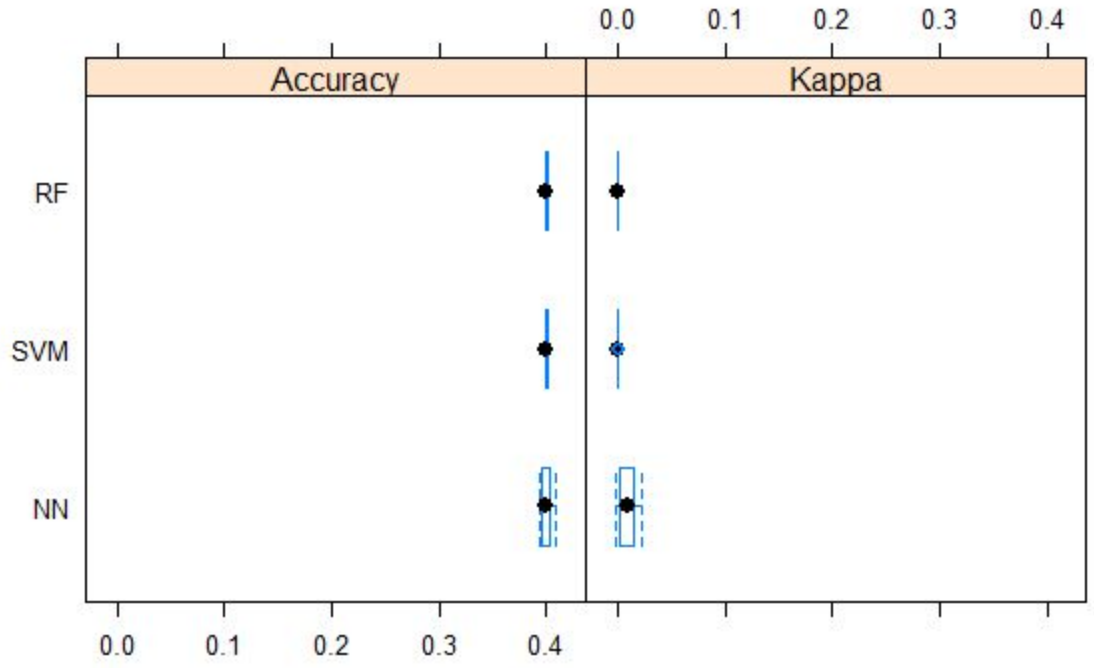


From the accuracy and kappa values, we observe that both Random Forests and Support Vector Machine have same accuracy.

- **Repeated K-Fold with 10 folds and 3 repeats**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.3933649	0.3959734	0.399684	0.4000545	0.4033217	0.4082278
RF	0.399684	0.4003165	0.4004737	0.4006326	0.4012638	0.4018987
SVM	0.399684	0.4003165	0.4003165	0.4005797	0.4012638	0.4012638

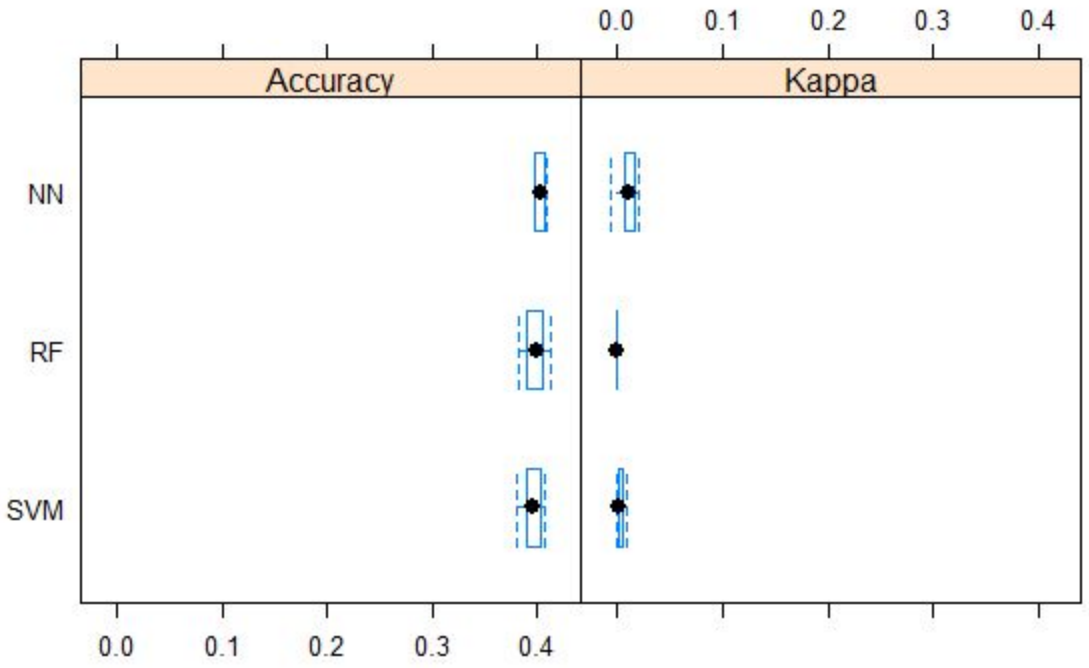


From the table and plot above, we can say that Random Forests give the most accurate results.

- **Bootstrap**

Accuracy:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	0.3978541	0.3984102	0.4040858	0.4034615	0.4074712	0.4096802
RF	0.3820612	0.391531	0.3986125	0.3982897	0.4047227	0.4119429
SVM	0.3810137	0.3910716	0.3963272	0.3951147	0.4016323	0.40625



From the plot and table, it can be observed that neural networks show the maximum accuracy.

Research Question 5:
Predict the number of complaints based on correlation between number of complaints and sum total of assets in the financial institutions

Question 1: Support Vector Machine
To apply this technique in RStudio, we used the kernlab library. We used two different kernels to build our svm (nu-svr) regression model and tested them out. Following are the results:

- **Kernel: Linear**
  The 'vanilladot' kernel in ksvm was used to create the model.

Results:

Predicted	**Actual**
30778.01	26875
39240.41	39075
38877.79	35865
38414.86	40700

Correlation Coefficient: 0.9314985

- **Kernel: Non-linear (Gaussian)**
  The 'rbfdot' kernel in ksvm was used to create the model.

Results:

Predicted	**Actual**
30539.12	26875
33237.65	39075
33408.08	35865
33660.43	40700

Correlation Coefficient: 0.9556043

Question 2: Neural Networks

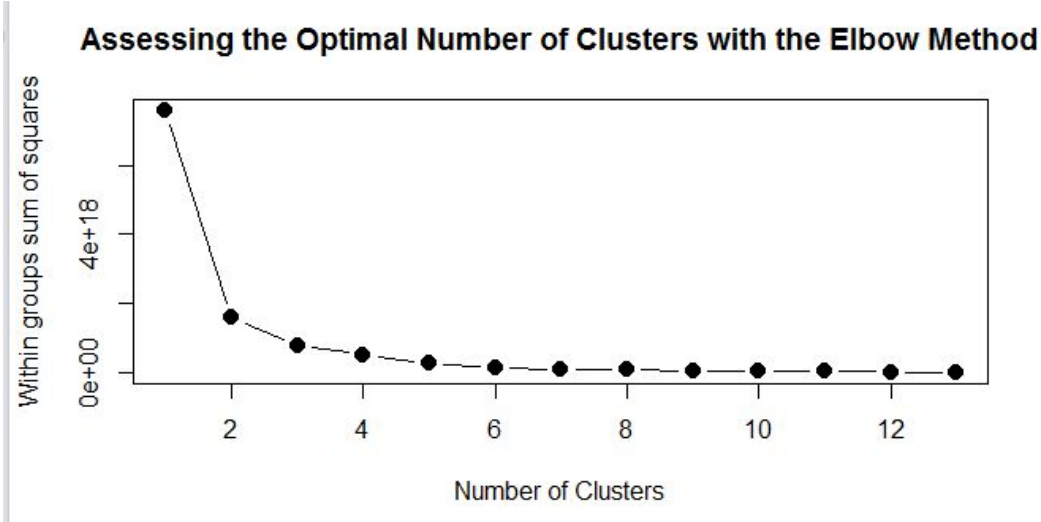
For this question, since the number of records in the training and testing sets is very low (10 and 4), the values for Consumer_Count predicted by neuralnet for the test set are all the same. Hence, no correlation value is generated as there is no standard deviation in the predicted values (shown below).

```
> predicted_count <- d3_result$net.result
> d3_result$net.result
          [,1]
11 30770.80057
12 30770.80057
13 30770.80057
14 30770.80057
> cor(predicted_count,d3_test[,3])
     [,1]
[1,]   NA
Warning message:
In cor(predicted_count, d3_test[, 3]) : the standard deviation is zero
>
```

Question 3: Clustering

We have used an array of techniques for analysis of this dataset. We use the assets_institutional dataset. We have a very limited amount of rows, 14 to be precise, and we will use these for the clustering analysis.

Determining the optimum number of cluster using the elbow method(). With the Elbow method, the solution criterion value (within groups sum of squares) will tend to decrease substantially with each successive increase in the number of clusters. Simplistically, an optimal number of clusters is identified once a “kink” in the line plot is observed. As you can grasp, identifying the point in which a “kink” exists is not a very objective approach and is very prone to heuristic processes.



From 2 onwards, the sum of squares remains more or less constant.

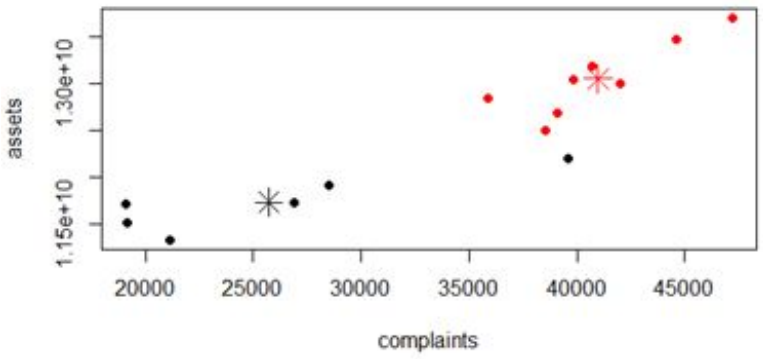
Technique 1: K-means clustering

Since the data is on ratio type, we can use k-means for identifying the clusters. We use the kmeans() function.

Columns used: Count.of.Complaints, Sum.of.Assets

```
> cl
K-means clustering with 2 clusters of sizes 6, 8

Cluster means:
  complaints      assets
1   25730.17 11730239059
2   40979.88 13054308477

Clustering vector:
 6  7  8  9 11 13 14 15 16 17 18 20 21 22
 1  1  1  1  1  1  2  2  2  2  2  2  2  2
```

Scatterplot with centre points



Accuracy : 79.1%

```
Within cluster sum of squares by cluster:
[1] 4.708334e+17 1.113114e+18
 (between_SS / total_SS =  79.1 %)
```

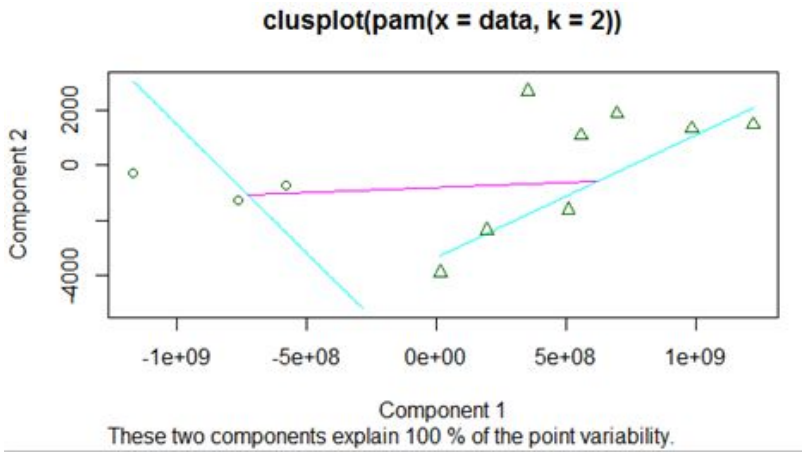
Technique 2: K medoids clustering

For more accurate results, we use clustering around the medoids, and for this we have to make use of pam function

Columns used: Count.of.Complaints, Sum.of.Assets

```
Medoids:
   ID complaints      assets
9   4      26875 11725808508
17 10      39854 13044187413
Clustering vector:
 6  7  8  9 11 13 14 15 16 17 18 20 21 22
 1  1  1  1  1  1  2  2  2  2  2  2  2  2
Objective function:
    build      swap
330011770 263103789

Numerical information per cluster:
     size  max_diss   av_diss   diameter separation
[1,]    6 475505214 215567137  879904858  302023745
[2,]    8 665041892 298756278 1205891838  302023745

Isolated clusters:
 L-clusters: character(0)
 L*-clusters: character(0)


Silhouette plot information:
   cluster neighbor sil_width
8        1        2 0.8070150
9        1        2 0.8052837
7        1        2 0.7795341
6        1        2 0.7168929
11       1        2 0.7093775
13       1        2 0.3372883
17       2        1 0.7401452
20       2        1 0.7376369
18       2        1 0.7303890
21       2        1 0.6865455
16       2        1 0.6518756
22       2        1 0.6217864
15       2        1 0.4999594
14       2        1 0.1855097

Average silhouette width per cluster:
[1] 0.6925652 0.6067310
Average silhouette width of total data set:
[1] 0.6435171

91 dissimilarities, summarized :
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
1.192e+07 4.030e+08 7.953e+08 9.083e+08 1.324e+09 2.388e+09
Metric :  euclidean
Number of objects : 14
```

Scatterplot

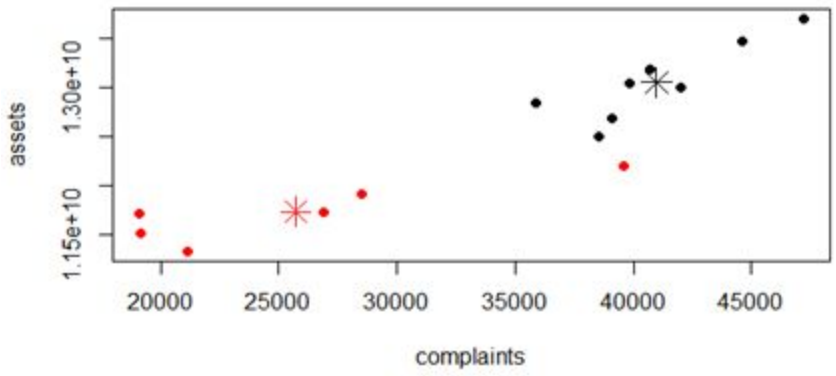


Technique 3: Spectral Clustering

Spectral clustering works by embedding the data points of the partitioning problem into the subspace of the k largest eigenvectors of a normalized affinity/kernel matrix. We use the specc() function in the kernlab package

```
> spcc
Spectral Clustering object of class "specc"

 Cluster memberships:

2 2 2 2 2 2 1 1 1 1 1 1 1 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  5.52928107923125e-17


Centers:
         [,1]        [,2]
[1,] 40979.88 13054308477
[2,] 25730.17 11730239059

Cluster size:
[1] 8 6

Within-cluster sum of squares:
[1] 1.353877e+21 8.203074e+20
```

Scatter plot with centre points



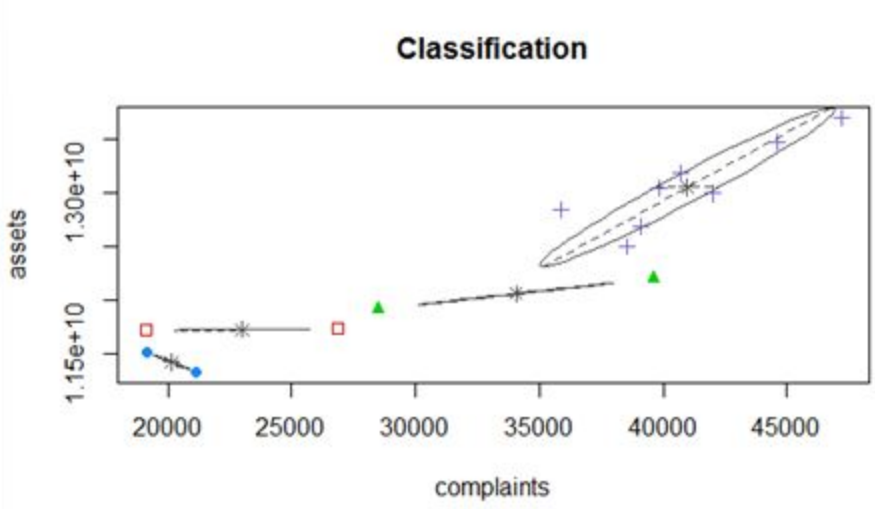
Technique 4: Mixture Model Clustering

Mixture model is a model-based approach, which consists in using certain models for clusters and attempting to optimize the fit between the data and the model.

```
> summary(mcl)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 4 components:

 log.likelihood  n df       BIC      ICL
      -403.6372 14 20 -860.0556 -860.058

Clustering table:
1 2 3 4
2 2 2 8
```

```
> summary(mcl,parameters = TRUE)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 4 components:

 log.likelihood  n df       BIC      ICL
      -403.6372 14 20 -860.0556 -860.058

Clustering table:
1 2 3 4
2 2 2 8

Mixing probabilities:
        1         2         3         4
0.1428571 0.1428571 0.1427726 0.5715132
```

```
> summary(mcl,parameters = TRUE)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VEV (ellipsoidal, equal shape) model with 4 components:

 log.likelihood  n df       BIC      ICL
      -403.6372 14 20 -860.0556 -860.058

Clustering table:
1 2 3 4
2 2 2 8

Mixing probabilities:
        1         2         3         4
0.1428571 0.1428571 0.1427726 0.5715132

Means:
                [,1]        [,2]         [,3]         [,4]
complaints     20150       22996 3.404769e+04 4.097805e+04
assets     11415062468 11719849668 1.205589e+10 1.305414e+10

Variances:
[,,1]
             complaints        assets
complaints  5.298495e+05 -4.762286e+10
assets     -4.762286e+10  4.385499e+15
[,,2]
             complaints       assets
complaints     7523372 1.155716e+10
assets     11557164236 1.775387e+13
[,,3]
             complaints       assets
complaints     15540329 4.052052e+11
assets     405205166029 1.058638e+16

             complaints       assets
complaints 3.543999e+07 4.339418e+12
assets     4.339418e+12 5.566674e+17
>
```



Question 4: Comparative Analysis

We compared three models: Support Vector Machine, Neural Networks and Random Forests using the caret library for three different CV methods: K-Fold, Leave-One-Out Cross Validation and Repeated K-Fold. Following are the results:

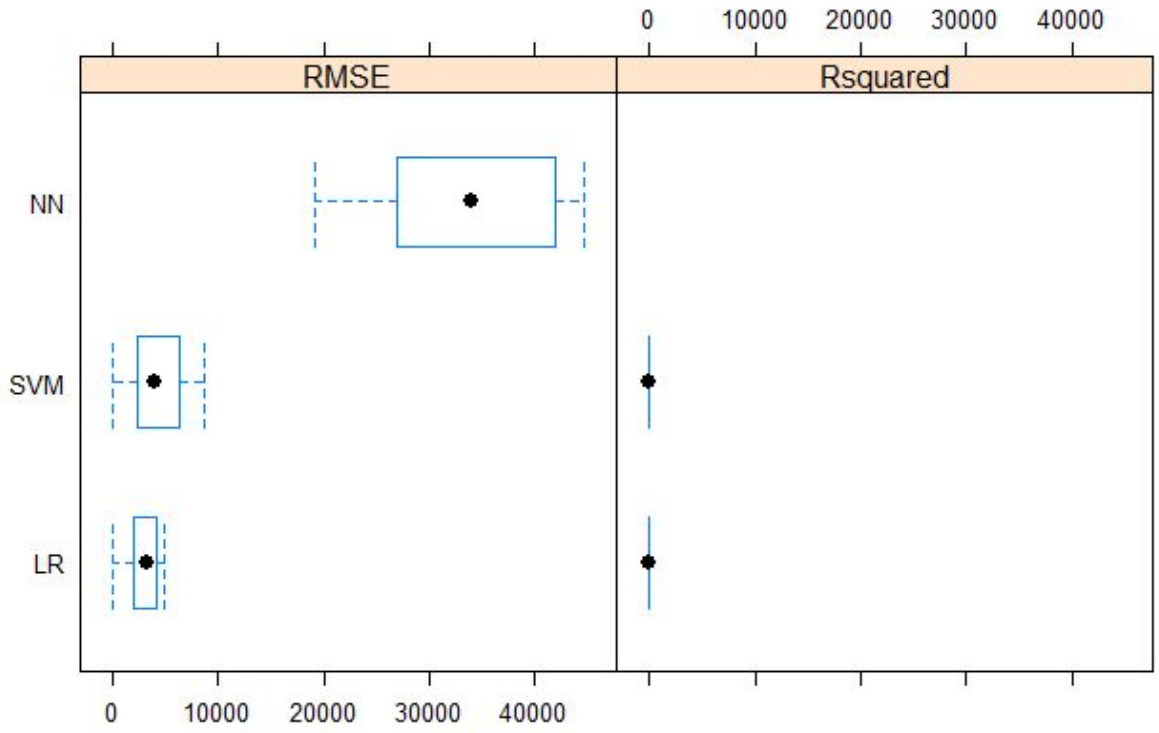
- **K-Fold with 10 folds**

Root Mean Square Error:

Model	**Minimum**	**1st Quantile**	**Median**	**Mean**	**3rd Quantile**	**Maximum**
NN	19130	27270	33970	33370	41530	44620
LR	61.23	2094	3320	2924	4174	4990
SVM	1.107	2659	4077	4280	6065	8630

R-Squared:

Model	**Minimum**	**1st Quantile**	**Median**	**Mean**	**3rd Quantile**	**Maximum**
NN	NA	NA	NA	NaN	NA	NA
LR	1	1	1	1	1	1
SVM	1	1	1	1	1	1



Linear Regression gives the least root mean square error and high R square. Thus we can say that it is the best model out of the three for this question.

- **Repeated K-Fold with 10 folds and 3 repeats**

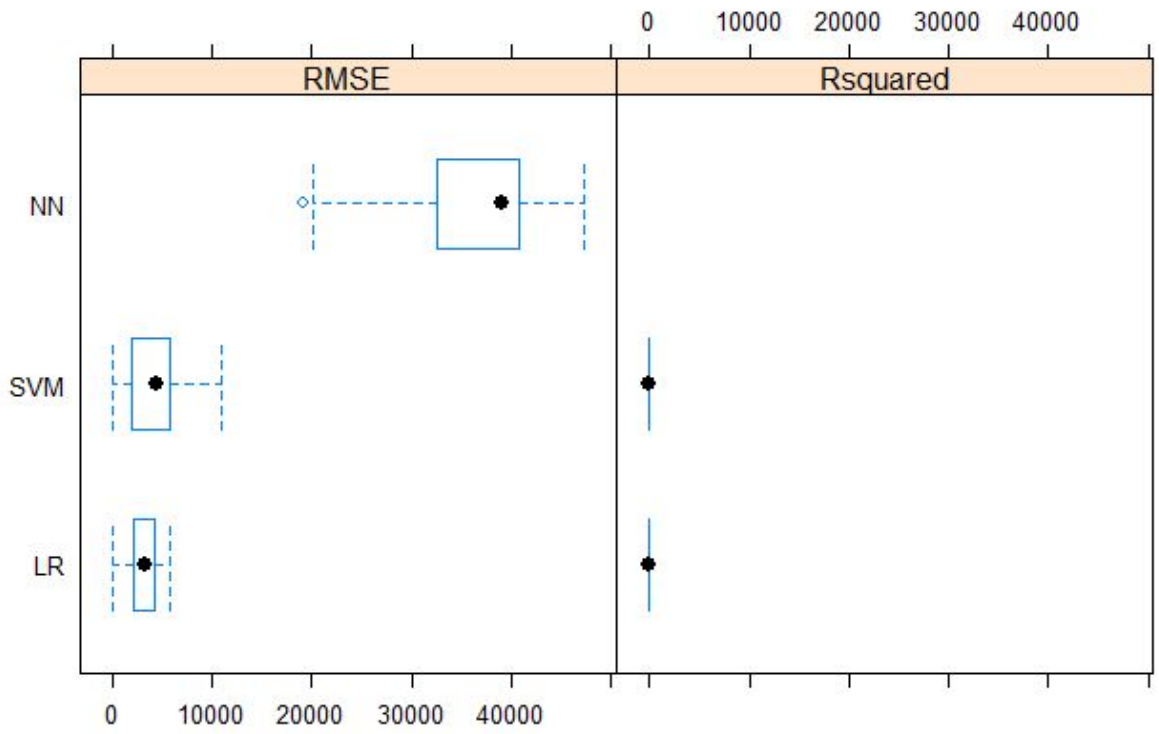
Root Mean Square Error:

Model	**Minimum**	**1st Quantile**	**Median**	**Mean**	**3rd Quantile**	**Maximum**
NN	19130	32480	39060	35920	40490	47200
LR	61.23	2170	3272	3155	4277	5765
SVM	40.44	1939	4481	4111	5726	10960

R-Squared:

Model	**Minimum**	**1st Quantile**	**Median**	**Mean**	**3rd Quantile**	**Maximum**

NN	NA	NA	NA	NaN	NA	NA
LR	1	1	1	1	1	1
SVM	1	1	1	1	1	1



Similar to the above Cross validation technique, Linear Regression gives the least root mean square error and high R square. Thus we can say that it is the best model out of the three for this question.

- **Leave-One-Out Cross Validation**

We tried to use the LOOCV technique but were faced with the following error, which we were unable to solve. Hence, we applied the Bootstrap cross-validation technique.

```
converged
# weights:  26
initial  value 15577966556.647234
final  value 15577683497.920065
converged
# weights:  6
initial  value 17806271889.668900
iter  10 value 17805900788.837254
iter  10 value 17805900653.175842
iter  10 value 17805900547.661411
final  value 17805900547.661411
converged
> model_rf2<-train(Complaint_Count~., data = assets, trControl=train_control2, method = "lm")
> results2      <-      resamples(list(NN=model_nn2,    LM=model_rf2,   SVM=model_svm2))
Error in `[.data.frame`(out, , c(x$perfNames, "Resample")) :
  undefined columns selected
```

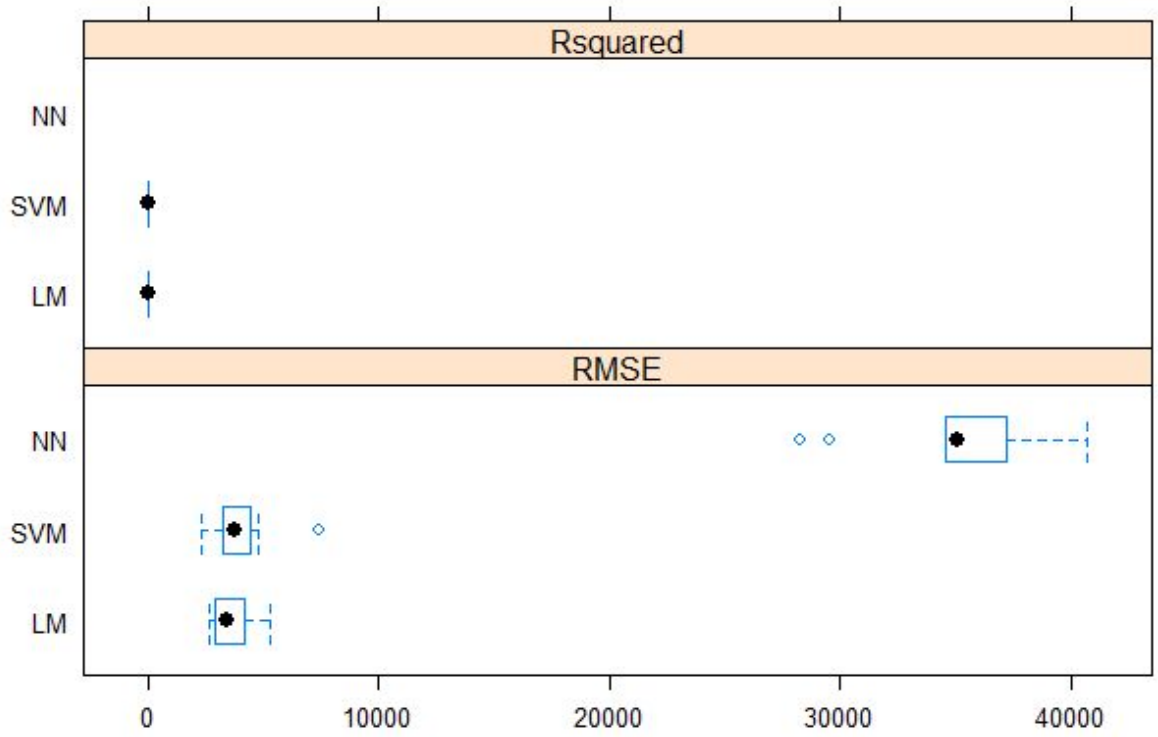
- **Bootstrap**

Root Mean Square Error:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	28250	34570	35090	35020	36850	40710
RF	2629	2910	3433	3567	4089	5309
SVM	2305	3218	3750	4003	4417	7410

R-Squared:

Model	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum
NN	NA	NA	NA	NaN	NA	NA
LM	0.7435	0.8467	0.9092	0.8916	0.9446	0.9743
SVM	0.8582	0.8609	0.888	0.897	0.9165	0.9872



Again, Linear Regression gives the least root mean square error and high R square. Thus we can say that it is the best model out of the three for this question.

Research Question 6:
Predicting the emotion of the consumer based on their complaints

We have a field called as 'Consumer_Complaint_Narative' in our dataset, which contains people's narrations about the complaints. Since these are all complaints, we expect the narratives to have a negative sentiment. But, there are different emotions in these narratives: anger, anticipation, disgust, fear, joy, sadness, surprise, trust. We aim to predict the emotion of the consumer based on his/her narrative.

Predictors: Consumer_Complaint_Narrative
Outcome: emotion

We have used an API called the Aylien API to classify and thus label about 1000 rows of our dataset where each observation is the narrative of the consumer's complaint. We wrote python script to use this API.

NOTE: To be able to build a classifier, we need an emotion lexicon. We have applied to get access to one, but have not still received any correspondence back from the associated party.