Exploring Variations in Clustering and Predictive Analysis

Ran Wei

Mar 23, 2022

Part 1 Plot the Data

For this analysis, I start by importing the data into R. I also read the agaricus-lepiota.names file to gather some additional information. From that file, I learn that this data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. Each species is identified as edible, or poisonous, as shown in the first column. All of the attribute information is listed as Appendix A. According to this information, I assigned names for the features, and a sample of my dataset can be seen in Figure 1. The names file also mentions that missing values are coded as '?', and that there are 2480 of them. Before continuing, these rows containing missing values will be removed from the data.

	class	cshape	csurface	ccolor	odor	bruises	gattachment	gspacing	gsize	gcolor	sshape	sroot	ssar	ssbr	scar	scbr	vtype	vcolor	rnumber	rtype	spc	population	habitat
1	poisonous	x	s	n	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	k	s	u
2	edible	x	s	y	t	a	f	c	b	k	e	c	s	s	w	w	p	w	o	p	n	n	g
3	edible	b	s	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	n	m
4	poisonous	x	y	w	t	p	f	c	n	n	e	e	s	s	w	w	p	w	o	p	k	s	u
5	edible	x	s	g	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g
6	edible	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	n	g
7	edible	b	s	w	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	n	m
8	edible	b	y	w	t	l	f	c	b	n	e	c	s	s	w	w	p	w	o	p	n	s	m
9	poisonous	x	y	w	t	p	f	c	n	p	e	e	s	s	w	w	p	w	o	p	k	v	g
10	edible	b	s	y	t	a	f	c	b	g	e	c	s	s	w	w	p	w	o	p	k	s	m
11	edible	x	y	y	t	l	f	c	b	g	e	c	s	s	w	w	p	w	o	p	n	n	g
12	edible	x	y	y	t	a	f	c	b	n	e	c	s	s	w	w	p	w	o	p	k	s	m
13	edible	b	s	y	t	a	f	c	b	w	e	c	s	s	w	w	p	w	o	p	n	s	g
14	poisonous	x	y	w	t	p	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	v	u
15	edible	x	f	n	f	n	f	w	b	n	t	e	s	f	w	w	p	w	o	e	k	a	g
16	edible	s	f	g	f	n	f	c	n	k	e	e	s	s	w	w	p	w	o	p	n	y	u
17	edible	f	f	w	f	n	f	w	b	k	t	e	s	s	w	w	p	w	o	e	n	a	g

Showing 1 to 18 of 8,124 entries, 23 total columns

Figure 1. Raw Dataset

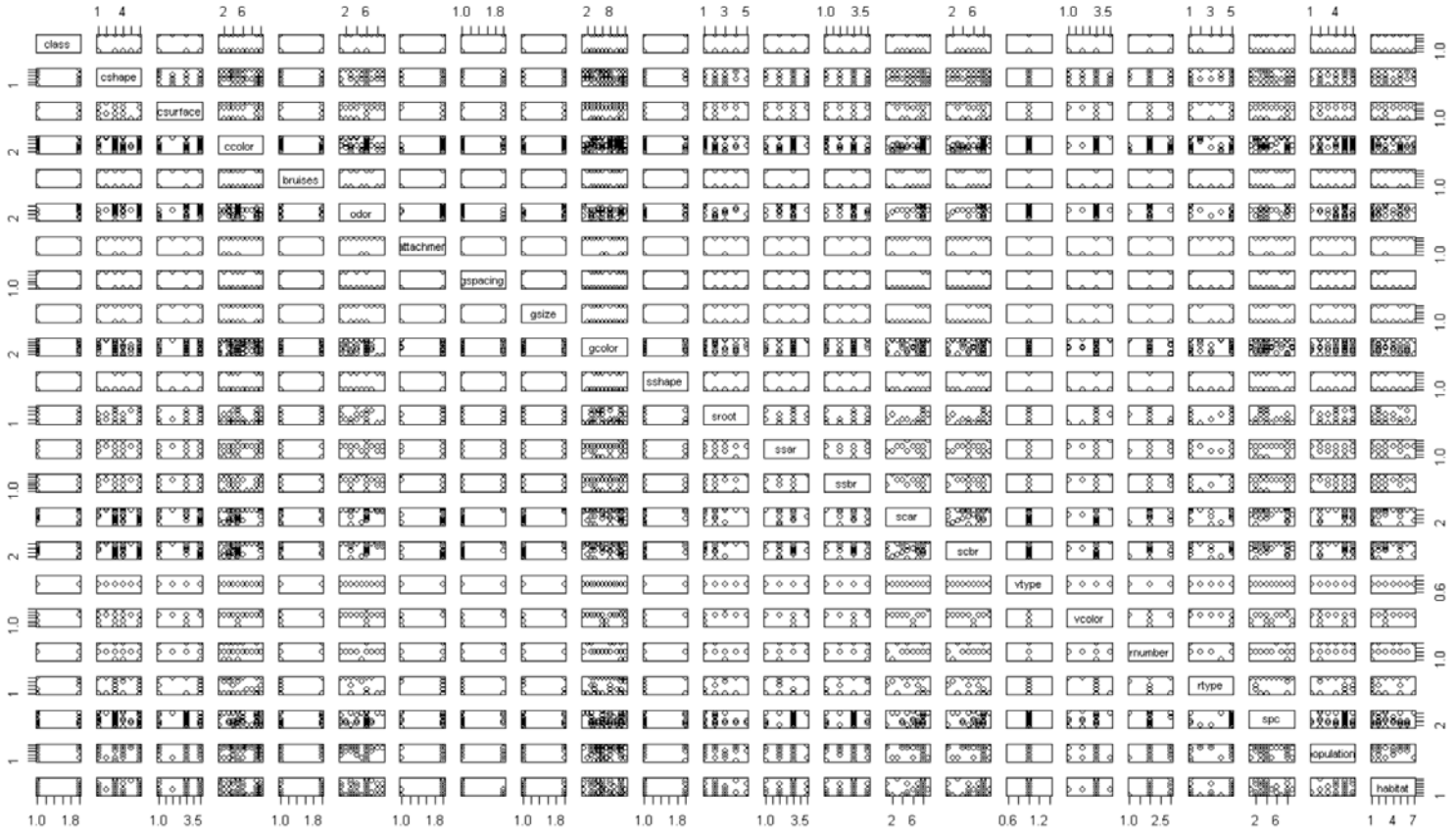


Figure 2. Raw Dataset Plot

The next step in my analysis is to plot the data and see if I can spot any relationships between my features and my class variable, or between two features. The result of this plotting is shown above in Figure 2. It is very difficult to tell if there are any relationships from this plot, so I will need to take a closer look at some smaller sets of relationships. The plots below show my results from plotting 'stalk-surface-below-ring' and 'stalk-surface-above-ring' (Figure 3).

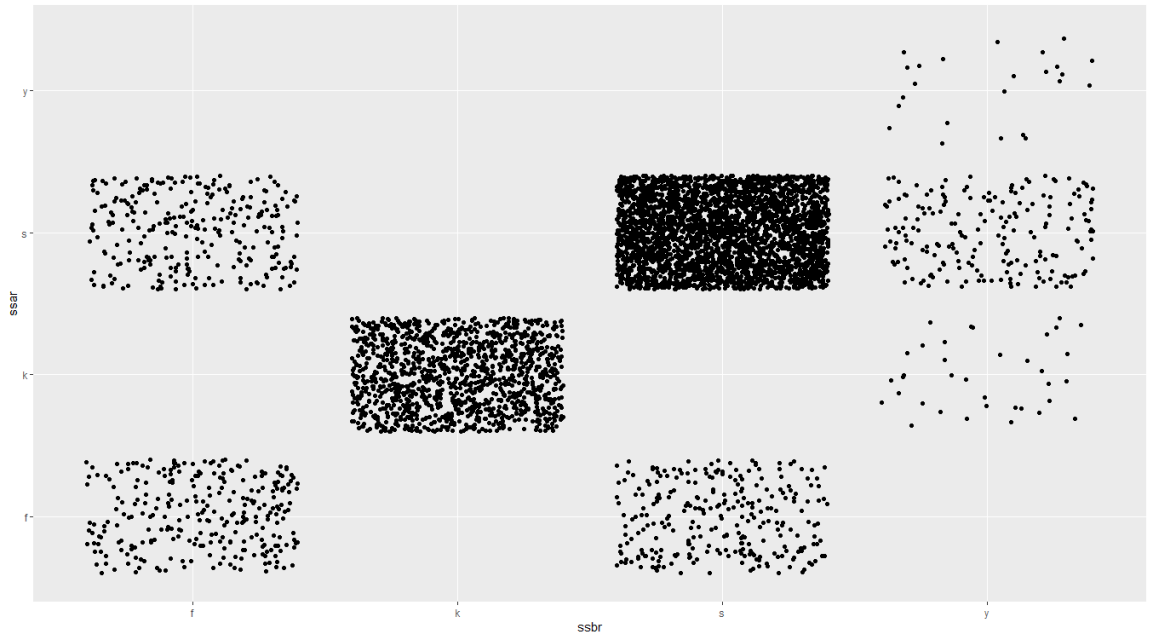


Figure 3. Relationship between ‘ssar’ and ‘ssbr’

It appears there may be a relationship between 'ssar' and 'ssbr'. I also tried to find a relationship between 3 variables, so I investigated the 3 cap related features (‘cshape’, ‘csurface’, and ‘ccolor’) .The result is shown in Figure 4. There seems to be no clear relationship between these three variables.

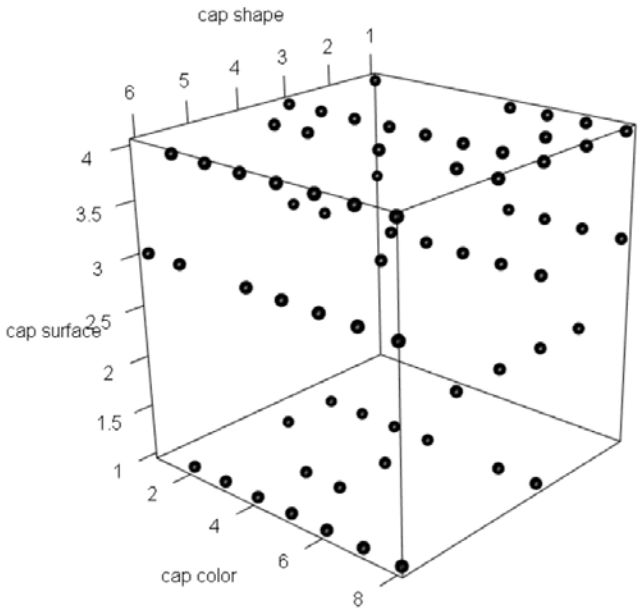


Figure 4. Relationship between ‘cshape’, ‘csurface’, and ‘ccolor’

Part 2 Prepare the Data

The first step in preparing my data for clustering is to convert the categorical variables into numeric. Each variable was assigned a value from 10 to 150 (space by 10) for each category. The mapping for each feature of values to integers is shown in Table 1. All the code for this project can be found in the file 'Project 2.R'. Moreover, I also separated the 'class' column since it's the target value that I want to predict.

Table 1. Character to Number Mapping

cshape	csurface	ccolor	odor	bruises	gattachment	gspacing	gsize	gcolor	sshape	sroot
x=10	s=10	n=10	p=10	t=10	f=10	c=10	n=10	k=10	e=10	e=10
b=20	y=20	y=20	a=20	f=20	a=20	w=20	b=20	n=20	t=20	c=20
s=30	f=30	w=30	l=30					g=30		b=30
f=40	g=40	g=40	n=40					p=40		r=40
k=50		e=50	f=50					w=50		
c=60		p=60	c=60					h=60		
		b=70	y=70					u=70		
		u=80	s=80					e=80		
		c=90	m=90					b=90		
		r=100						r=100		
								y=110		
								o=120		

ssar	ssbr	scar	scbr	vtype	vcolor	rnumber	rtype	spc	population	habitat
s=10	s=10	w=10	w=10	p=10	w=10	o=10	p=10	k=10	s=10	u=10
f=20	f=20	g=20	p=20	u=20	n=20	t=20	e=20	n=20	n=20	g=20
k=30	y=30	p=30	g=30		o=30	n=30	l=30	u=30	a=30	m=30
y=40	k=40	n=40	b=40		y=40		f=40	h=40	v=40	d=40
		b=50	n=50				n=50	w=50	y=50	p=50
		e=60	e=60					r=60	c=60	w=60
		o=70	y=70					o=70		l=70
		c=80	o=80					y=80		
		y=90	c=90					b=90		

- The results from applying some basic statistics are shown in Figure 5 and Figure 6.

```
> summary(mushroom.nc)
     cshape          csurface         ccolor          bruises          odor          gattachment      gspacing         gsize      
 Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00  
 1st Qu.:10.00   1st Qu.:20.00   1st Qu.:20.00   1st Qu.:10.00   1st Qu.:40.00   1st Qu.:10.00   1st Qu.:10.00   1st Qu.:20.00  
 Median :10.00   Median :20.00   Median :30.00   Median :10.00   Median :40.00   Median :10.00   Median :10.00   Median :20.00  
 Mean   :23.86   Mean   :21.61   Mean   :30.92   Mean   :14.36   Mean   :40.32   Mean   :10.03   Mean   :11.81   Mean   :18.75  
 3rd Qu.:40.00   3rd Qu.:30.00   3rd Qu.:40.00   3rd Qu.:20.00   3rd Qu.:50.00   3rd Qu.:10.00   3rd Qu.:10.00   3rd Qu.:20.00  
 Max.   :60.00   Max.   :40.00   Max.   :90.00   Max.   :20.00   Max.   :90.00   Max.   :20.00   Max.   :20.00   Max.   :20.00  
     gcolor          sshape          sroot           ssar            ssbr            scar            scbr           vtype      
 Min.   : 10.00   Min.   :10.0   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.0   Min.   :10.00   Min.   :10  
 1st Qu.: 30.00   1st Qu.:10.0   1st Qu.:20.00   1st Qu.:10.00   1st Qu.:10.00   1st Qu.:10.0   1st Qu.:10.00   1st Qu.:10  
 Median : 40.00   Median :20.0   Median :30.00   Median :10.00   Median :10.00   Median :10.0   Median :10.00   Median :10  
 Mean   : 40.52   Mean   :15.1   Mean   :25.39   Mean   :15.83   Mean   :16.91   Mean   :20.6   Mean   :20.23   Mean   :10  
 3rd Qu.: 50.00   3rd Qu.:20.0   3rd Qu.:30.00   3rd Qu.:20.00   3rd Qu.:30.00   3rd Qu.:30.0   3rd Qu.:30.00   3rd Qu.:10  
 Max.   :110.00   Max.   :20.0   Max.   :40.00   Max.   :40.00   Max.   :40.00   Max.   :90.0   Max.   :90.00   Max.   :10  
     vcolor          rnumber          rtype            spc          population      habitat      
 Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00   Min.   :10.00  
 1st Qu.:10.00   1st Qu.:10.00   1st Qu.:10.00   1st Qu.:10.00   1st Qu.:30.00   1st Qu.:20.00  
 Median :10.00   Median :10.00   Median :10.00   Median :20.00   Median :40.00   Median :40.00  
 Mean   :10.04   Mean   :10.34   Mean   :16.31   Mean   :23.68   Mean   :35.72   Mean   :32.28  
 3rd Qu.:10.00   3rd Qu.:10.00   3rd Qu.:20.00   3rd Qu.:40.00   3rd Qu.:50.00   3rd Qu.:40.00  
 Max.   :40.00   Max.   :30.00   Max.   :50.00   Max.   :60.00   Max.   :60.00   Max.   :70.00  
>
```

Figure 5. Summary of Numerical Dataset

```
> describe(mushroom.nc)
            vars    n  mean    sd median trimmed   mad min max range  skew kurtosis   se
cshape         1 5644 23.86 14.67     10   23.48  0.00  10  60    50  0.20    -1.88 0.20
csurface       2 5644 21.61  7.64     20   22.00 14.83  10  40    30 -0.27    -1.22 0.10
ccolor         3 5644 30.92 15.80     30   30.02 14.83  10  90    80  0.54     0.45 0.21
bruises        4 5644 14.36  4.96     10   14.20  0.00  10  20    10  0.26    -1.93 0.07
odor           5 5644 40.32 11.57     40   41.47 14.83  10  90    80 -0.41     2.45 0.15
gattachment    6 5644 10.03  0.56     10   10.00  0.00  10  20    10 17.62   308.45 0.01
gspacing       7 5644 11.81  3.85     10   11.02  0.00  10  20    10  1.65     0.73 0.05
gsize          8 5644 18.75  3.30     20   19.69  0.00  10  20    10 -2.27     3.16 0.04
gcolor         9 5644 40.52 18.04     40   40.04 14.83  10 110   100  0.30     0.05 0.24
sshape        10 5644 15.10  5.00     20   15.13  0.00  10  20    10 -0.04    -2.00 0.07
sroot         11 5644 25.39  8.45     30   26.31  0.00  10  40    30 -0.94    -0.44 0.11
ssar          12 5644 15.83  8.61     10   14.73  0.00  10  40    30  0.95    -0.86 0.11
ssbr          13 5644 16.91  9.70     10   15.58  0.00  10  40    30  0.94    -0.64 0.13
scar          14 5644 20.60 14.27     10   18.13  0.00  10  90    80  1.32     1.56 0.19
scbr          15 5644 20.23 14.47     10   17.49  0.00  10  90    80  1.57     2.64 0.19
vtype         16 5644 10.00  0.00     10   10.00  0.00  10  10     0   NaN      NaN 0.00
vcolor        17 5644 10.04  1.13     10   10.00  0.00  10  40    30 26.50   700.25 0.02
rnumber       18 5644 10.34  2.14     10   10.00  0.00  10  30    20  6.94    51.67 0.03
rtype         19 5644 16.31  8.77     10   15.23  0.00  10  50    40  1.02    -0.07 0.12
spc           20 5644 23.68 13.30     20   22.70 14.83  10  60    50  0.62    -0.88 0.18
population    21 5644 35.72 14.71     40   37.03 14.83  10  60    50 -0.78    -0.77 0.20
habitat       22 5644 32.28 12.51     40   32.13 14.83  10  70    60  0.02    -0.63 0.17
```

Figure 6. Describe of Numerical Dataset

It's worth noticing that:

- There are no 'NA' values in the whole dataset, as they have all been removed.
- The veil-type feature only has one category, as all the entries are of value 10 (belonging to category 'p').
- Most of features have positive skew value, which means that their tails are on the right side of the possibility density distribution, except cap-surface, odor, gill-size, stalk-shape, stalk-root, and population.
- Most of fields are distributed platykurtic with negative kurtosis value, however, veil-color (kurtosis=700.25), gill-attachment'(kurtosis=308.45), and ring-number (kurtosis=51.67) have relatively higher kurtosis values than other fields. This means that outliers might exist in these three features.

In order to subset the dataset for prediction, I need to eliminate some attributes. 'vtype' should be dropped first since other features cannot have a relationship with it. Then I plotted the remaining 21 attributes again shown in Figure 7.

- Some attributes seem to present only horizontal or vertical lines for all relationships. These features are 'cshape', 'csurface', 'bruises', 'gattachment', 'gspacing', 'gsize', 'sshape', 'vcolor', and 'rnumber', and I choose to get rid of them.
- But for other features, like 'ccolor', 'odor', 'gcolor', 'sroot', 'ssar', 'ssbr', 'scar', 'scbr', 'rtype', 'spc', 'popilation', and 'habitat', the dots in subplot tend to converge towards the bottom left corner, which means that these attributes may have some relationship with others. After keeping only these attributes, I now have 12 variables remaining.

The next step is to normalize the data and check the correlation. For my analysis, I chose to use Z-score normalization, but my code shows how to execute both.

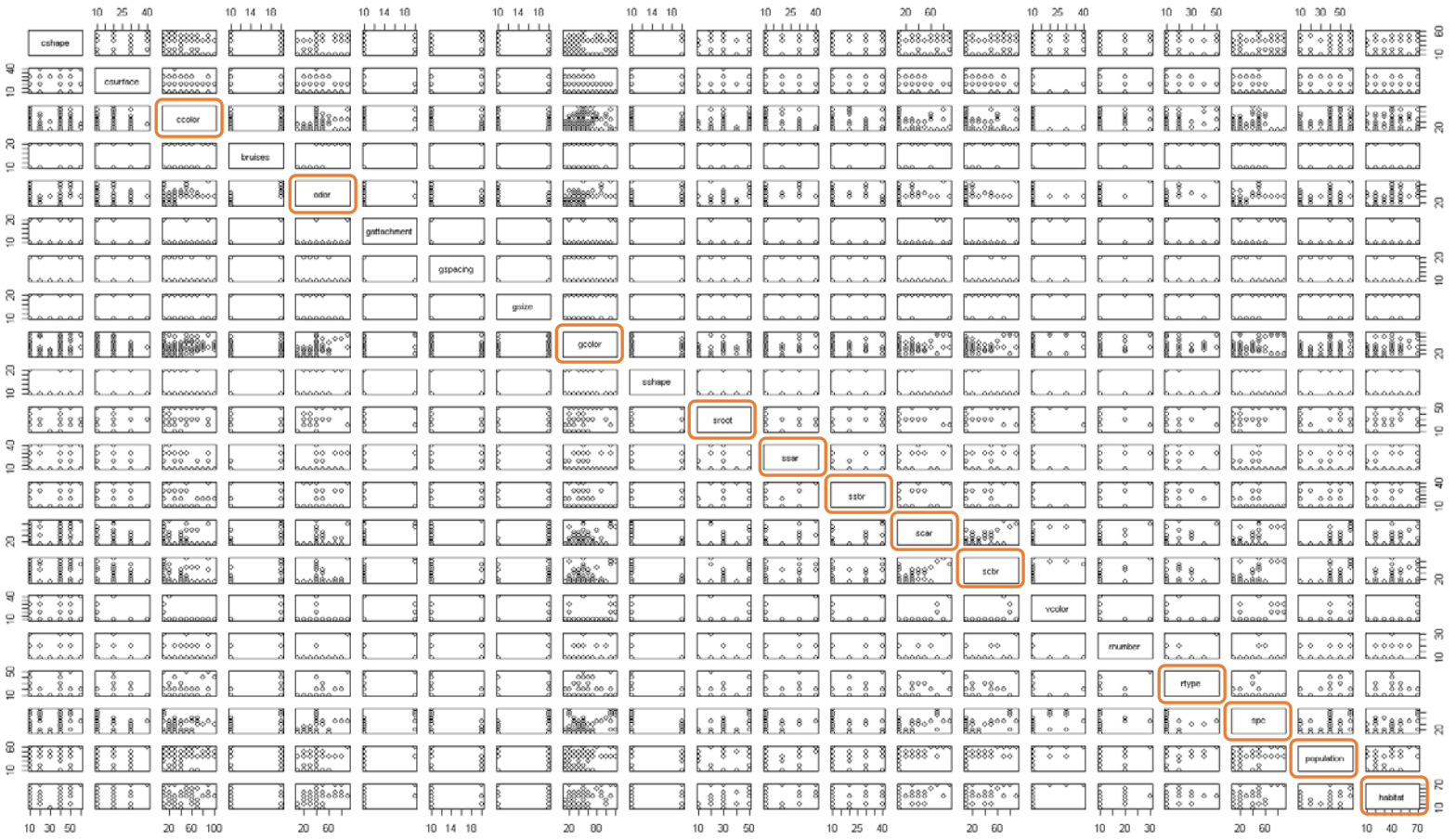


Figure 7. Plot of Numeric Dataset

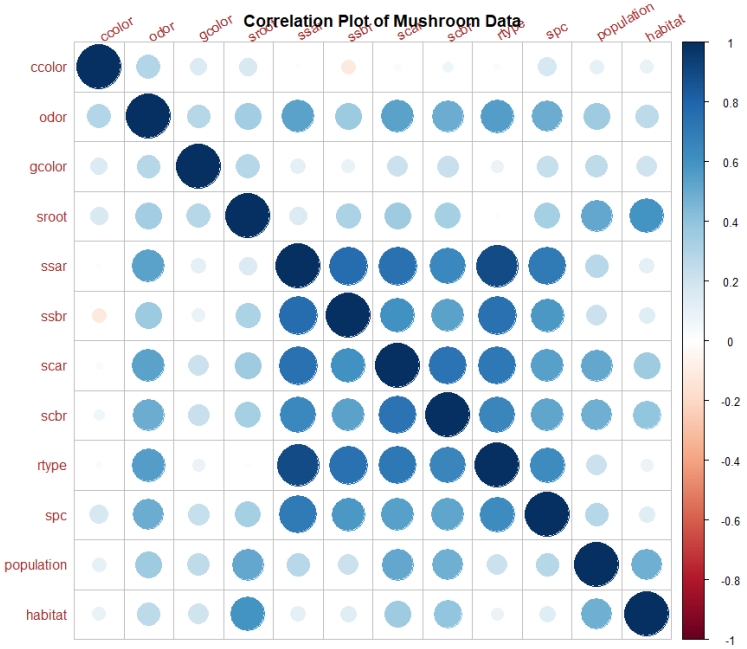


Figure 8. Correlation Plot

According to Figure 8, ‘ccolor’ has a weak relationship with other attributes, so I choose to eliminate it as well. Finally, I got 11 features as a subset.

Then I split the original dataset into 3 training and testing sets of varying sizes (70/30,60/40,50/50).

Part 3 Clustering Data Set

The next step for the analysis is to take the normalized features and apply kmeans clustering. I started with 2 clusters and repeated the analysis for up to 7 clusters. The results of the analysis are shown below. All the cluster plots are shown in Appendix B.

Note: When creating the cluster plots, I applied a function so that only every third point label is displayed (see code). This will help me identify nearby points in a later part of the analysis.

Table 2. Kmeans Clustering Performance

k	2	3	4	5	6	7
cluster size	1404 4240	1368 1776 2500	2444 1008 824 1368	2302 1368 768 1014 192	1072 360 1632 192 1020 1368	864 40 944 432 2444 288 632
within cluster sum of squares	6661.72 29752.72	5870.57 8469.03 11764.14	10956.17 4057.70 2926.89 5870.57	8634.28 5870.57 1911.02 4123.10 786.10	2224.26 1104.26 6399.90 786.10 2560.47 5870.57	2337.41 253.19 4537.19 718.83 10785.46 311.32 1496.33
between_SS/total_SS	**41.30%**	**57.90%**	**61.60%**	**65.60%**	**69.50%**	**67.10%**

It’s obvious that:

- According to the ratio of between_SS and total_SS, the metrics improve along with the increasing number of clusters, though there is a slight dip when the number of clusters is equal to 7.
- When the number of clusters increased from 5 to 6, the ratio of between_SS and total_SS still improved almost 4%.
- Some clear distinctions appear between 2 and 3 clusters. However, as shown in Appendix B, lots of overlap appears when the number of clusters is larger than 3.

To determine the characteristics which have influence on classification, I chose two points: ‘5436’ and ‘5142’ and go back to their character features. These points are

clustered together when there's 2-6 clusters, but clustered separately when there's 7 clusters. As indicated in Figure 9, the features for these two points are nearly identical, with the only difference being 'ssbr'. From this I can assume 'ssbr' plays a larger role when there are 7 clusters. This explains why the overlapping increases when the number of clusters becomes larger.

```
> mushroom.nc.11[5436,]
     odor gcolor sroot ssar ssbr scar scbr rtype spc population habitat
5805   40     50    30   10   10   10   50    20  50         40      70
> mushroom.nc.11[5142,]
     odor gcolor sroot ssar ssbr scar scbr rtype spc population habitat
5319   40     50    30   10   20   10   50    20  50         40      70
```

Figure 9. Attributes Record for Two Specific Points

Part 4 Prediction

I start part 4 by performing knn on my testing data and comparing the results to the clustering. To select the number of clusters to use in this test, I look at the within sum of squares plot generated in part 3 (Figure 10). The chart shows that the optimal number of clusters should be around 3-4. Since the question suggests I also perform my analysis for 5,7,and 9 clusters, I perform this analysis 4 times, with 3,5,7, and 9 clusters.

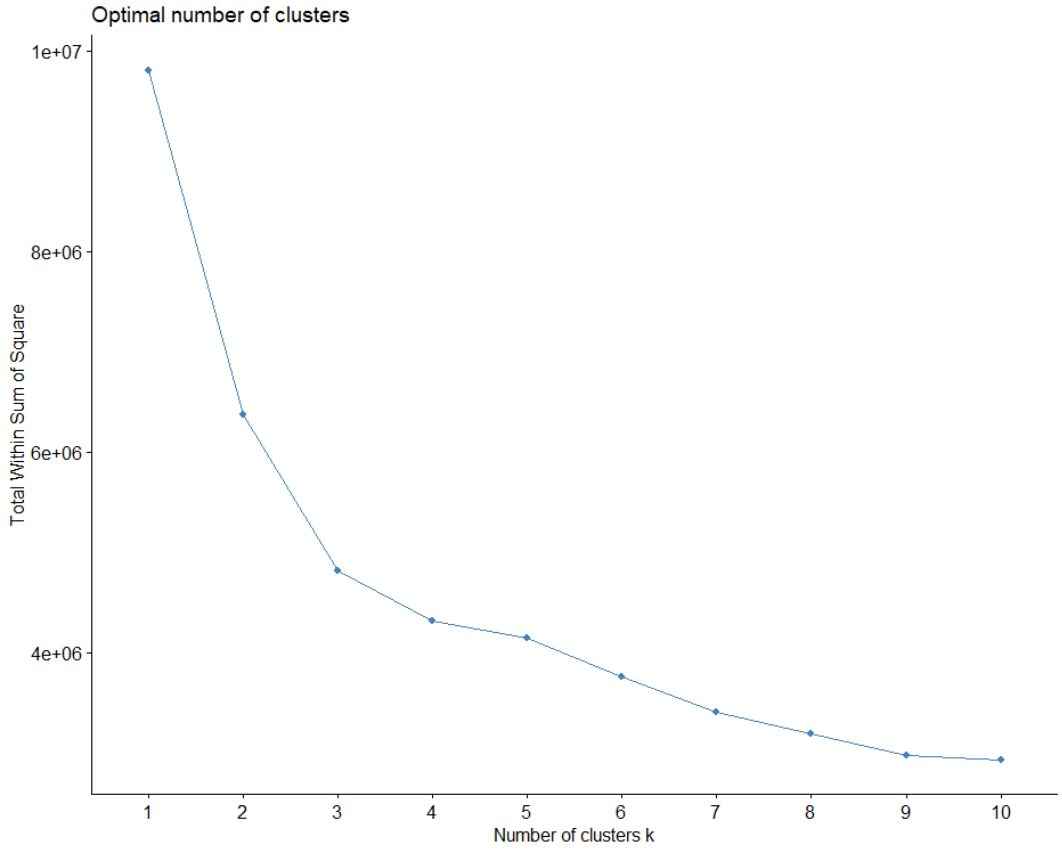


Figure 3. Cluster Sum of Squares

If I compare the results of knn classification of the test set (Figure 11) to the kmeans clustering labels (Figure 12) when k=3, I notice they vary drastically. This could be

because the train and test sets possess different qualities, so their clusters are created using different factor weights.

```
> mush.test.k3
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 1 3 3 3 3 1 1 1 3 3 3 3 1 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 1 3 3 1 3 3 3 3 3 1 3 3 3 3 1 3 3 1 3 3 3 3 3
 [67] 3 1 3 1 3 3 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 3 3 3 3 1 3 1 3 3 3 3 3 1 3 3 3 1 3 3 1 3 3 1 3 1 3 1 3 1 3 3 3 3 3 1 3 3 1 3 1
[133] 3 3 3 3 3 3 3 3 3 3 3 1 3 3 1 3 3 3 1 1 3 3 3 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 1 1 3 3 3 3 1 1 1 3 3 3 3 3 3 3 3 3 3 3
[199] 3 3 3 3 3 1 3 3 1 3 3 1 1 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 1 3 1 3 3 3 3 1 3 3 3 3 3 3 3 1 3 3 1 3 1 1 1 1 1 1 1 3 1 3 3 3 3 3 3 3 1 3 3 3 1
[265] 3 3 3 3 3 3 3 3 3 3 1 3 3 1 3 1 1 3 3 3 3 3 1 3 3 3 3 3 1 1 1 3 3 3 3 1 3 3 3 3 3 3 3 3 1 1 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3
[331] 3 3 1 1 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 3 1 1 3 3 3 3 3 3 3 3 3 3 1
[397] 3 1 1 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[463] 3 3 3 3 3 3 1 3 3 3 1 3 3 3 1 3 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 1 1 3 3 3 1 3 3 3 3 3 3 3 2 3 3 1 3 3 1
[529] 1 1 1 1 1 1 3 3 1 3 1 1 3 1 3 3 3 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 3 3 1 1 1 3 3 3 3 3 3 3 3 3 3 3 1 1 3 1 3 3 3 3 3 3 3 3 3 3
[595] 3 1 3 3 3 3 1 1 3 3 3 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 2 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1
[661] 1 3 1 1 3 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[727] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1
[793] 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
[859] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 1 1 2 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1
[925] 1 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1
[991] 1 1 1 1 1 1 1 1 2 1
[ reached getOption("max.print") -- omitted 694 entries ]
```

Figure 11: KNN Labels on Test Set (k=3)

```
> mush.test.k3.labels
  3   4   6   8  11  14  15  22  23  24  29  33  38  39  40  43  46  48  50  53  62  64  73  74  80  84  85
  2   2   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   1   2   2   2   1   1   1   2   2
 86  88  89  91  93 101 104 105 106 110 111 115 126 128 129 130 135 142 146 155 156 157 158 160 174 175 176
  2   2   1   2   2   2   1   2   2   2   2   2   2   2   2   2   1   2   2   1   2   2   2   2   2   2   2
177 186 189 190 199 202 205 208 210 221 225 226 228 233 235 236 237 238 245 247 248 253 257 265 273 285 288
  2   2   2   1   2   2   1   2   2   2   2   2   2   1   2   1   2   2   1   1   2   2   2   1   2   2   2
293 298 301 305 308 310 312 313 320 322 324 330 332 339 348 349 356 360 367 380 385 389 391 397 399 405 410
  2   2   2   2   2   2   2   2   1   1   2   2   2   2   2   2   2   2   1   2   1   2   2   2   2   2   1
411 412 416 417 423 425 426 430 437 439 440 446 450 452 461 463 465 468 475 477 480 486 488 491 495 499 501
  2   2   2   1   2   2   1   2   1   2   1   2   1   2   2   2   2   2   1   2   2   1   2   1   2   2   2
508 510 512 515 516 519 520 521 530 533 538 540 545 546 548 554 563 565 566 571 575 576 579 580 589 591 592
  2   2   2   2   2   2   2   2   1   2   2   1   2   2   2   1   1   2   2   2   1   2   1   2   2   2   2
593 595 597 611 619 622 628 630 631 632 635 637 638 644 645 648 650 653 654 657 662 676 679 681 685 687 689
  2   2   2   2   2   2   2   2   2   2   2   2   2   1   1   2   1   1   2   2   2   2   1   1   1   2   2
691 692 695 698 699 700 702 709 717 722 724 725 726 731 734 736 737 738 749 751 762 766 768 769 770 771 773
  2   2   2   2   2   2   2   2   2   2   2   2   2   1   2   2   1   2   2   1   1   2   2   2   2   2   2
775 776 781 784 792 800 801 806 807 808 811 818 823 824 826 833 838 844 848 849 852 853 857 863 865 868 869
  2   2   2   2   1   1   2   2   2   1   2   1   2   2   2   1   2   2   2   2   2   2   1   2   2   1   2
874 876 888 889 891 892 895 901 902 908 909 911 917 921 922 924 927 935 937 939 940 941 942 944 947 952 957
  1   1   1   1   1   1   2   1   2   2   2   2   2   2   1   2   2   2   2   1   2   2   2   2   2   2   2
```

Figure 12: KMeans Labels on Test Set(k=3)

I take the kmeans generated labels and move on to linear modeling. For the linear model, I bring the poisonous/edible class back into the data. I convert these categories to 1,0 values respectively and use them as the dependent variable in my regression. The results of my first regression model (Figure 13) show that feature ‘rtype’ is the least significant and should be eliminated. Applying stepwise regression, I retrain the model with my new features and check the results (Figure 14). The p-values of all features after stepwise regression are less than 0.05, which means that this model passes the significance test. The anova - Analysis of Variance on the model (Figure 15) also shows the gap between the null deviance and the residual deviance is significant.

```
> summary(mush.train.glm)

Call:
glm(formula = mush.train$class ~ ., family = "gaussian", data = mush.train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.19889  -0.18468  -0.04705   0.08540   1.00253

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.384163   0.004959  77.468  < 2e-16 ***
odor         0.009326   0.007272   1.282  0.19978
gcolor      -0.004229   0.005393  -0.784  0.43299
sroot        0.061683   0.010003   6.167 7.68e-10 ***
ssar         0.173709   0.013532  12.837  < 2e-16 ***
ssbr        -0.031371   0.009965  -3.148  0.00165 **
scar         0.052570   0.009537   5.512 3.77e-08 ***
scbr        -0.025376   0.008208  -3.092  0.00200 **
rtype       -0.004309   0.016511  -0.261  0.79414
spc          0.223388   0.007887  28.325  < 2e-16 ***
population  -0.029797   0.006675  -4.464 8.26e-06 ***
habitat     -0.105915   0.006978 -15.178  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.09709393)

    Null deviance: 933.23  on 3949  degrees of freedom
Residual deviance: 382.36  on 3938  degrees of freedom
AIC: 2011.9

Number of Fisher Scoring iterations: 2
```

Figure 13. Linear Regression Model for All Variables in Training Set

```
> summary(mush.train.glm.step)

Call:
glm(formula = mush.train$class ~ sroot + ssar + ssbr + scar +
    scbr + spc + population + habitat, family = "gaussian", data = mush.train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.19944  -0.18611  -0.04847   0.08334   1.00058

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.384110   0.004958  77.471  < 2e-16 ***
sroot        0.064621   0.007505   8.610  < 2e-16 ***
ssar         0.176070   0.011491  15.322  < 2e-16 ***
ssbr        -0.034401   0.008389  -4.101 4.20e-05 ***
scar         0.052657   0.009204   5.721 1.14e-08 ***
scbr        -0.025172   0.008023  -3.138  0.00172 **
spc          0.223350   0.007728  28.900  < 2e-16 ***
population  -0.029843   0.006663  -4.479 7.71e-06 ***
habitat     -0.106269   0.006860 -15.490  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.09707253)

    Null deviance: 933.23  on 3949  degrees of freedom
Residual deviance: 382.56  on 3941  degrees of freedom
AIC: 2008

Number of Fisher Scoring iterations: 2
```

Figure 14. Linear Regression Model after Stepwise Regression

```
> anova(object = mush.train.glm.step, test='Chisq')
Analysis of Deviance Table

Model: gaussian, link: identity

Response: mush.train$class

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                        3949     933.23
sroot       1    30.40      3948     902.83 < 2.2e-16 ***
ssar        1   385.59      3947     517.24 < 2.2e-16 ***
ssbr        1     0.33      3946     516.91  0.066754 .
scar        1     0.89      3945     516.02  0.002412 **
scbr        1     3.38      3944     512.64 3.539e-09 ***
spc         1   102.95      3943     409.69 < 2.2e-16 ***
population  1     3.83      3942     405.86 3.347e-10 ***
habitat     1    23.29      3941     382.56 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 15. ANOVA Analysis result

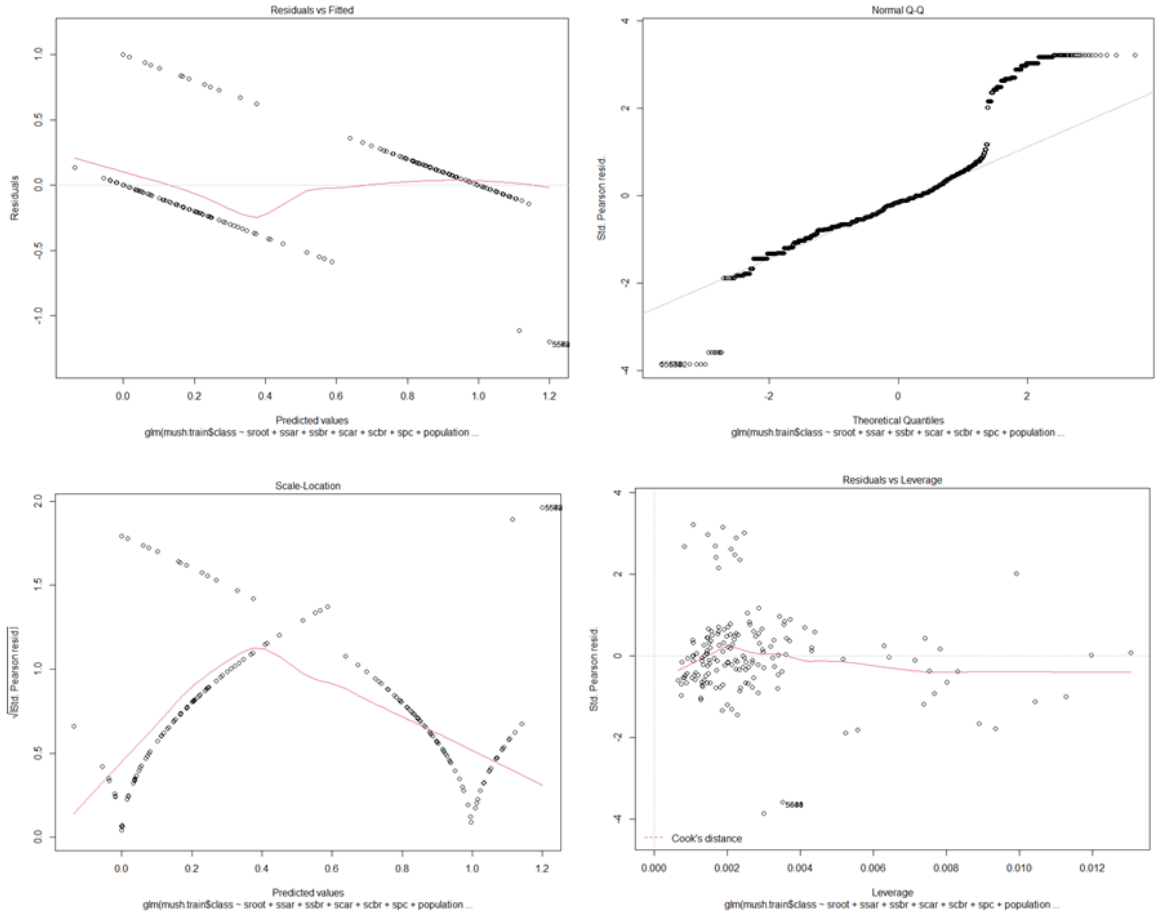


Figure 16. Model Plot

Figure 16 shows me the residual plots from the model, and they can be interpreted as follows:

- Residuals vs Fitted: my residuals appear to be independent from the predicted values and relatively normally distributed. The strange shape is due to the nature of my true values (0,1 classes).
- Normal Q-Q: For this plot, the residuals should follow the trend line top confirm normal distribution. In my case this is roughly true, however some points are a bit skewed.
- Scale Location: Again, the strange shape is expected due to the nature of my true values.
- Residuals vs Leverage: In this plot, those points in the upper left and upper right may be highly influential on my model.

To perform the prediction, I used the predict() function on the test set and retrieved values. Then, I clustered these predictions using kmeans, and compared the results to the true values generated earlier in this section. The results for cluster sizes 3,5,7,and 9 are shown below.

	mush.test.k3.labels			
mush.test.pred.k3$cluster	1	2	3	Row Total
1	34	20	184	238
	48.846	39.704	276.058	
	0.143	0.084	0.773	0.140
	0.045	0.038	0.448	
	0.020	0.012	0.109	
2	51	26	227	304
	52.533	50.054	318.391	
	0.168	0.086	0.747	0.179
	0.068	0.049	0.552	
	0.030	0.015	0.134	
3	669	483	0	1152
	47.610	42.229	279.499	
	0.581	0.419	0.000	0.680
	0.887	0.913	0.000	
	0.395	0.285	0.000	
Column Total	754	529	411	1694
	0.445	0.312	0.243	

Figure 17. Comparing Result (k=3)

mush.pred.k5$cluster	mush.test.k5.labels 1	2	3	4	5	Row Total
1	0	33	0	0	143	176
	65.558	17.398	3.532	48.623	242.057	
	0.000	0.188	0.000	0.000	0.812	0.104
	0.000	0.212	0.000	0.000	0.353	
	0.000	0.019	0.000	0.000	0.084	
2	0	43	0	0	224	267
	99.455	13.787	5.359	73.764	401.871	
	0.000	0.161	0.000	0.000	0.839	0.158
	0.000	0.276	0.000	0.000	0.553	
	0.000	0.025	0.000	0.000	0.132	
3	0	41	20	0	38	99
	36.877	111.500	163.294	27.351	8.677	
	0.000	0.414	0.202	0.000	0.384	0.058
	0.000	0.263	0.588	0.000	0.094	
	0.000	0.024	0.012	0.000	0.022	
4	284	20	14	323	0	641
	8.569	25.806	0.100	120.223	153.250	
	0.443	0.031	0.022	0.504	0.000	0.378
	0.450	0.128	0.412	0.690	0.000	
	0.168	0.012	0.008	0.191	0.000	
5	347	19	0	145	0	511
	128.933	16.729	10.256	0.104	122.169	
	0.679	0.037	0.000	0.284	0.000	0.302
	0.550	0.122	0.000	0.310	0.000	
	0.205	0.011	0.000	0.086	0.000	
Column Total	631	156	34	468	405	1694
	0.372	0.092	0.020	0.276	0.239	

Figure 18. Comparing Result (k=5)

mush.pred.k7$cluster	mush.test.k7.labels 1	2	3	4	5	6	7	Row Total
1	152	0	135	132	0	17	0	436
	56.448	62.028	38.207	1.107	46.071	0.571	30.113	
	0.349	0.000	0.310	0.303	0.000	0.039	0.000	0.257
	0.469	0.000	0.435	0.282	0.000	0.309	0.000	
	0.090	0.000	0.080	0.078	0.000	0.010	0.000	
2	0	124	0	0	50	0	24	198
	37.870	326.020	36.234	54.701	40.413	6.429	7.795	
	0.000	0.626	0.000	0.000	0.253	0.000	0.121	0.117
	0.000	0.515	0.000	0.000	0.279	0.000	0.205	
	0.000	0.073	0.000	0.000	0.030	0.000	0.014	
3	0	21	12	0	0	0	35	68
	13.006	13.260	0.016	18.786	7.185	2.208	195.525	
	0.000	0.309	0.176	0.000	0.000	0.000	0.515	0.040
	0.000	0.087	0.039	0.000	0.000	0.000	0.299	
	0.000	0.012	0.007	0.000	0.000	0.000	0.021	
4	0	49	0	0	46	0	37	132
	25.247	48.633	24.156	36.468	73.654	4.286	85.278	
	0.000	0.371	0.000	0.000	0.348	0.000	0.280	0.078
	0.000	0.203	0.000	0.000	0.257	0.000	0.316	
	0.000	0.029	0.000	0.000	0.027	0.000	0.022	
5	160	0	151	225	0	29	0	565
	24.961	80.381	21.919	30.420	59.702	6.190	39.023	
	0.283	0.000	0.267	0.398	0.000	0.051	0.000	0.334
	0.494	0.000	0.487	0.481	0.000	0.527	0.000	
	0.094	0.000	0.089	0.133	0.000	0.017	0.000	
6	12	13	12	111	0	9	0	157
	10.824	3.902	9.743	105.437	16.590	2.988	10.844	
	0.076	0.083	0.076	0.707	0.000	0.057	0.000	0.093
	0.037	0.054	0.039	0.237	0.000	0.164	0.000	
	0.007	0.008	0.007	0.066	0.000	0.005	0.000	
7	0	34	0	0	83	0	21	138
	26.394	10.514	25.254	38.125	321.012	4.481	13.800	
	0.000	0.246	0.000	0.000	0.601	0.000	0.152	0.081
	0.000	0.141	0.000	0.000	0.464	0.000	0.179	
	0.000	0.020	0.000	0.000	0.049	0.000	0.012	
Column Total	324	241	310	468	179	55	117	1694
	0.191	0.142	0.183	0.276	0.106	0.032	0.069	

Figure 19. Comparing Result (k=7)

mush.test.pred.k9$cluster	mush.test.k9.labels 1	2	3	4	5	6	7	8	9	Row Total
1	0	20	0	0	0	0	0	15	0	35
	5.434	530.114	5.537	9.401	2.831	3.306	4.236	65.494	1.136	
	0.000	0.571	0.000	0.000	0.000	0.000	0.000	0.429	0.000	0.021
	0.000	0.588	0.000	0.000	0.000	0.000	0.000	0.128	0.000	
	0.000	0.012	0.000	0.000	0.000	0.000	0.000	0.009	0.000	
2	0	0	72	0	61	0	0	38	0	171
	26.548	3.432	74.676	45.930	160.894	16.151	20.694	58.075	5.552	
	0.000	0.000	0.421	0.000	0.357	0.000	0.000	0.222	0.000	0.101
	0.000	0.000	0.269	0.000	0.445	0.000	0.000	0.325	0.000	
	0.000	0.000	0.043	0.000	0.036	0.000	0.000	0.022	0.000	
3	48	0	0	109	0	37	59	0	8	261
	1.380	5.238	41.292	21.582	21.108	6.185	23.796	18.027	0.027	
	0.184	0.000	0.000	0.418	0.000	0.142	0.226	0.000	0.031	0.154
	0.183	0.000	0.000	0.240	0.000	0.231	0.288	0.000	0.145	
	0.028	0.000	0.000	0.064	0.000	0.022	0.035	0.000	0.005	
4	58	14	0	17	0	0	37	0	4	130
	70.858	49.728	20.567	9.194	10.514	12.279	28.752	8.979	0.012	
	0.446	0.108	0.000	0.131	0.000	0.000	0.285	0.000	0.031	0.077
	0.221	0.412	0.000	0.037	0.000	0.000	0.180	0.000	0.073	
	0.034	0.008	0.000	0.010	0.000	0.000	0.022	0.000	0.002	
5	0	0	84	0	41	0	0	27	0	152
	23.599	3.051	149.470	40.826	67.040	14.357	18.394	25.939	4.935	
	0.000	0.000	0.553	0.000	0.270	0.000	0.000	0.178	0.000	0.090
	0.000	0.000	0.313	0.000	0.299	0.000	0.000	0.231	0.000	
	0.000	0.000	0.050	0.000	0.024	0.000	0.000	0.016	0.000	
6	0	0	112	0	35	0	0	37	0	184
	28.567	3.693	236.030	49.421	27.202	17.379	22.267	46.433	5.974	
	0.000	0.000	0.609	0.000	0.190	0.000	0.000	0.201	0.000	0.109
	0.000	0.000	0.418	0.000	0.255	0.000	0.000	0.316	0.000	
	0.000	0.000	0.066	0.000	0.021	0.000	0.000	0.022	0.000	
7	12	0	0	104	0	37	0	0	9	162
	6.877	3.251	25.629	84.085	13.102	30.772	19.604	11.189	2.660	
	0.074	0.000	0.000	0.642	0.000	0.228	0.000	0.000	0.056	0.096
	0.046	0.000	0.000	0.229	0.000	0.231	0.000	0.000	0.164	
	0.007	0.000	0.000	0.061	0.000	0.022	0.000	0.000	0.005	
8	122	0	0	85	0	29	51	0	16	303
	119.441	6.081	47.936	0.161	24.505	0.005	5.602	20.927	3.860	
	0.403	0.000	0.000	0.281	0.000	0.096	0.168	0.000	0.053	0.179
	0.464	0.000	0.000	0.187	0.000	0.181	0.249	0.000	0.291	
	0.072	0.000	0.000	0.050	0.000	0.017	0.030	0.000	0.009	
9	23	0	0	140	0	57	58	0	18	296
	11.466	5.941	46.829	46.032	23.939	30.170	13.733	20.444	7.324	
	0.078	0.000	0.000	0.473	0.000	0.193	0.196	0.000	0.061	0.175
	0.087	0.000	0.000	0.308	0.000	0.356	0.283	0.000	0.327	
	0.014	0.000	0.000	0.083	0.000	0.034	0.034	0.000	0.011	
Column Total	263	34	268	455	137	160	205	117	55	1694
	0.155	0.020	0.158	0.269	0.081	0.094	0.121	0.069	0.032	

Figure 20. Comparing Result (k=9)

The cluster counts and calculation of precision are shown in Figures 21 & 22.

- K=5 performs best according to the precision rate which equals to 22.79%;
- The precision rate almost remains the same for k=3 vs. k=9.

	mush.k3.counts	true_labels
1	238	754
2	304	529
3	1152	411

Figure 21.1 Cluster counts(k=3)

	mush.k5.counts	true_labels
1	176	631
2	267	156
3	99	34
4	641	468
5	511	405

Figure 21.2 Cluster counts(k=5)

	mush.k7.counts	true_labels
1	436	324
2	198	241
3	68	310
4	132	468
5	565	179
6	157	55
7	138	117

Figure 21.3 Cluster counts(k=7)

	mush.k9.counts	true_labels
1	35	263
2	171	34
3	261	268
4	130	455
5	152	137
6	184	160
7	162	205
8	303	117
9	296	55

Figure 21.4 Cluster counts(k=9)

K	F1_score
3	0.0354
5	0.2279
7	0.1877
9	0.0449

Figure 21.5 Accuracy Scores

Part 5 Conclusion

After comparing the precision results for different numbers of clusters, I found that the optimal classification number should be 5.

K-Means and KNN are both machine learning algorithms. However, KNN is an algorithm used for classification, while K-Means is an unsupervised algorithm used for clustering. I need to generate labels by K-Means method at first, then apply these labels to data set for KNN analysis.

This project has taught me multiple methods as to how to deal with categorical data. I also learned about different normalization methods. Finally, I developed a strong understanding of K-Means clustering, including experimenting between different k values, investigating closely related cases, and optimizing the number of clusters.

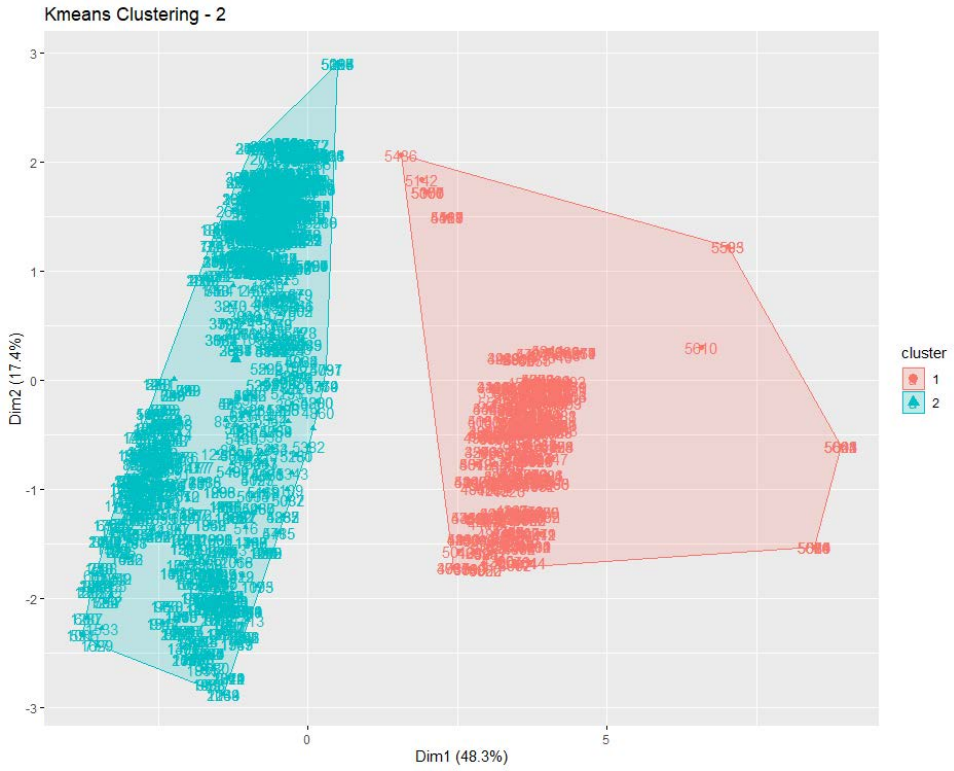
Appendix A

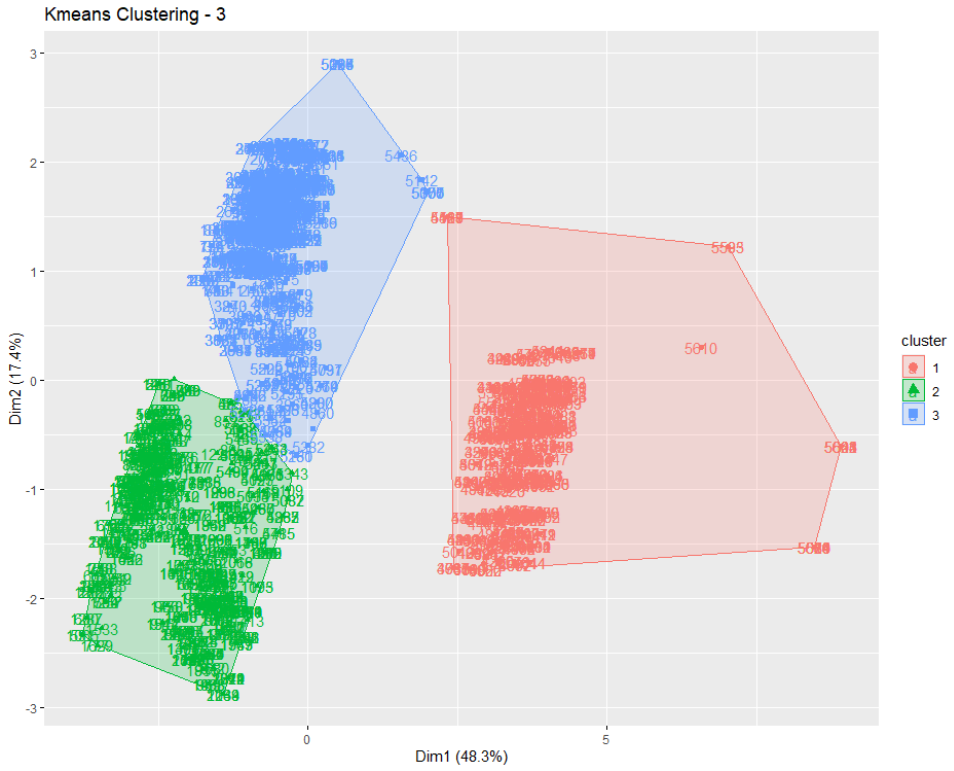
Attribute Information

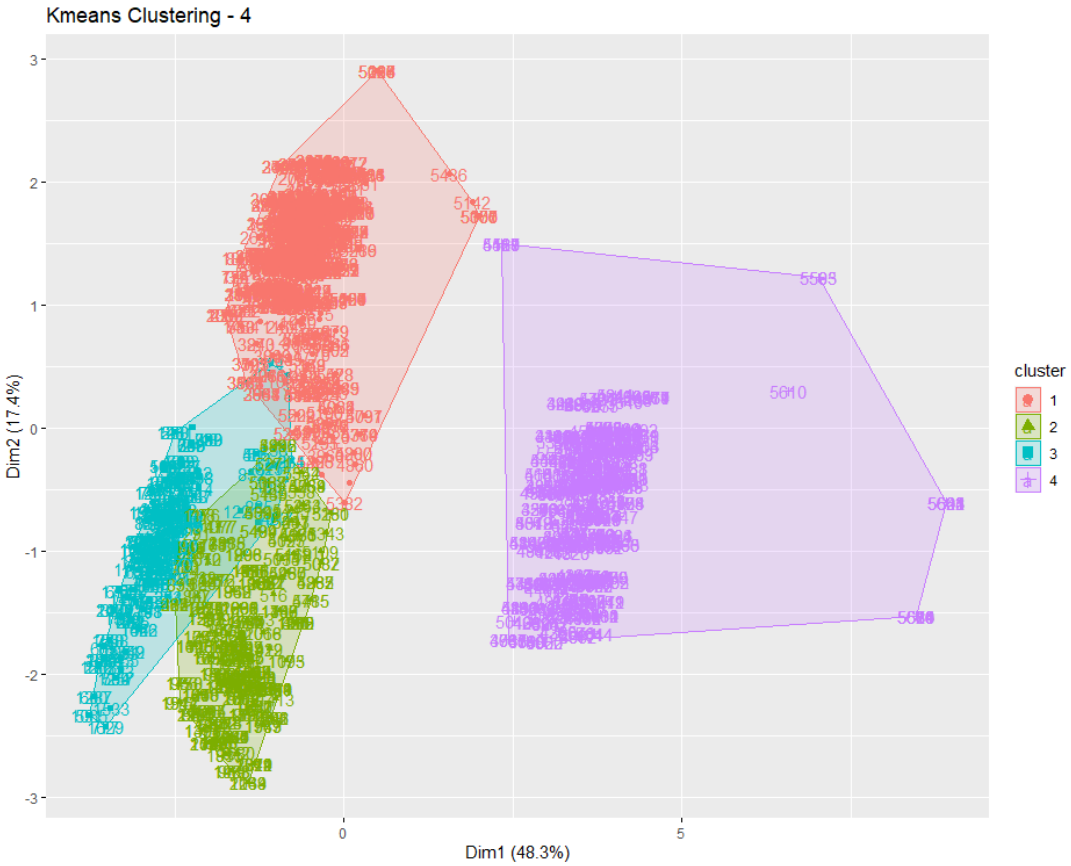
-1. cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
-2. cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
-3. cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
-4. bruises?: bruises=t, no=f
-5. odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
-6. gill-attachment: attached=a, descending=d, free=f, notched=n
-7. gill-spacing: close=c, crowded=w, distant=d
-8. gill-size: broad=b, narrow=n
-9. gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
-10. stalk-shape: enlarging=e, tapering=t
-11. stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
-12. stalk-surface-above-ring: fibrous=f, scaly=y, silky=k, smooth=s
-13. stalk-surface-below-ring: fibrous=f, scaly=y, silky=k, smooth=s
-14. stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
-15. stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
-16. veil-type: partial=p, universal=u
-17. veil-color: brown=n, orange=o, white=w, yellow=y
-18. ring-number: none=n, one=o, two=t
-19. ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
-20. spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
-21. population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
-22. habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

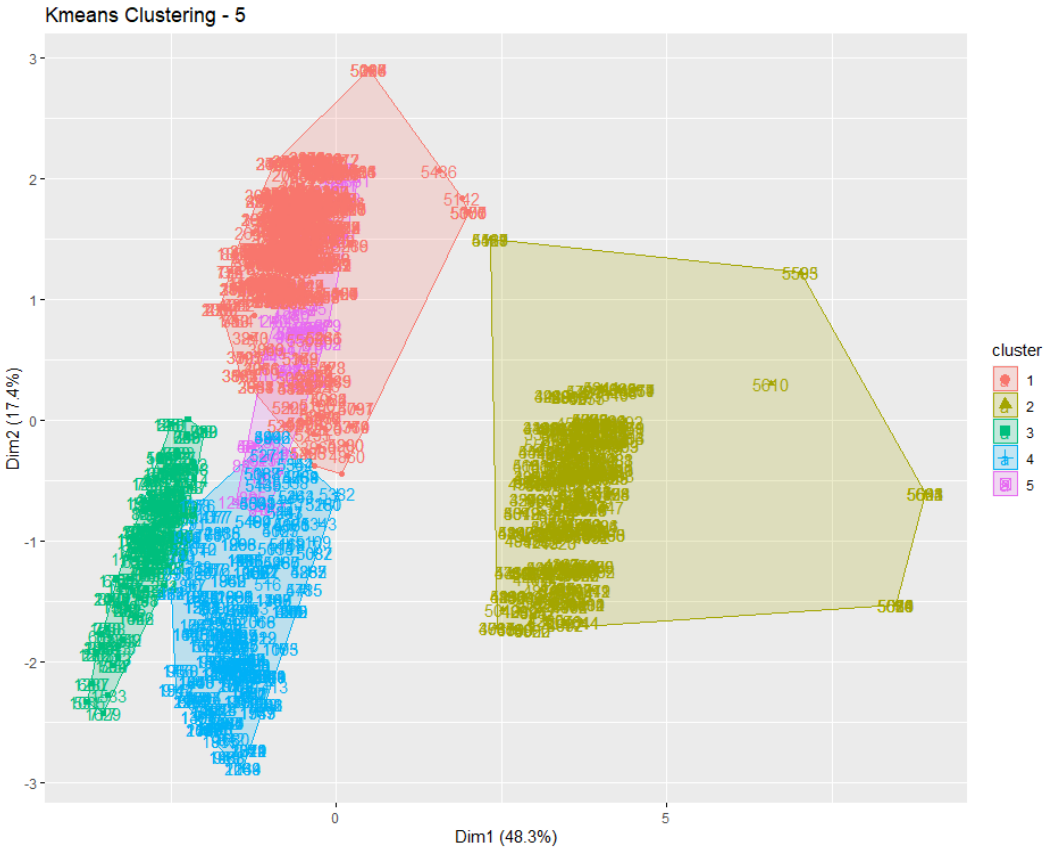
Appendix B

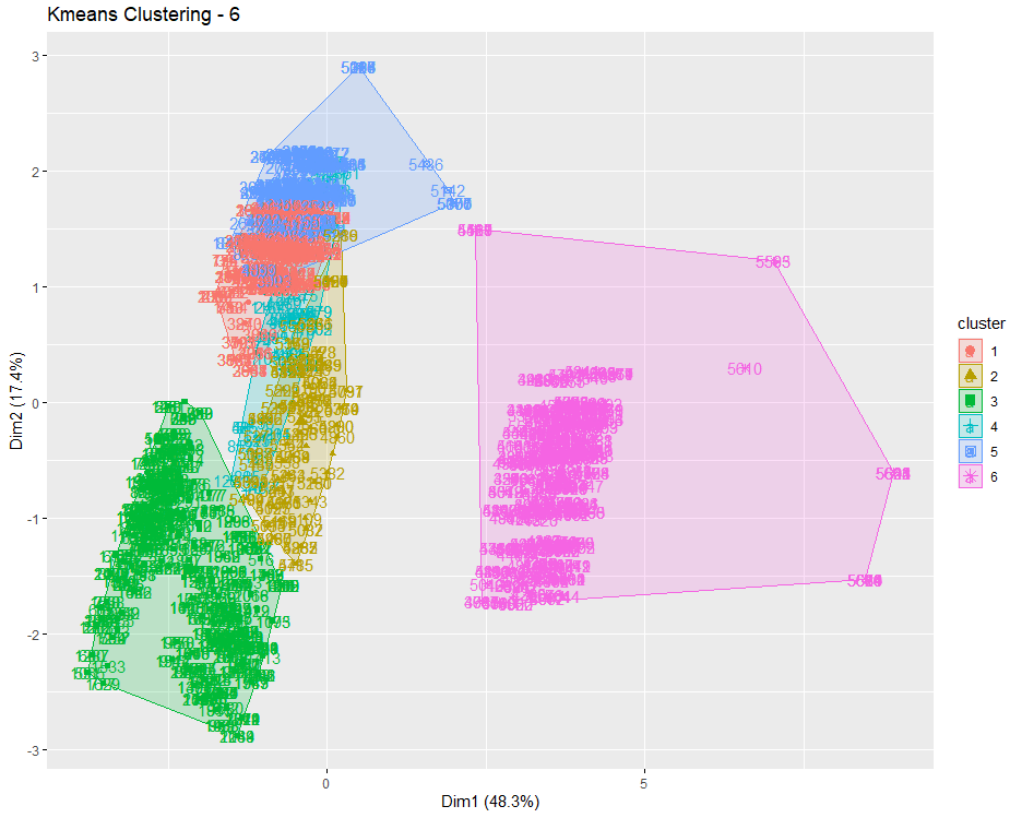
Kmeans Clustering Plot

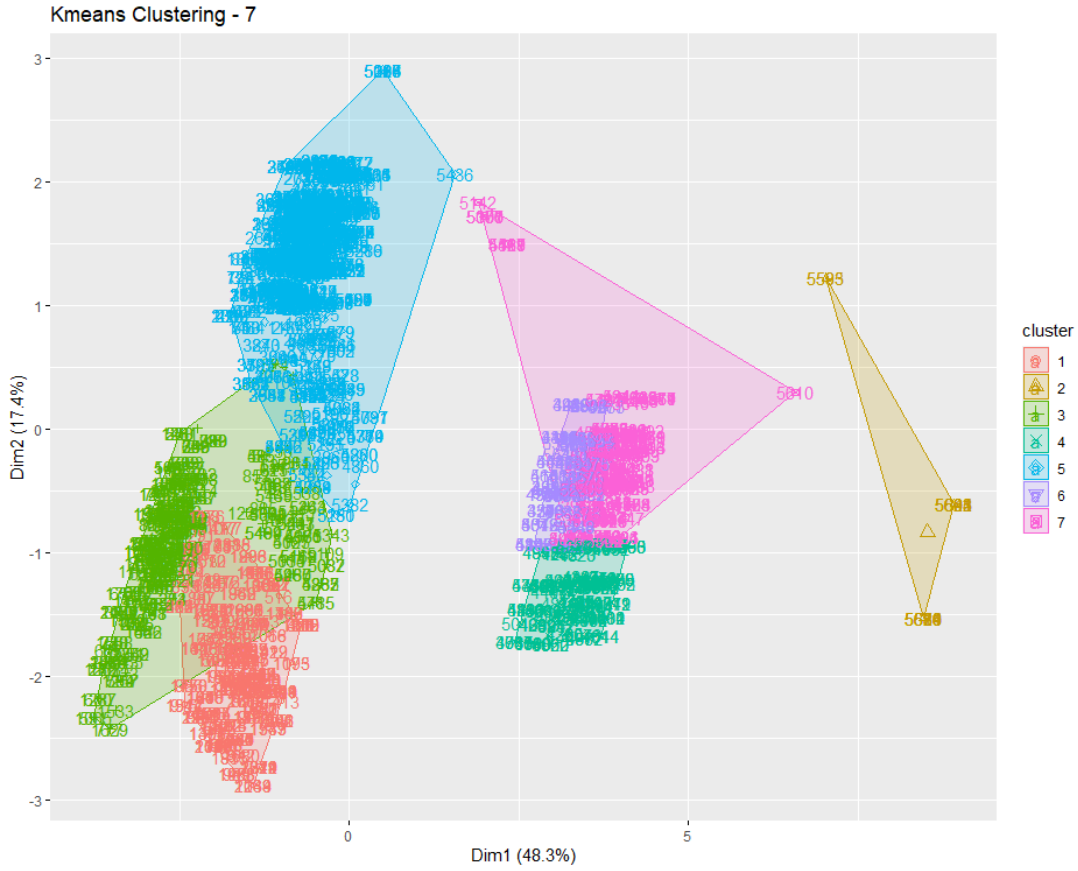





Kmeans Clustering - 4
Dim2 (17.4%)
Dim1 (48.3%)
cluster
1
2
3
4


Kmeans Clustering - 5
Dim2 (17.4%)
Dim1 (48.3%)
cluster
1
2
3
4
5


Kmeans Clustering - 6
Dim2 (17.4%)
Dim1 (48.3%)
cluster
1
2
3
4
5
6


Kmeans Clustering - 7
Dim2 (17.4%)
Dim1 (48.3%)
cluster
1
2
3
4
5
6
7