Excercise 4

Xiaohan Sun / Liyuan Zhang / Evelyn Cheng

2021/5/6

Problem 1: Clustering and PCA

Run both PCA and a clustering algorithm of your choice on the 11 chemical properties (or suitable transformations thereof) and summarize your results.

PCA MODEL

```
PCA = prcomp(test_features, nfactors=3, scale = TRUE, scores=TRUE)
print(summary(PCA))
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
PC7
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930
0.72330
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521
0.04756
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253
0.90009
##                            PC8     PC9   PC10    PC11
## Standard deviation     0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.94568 0.97632 0.9970 1.00000
```

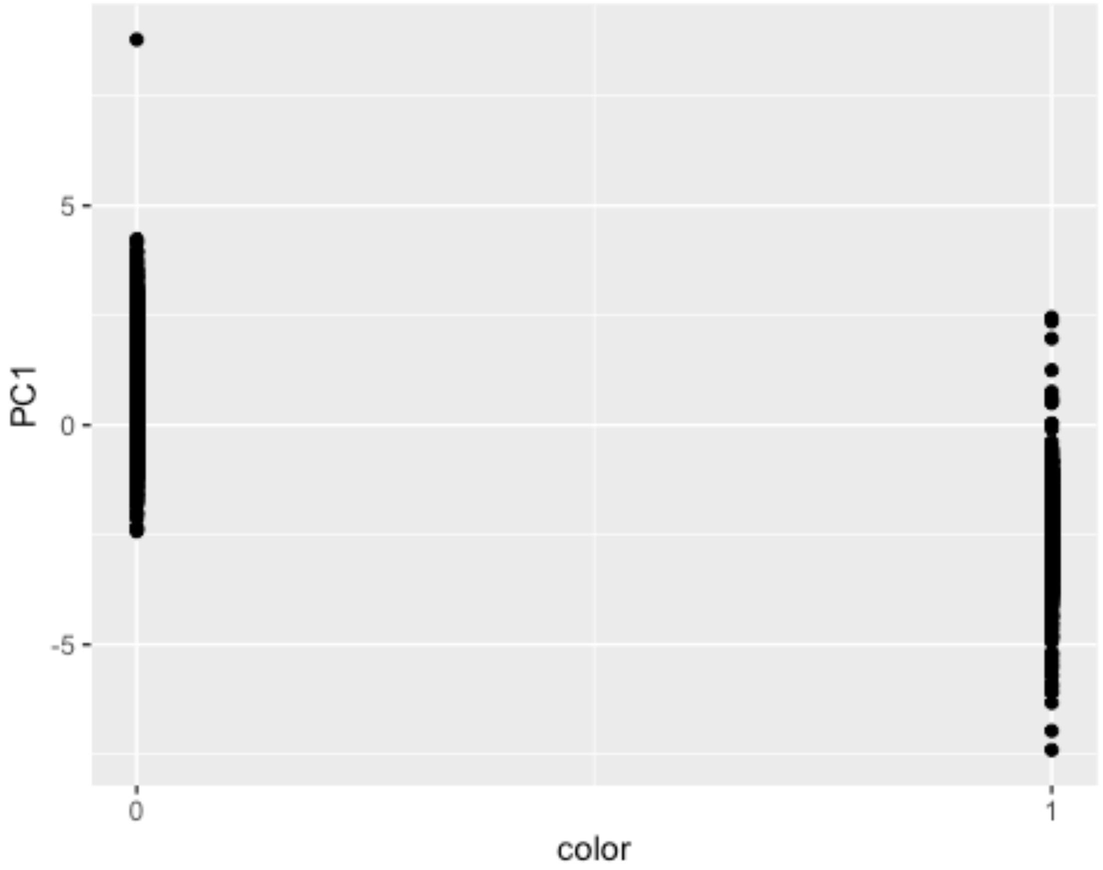
```
PCA_model = predict(PCA,test_features)
PCA_model = as.data.frame(PCA_model)
PCA_model$color = wine$color
PCA_model$quality = wine$quality
PCA_model1 <- lm(quality ~ PC1, data=PCA_model)
PCA_model2 <- glm(color ~ PC1, family = binomial(), data=PCA_model)
PCA1 = predict(PCA_model1)
PCA2 = predict(PCA_model2,type = "response")
print(rmse(PCA_model$quality,PCA1))
```

```
## [1] 0.8706529
```

```
print(f1Score(PCA_model$color,PCA2))
```

```
## [1] 0.9538267
```

```
PCA_graph = ggplot(data = PCA_model) +
  geom_point(aes(x = color, y = PC1))+
  scale_x_continuous( breaks=seq(0,1,1))
PCA_graph
```



Above are the PCA model, RMSE is 0.8706529 and f1Score is 0.9538267. We only utilize one chemical property instead of 11 properties.

KMEANS MODEL

```
set.seed(100)
kfeatures = kmeans(test_features, centers = 2)
wine$kcluster = kfeatures$cluster
kmodel1 <- lm(quality ~ kcluster, data = wine)
kmodel2 <- glm(color ~ kcluster, family = binomial(), data = wine)
k1 = predict(kmodel1)
k2 = predict(kmodel2,type = "response")
print(rmse(wine$quality,k1))

## [1] 0.8731613

print(f1Score(wine$color,k2))

## [1] 0.6870887
```

Above are the kmeans model, RMSE is 0.8731613 and f1Score is 0.6870887. We only utilize one class instead of 11 properties.

Benchmark MODEL

```
model1 <- lm(quality ~ .-color-quality, data=wine)
model2 <- glm(color ~ .-color-quality, data=wine, family = binomial())
wine1 = predict(model1)
wine2 = predict(model2,type = "response")
print(rmse(wine$quality, wine1))

## [1] 0.7330134

print(f1Score(wine$color, wine2))

## [1] 0.989355
```

Above are the Benchmark model, RMSE is 0.7346533 and f1Score is 0.9896714. We utilize 11 chemical properties.

Which dimensionality reduction technique makes more sense to you for this data?

F1-score is a calculation result that comprehensively considers the precision and recall of the model. The larger the F1-score, the higher the quality of the model. PMSE represents the sample standard deviation of the difference between the predicted value and the observed value, the smaller the better. So that compared with PCA model and kmeans model, F1-score is higher and RMSE is lower in PCA model. I prefer PCA model.

Convince yourself (and me) that your chosen method is easily capable of distinguishing the reds from the whites, using only the "unsupervised" information contained in the data on chemical properties. Does your unsupervised technique also seem capable of distinguishing the higher from the lower quality wines?

PCA model is easily capable of distinguishing the reds from the whites. It seems not capable of distinguishing the higher from the lower quality wines.

Problem 2: Market segmentation

In this report, we use K-means clustering to define "market segment".

Pre-process the data

Before we do K-means clustering, we need to remove some labels that are meaningless to our analysis, and then center and scale the data. Here are these labels:

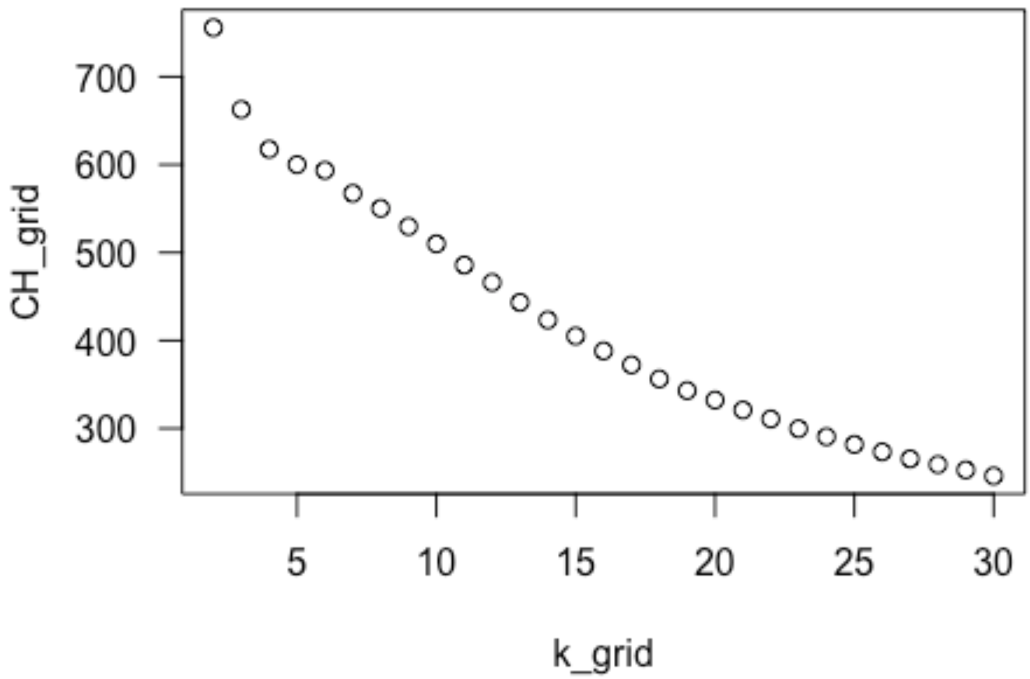
Labels	Definition
spam	i.e. unsolicited advertising
adult	posts that are pornographic or otherwise explicit

uncategorized	posts that don't fit at all into any of the listed interest categories
chatter	sometimes has same definition as "uncategorized" label

Define "Market Segment"

After cleaning the data, we can build a model to find interesting market segments that appear to stand out in their social-media audience. We have tried plenty of models to define the market segment, like hierarchical clustering, a principal component,etc. Finally, we decide to use k-means clustering to build the model.

For choosing K, we use CH index to choose the best K. As the following graph shows, the K-means clustering with 4 clusters might be the "best" one to define market segment for NutrientH20.



Here are the 4 clusters:

(1) cluster 1

```
## health_nutrition          cooking    photo_sharing personal_fitness
##         7.635319         6.332376         4.590093         4.147164
```

(2) cluster 2

```
##    photo_sharing   current_events      college_uni health_nutrition
##         2.200321         1.428371         1.402853         1.336347
```

(3) cluster 3

```
## sports_fandom       religion          food     parenting
##      5.877039       5.262233      4.534504      4.027604
```

(4) cluster 4

```
##      politics         travel          news photo_sharing
##      8.939860       5.618182      5.300699      2.531469
```

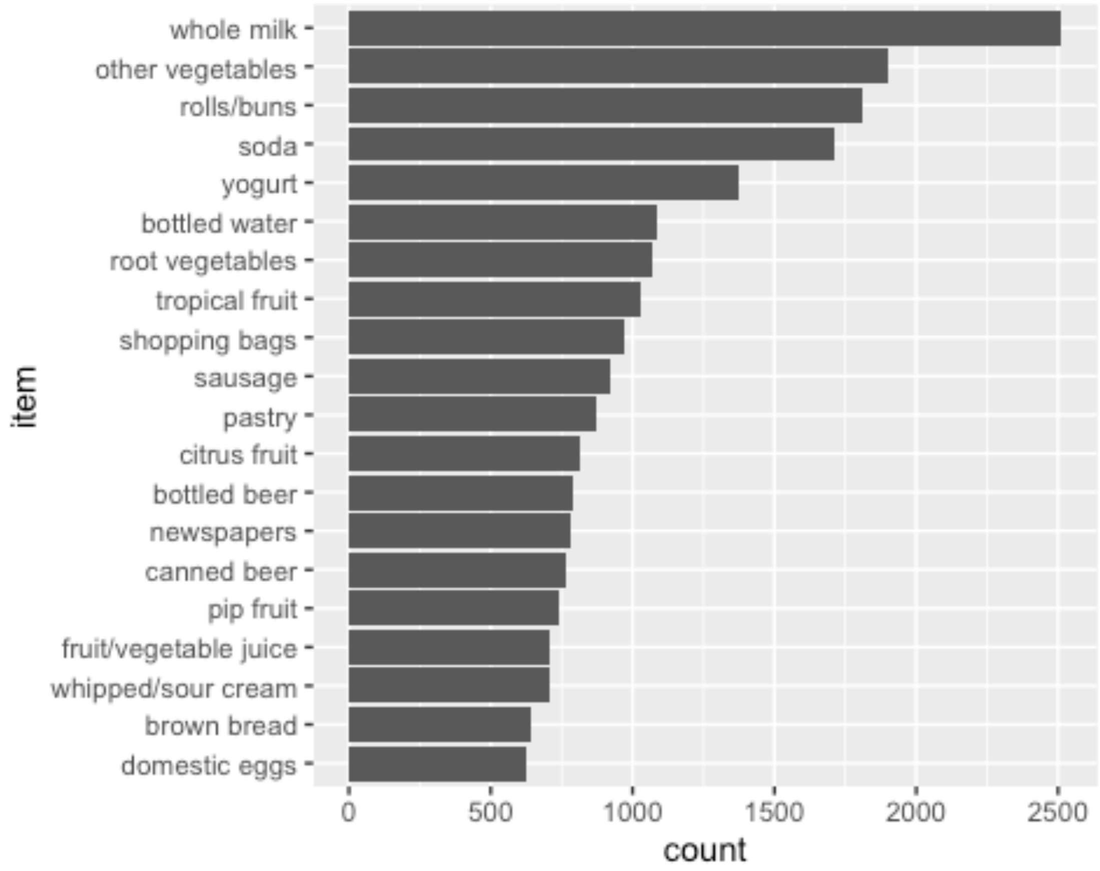
In conclusion, the four groups above represents some interesting market segments in their social-media audience, such as young people who love photo sharing, people who like travel and Social dynamics, people who pay more attention to keep fit, and people who are sports fans and also might have religious beliefs.

The advertising firm could make advertisement based on the characteristic of these four groups.

Problem 3: Association rules for grocery purchases

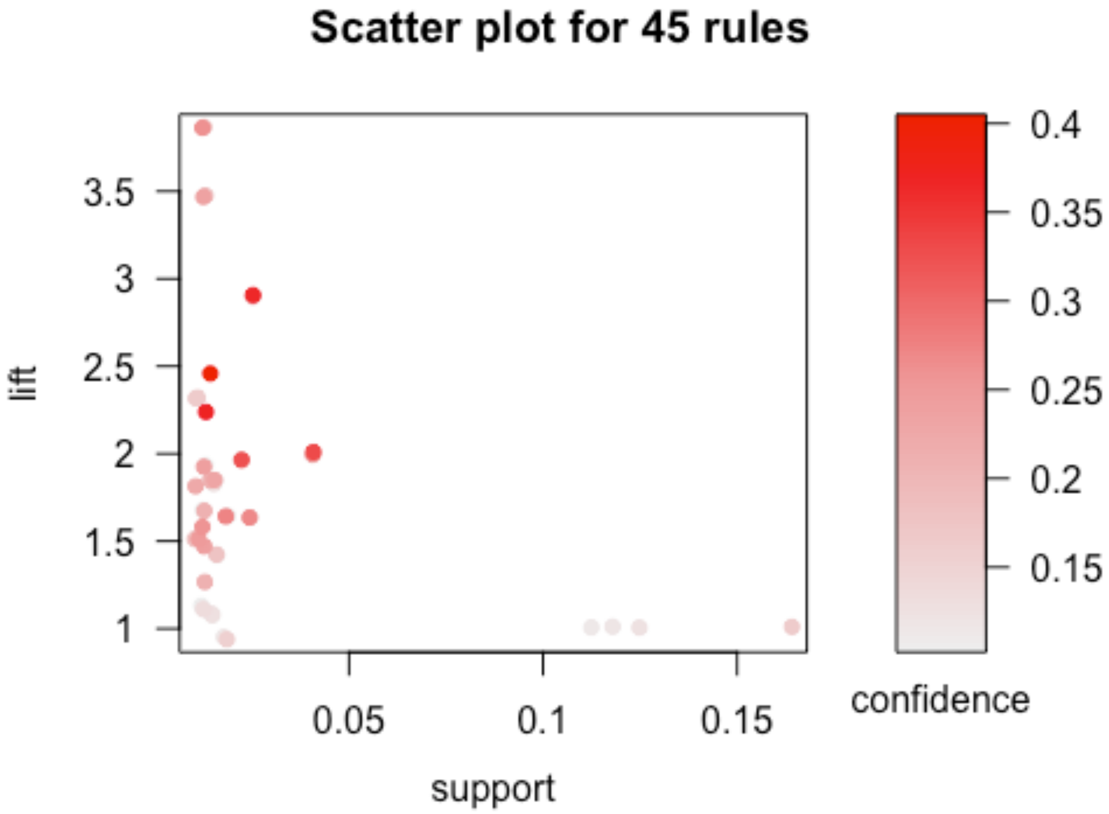
In order to find the association rules for the purchase of goods by grocery store customers. Firstly, organizing the data by using the data about grocery store purchases.

After processing the data, the top 20 items of this data in the following figure. As you can see, whole milk is the most popular product.



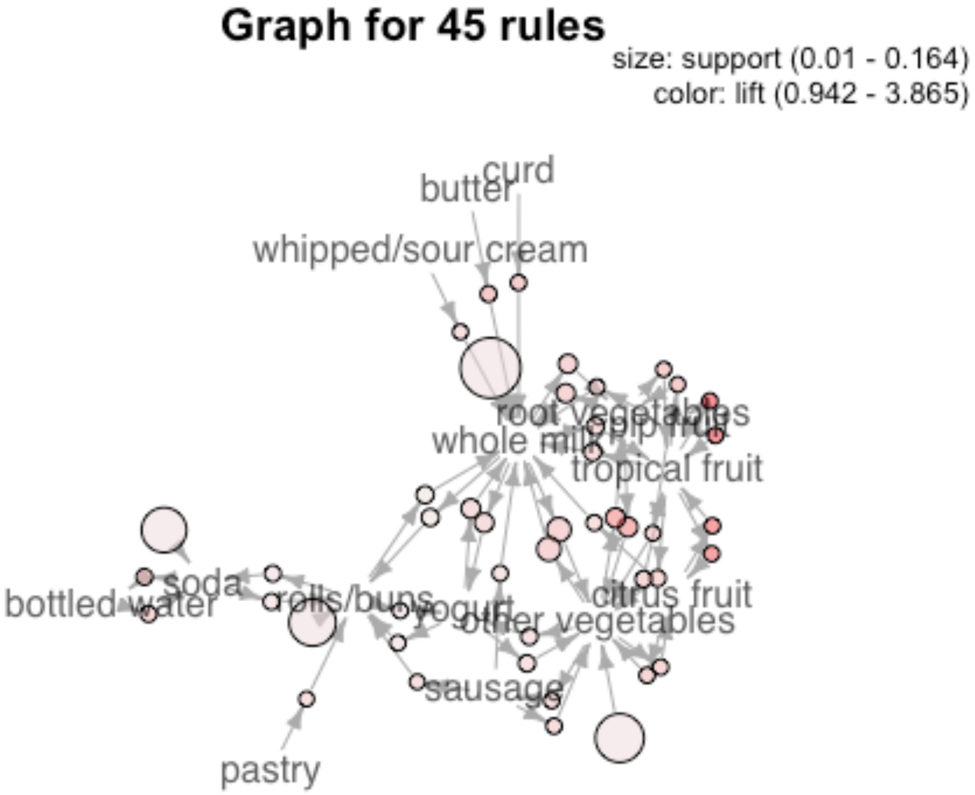
After having a general understanding of the data, the association rules for grocery purchases are obtained through the calculation of the Apriori method.

After having a general understanding of the data, the Apriori method is used to set the conditions of support=0.01, confidence=0.1 and maxlen=2 to calculate the association rules for grocery shopping. At the same time, the association rules are displayed in the form of a scatter plot.



Because the above-mentioned association rules are cumbersome and difficult to see clearly. In order to more easily understand the association rules of the grocery store products, we will reduce all the above-mentioned association rules to obtain a clearer subset of the association rules.

By restricting the two conditions of confidence which is greater than 1% and support which is greater than 0.5%, 49 rules were selected from the association rules to form a subset.



Problem 4: Author attribution

The purpose of this exercise is to use the articles of 50 different authors in c50train to build a model, so as to be able to predict the identity of the author in the test data set through the articles.

In order to study this issue, we first need to process the text. For the training set and test set, we use the sparse matrix method at the same time. By using a for loop on the data, the author name and text path of each file in the data set are obtained. Finally, the Corpus equation is used to form two different corpora. After obtaining the corpus, the training data and test data need to be processed separately. We set all letters to lowercase, delete numbers, delete punctuation marks and extra spaces.

After the data processing is completed, we decided to use the two methods of Naive Bayes and Random Forest to complete the model establishment, and use the test data to predict its accuracy, compare the two models, and select a better method.

```
author.predict.correct

##          author_names correct per.correct
## 1       AaronPressman      31        0.62
## 41       RogerFillion      10        0.20
## 32         MartinWolk       3        0.06
## 47    TheresePoletti       3        0.06
## 49        ToddNissen       3        0.06
## 10       EricAuchard       2        0.04
## 42       SamuelPerry       2        0.04
## 3     AlexanderSmith       1        0.02
## 6        BradDorfman       1        0.02
## 8        DavidLawder       1        0.02
## 11    FumikoFujisaki       1        0.02
## 26 KouroshKarimkhany       1        0.02
## 34     MichaelConnor       1        0.02
## 2         AlanCrosby       0        0.00
## 4    BenjaminKangLim       0        0.00
## 5      BernardHickey       0        0.00
## 7   DarrenSchuettler       0        0.00
## 9      EdnaFernandes       0        0.00
## 12    GrahamEarnshaw       0        0.00
## 13  HeatherScoffield       0        0.00
## 14        JanLopatka       0        0.00
## 15     JaneMacartney       0        0.00
## 16      JimGilchrist       0        0.00
## 17    JoWinterbottom       0        0.00
## 18          JoeOrtiz       0        0.00
## 19      JohnMastrini       0        0.00
## 20      JonathanBirt       0        0.00
## 21       KarlPenhaul       0        0.00
## 22         KeithWeir       0        0.00
## 23    KevinDrawbaugh       0        0.00
## 24     KevinMorrison       0        0.00
## 25     KirstinRidley       0        0.00
## 27         LydiaZajc       0        0.00
## 28    LynneO'Donnell       0        0.00
## 29   LynnleyBrowning       0        0.00
## 30   MarcelMichelson       0        0.00
## 31      MarkBendeich       0        0.00
## 33      MatthewBunce       0        0.00
## 35        MureDickie       0        0.00
## 36         NickLouth       0        0.00
## 37   PatriciaCommins       0        0.00
## 38     PeterHumphrey       0        0.00
## 39        PierreTran       0        0.00
## 40        RobinSidel       0        0.00
## 43      SarahDavison       0        0.00
## 44       ScottHillis       0        0.00
## 45       SimonCowell       0        0.00
## 46          TanEeLyn       0        0.00
## 48        TimFarrand       0        0.00
## 50      WilliamKazer       0        0.00

sum(author.predict.correct$correct)/nrow(X_test)
```

```
## [1] 0.024
```

When we use the Naive Bayes model to make predictions, the accuracy obtained is around 3%.

In order to be able to get a better prediction model, we also used the random forest method to predict the model.

```
accuracy

## [1] 0.5712
```

The accuracy of the random forest is about 60%, which is much better than Naive bayes method.