Food Composition Data, Clustering, and Classification

Beksultan Tuleev

## Introduction

Human nutrition always has been considered one of the pivotal issues because essential nutrients in food provide the necessary support for a healthy life. Poor nutrition often leads to numerous health diseases; therefore, a better understanding of nutrition science gives a better relation to maintenance, reproduction, and growth in general. To understand food nutrients, science digs in food composition data that gives information about values for each component in food, such as protein, fat, carbohydrates, minerals, vitamins, etc. Knowledge about the chemical composition of foods is crucial. For instance, it is pivotal in the dietary treatment of disease, in the treatment of allergies. The United States Department of Agriculture (USDA) National Nutrient Database (NND) for Standard Reference (SR) is the biggest source of data in food composition in the United States (US) that provides its databases for the public and private sectors. Food composition data has been published in an electronic copy on the Nutrient Data Laboratory (NDL) web page since 1992. The latest release has a number 28 (SR28)[[1]](#footnote-0) that contains a composition data for food groups and nutrients ever published before and completely replaces previous SR versions. SR28 contains 8790 units that have value for specific nutrients, such as protein, sugar, zinc, selenium, etc. Additionally, the database was divided into 25 groups. The main goal of this paper is to make a classification and clustering of each nutrition component using different methods and compare their performance.

## Literature Review

Research study made by Daniel Granato, Janio S. Santos, Graziela B. Escher, Bruno L. Ferreira, and Rubén M. Maggio “Use of principal component analysis (PCA) and hierarchical cluster analysis(HCA) for a multivariate association between bioactive compounds and functional properties in foods: A critical perspective”[[2]](#footnote-1) is a reference article for this research. PCA and HCA are mostly used in the analysis to determine similarities and unclear patterns in samples, especially in food chemistry analysis, because final results are easy to interpret. Additionally, variables in food chemistry tend to correlate with each other, and the PCA method does a great job even with this kind of datasets. As a result, authors get a decent classification.

Another fascinating research study was done by Lucia Giansante, Daria Di Vincenzo, and Giorgio Bianchi, ‘Classification of monovarietal Italian olive oils by unsupervised (PCA) and supervised (LDA) chemometrics’[[3]](#footnote-2). By the title of the research it is clear that in addition to PCA, Linear Discriminant Analysis (LDA) was done to perform clustering on different compounds of oils, such as hexacosanol, cycloartenol, methylene cycloartenol, oleic, linoleic. and etc. Since LDA is a supervised technique, the date was divided on training and testing set, where it has 57 and 19 oil samples respectively. This approach would be followed in this paper as well.

Next research that was done by Steffen Oppel and Falk Huettmann ‘Using a Random Forest Model and Public Data to Predict the Distribution of Prey for Marine Wildlife Management’[[4]](#footnote-3) introduces another classification method for this paper - Random Forest. The study field in that research is different from the previous ones and covers the food chain for marine wildlife and the prediction of the distribution of benthic biomass across the entire Bering Sea. There were used environmental variables, like chlorophyll a content, hydrography, water temperature, sea ice scouring, and sediment grain size. Random Forest was used to classify and determine the most important places for marine wildlife to have a sustainable food supply. The same method would be used in this research paper.

Research by Seunghee WIe, Carow W. Shankin, Kyung-Eun Lee, ‘A decision tree for selecting the most cost-effective waste disposal strategy in foodservice operations’[[5]](#footnote-4) is introducing another classification method that is used in the current paper - Decision tree. The purpose of that paper was in determining the costs of disposal strategies for wastes in foodservice operations and to define the most cost-effective disposal strategy for food services.

The last referred paper is written by T. Shanthanam and Padmavathi M.S, ‘Application of K-Means and Genetic Algorithms for Dimension Reduction by Integrating SVM for Diabetes Diagnosis’[[6]](#footnote-5) that introduces two methods for clustering and classification: Support Vector Machine and K-Means. Although authors use both methods cooperatively to reduce nosiness in data with k-means and classify diabetes cases with Support Vector Machine (SVM), in the current paper two methods would be used separately to see the performance of each of them.

## Data Description and Data Cleaning

Data was taken from the Standard Reference (SR28) file that contains 8790 observations. Data file offers for each observation the information of composition.

**library**(caret)

data <- **read.csv**('https://drive.google.com/u/1/uc?id=15x9KDfU2x8lkiBzh2XWNVccLArS-MIFr&export=download')  
**row.names**(data) <- data**$**NDB\_No  
data**$**NDB\_No <- NULL  
data**$**X <- NULL  
correlation\_dataframe = **cor**(data[**-length**(data)])  
hc = **sort**(**findCorrelation**(correlation\_dataframe, cutoff=0.75))   
data = data[,**-c**(hc)]

The table below shows all columns in the SR28 file that have been used in analysis. Detailed description of each variable is available in the source webpage[[7]](#footnote-6).

## [1] "Protein\_.g." "Lipid\_Tot\_.g." "Carbohydrt\_.g."   
## [4] "Fiber\_TD\_.g." "Sugar\_Tot\_.g." "Calcium\_.mg."   
## [7] "Iron\_.mg." "Magnesium\_.mg." "Phosphorus\_.mg."   
## [10] "Potassium\_.mg." "Sodium\_.mg." "Zinc\_.mg."   
## [13] "Copper\_mg." "Manganese\_.mg." "Selenium\_.Âµg."   
## [16] "Vit\_C\_.mg." "Thiamin\_.mg." "Riboflavin\_.mg."   
## [19] "Niacin\_.mg." "Panto\_Acid\_mg." "Vit\_B6\_.mg."   
## [22] "Folic\_Acid\_.Âµg." "Food\_Folate\_.Âµg." "Choline\_Tot\_..mg."  
## [25] "Vit\_B12\_.Âµg." "Vit\_A\_IU" "Retinol\_.Âµg."   
## [28] "Alpha\_Carot\_.Âµg." "Beta\_Carot\_.Âµg." "Beta\_Crypt\_.Âµg."   
## [31] "Lycopene\_.Âµg." "Lut.Zea\_..Âµg." "Vit\_E\_.mg."   
## [34] "Vit\_D\_Âµg" "Vit\_K\_.Âµg." "FA\_Poly\_.g."   
## [37] "Cholestrl\_.mg." "FoodGroup"

For the supervised learning methods, the dataset was divided into training and testing sets with a ratio of 75/25.

**library**(caTools)

**set.seed**(123)  
split = **sample.split**(data**$**FoodGroup, SplitRatio = 0.75)  
training\_data = **subset**(data, split **==** T)  
testing\_data = **subset**(data, split **==** F)

## Methodology description

The main strategy applied to the current dataset is classification and clustering, therefore few methods were applied to satisfy the targeted goal. Since the shape of the dataset is big enough, dimension reduction methods such as Linear Discriminant Analysis (LDA) and Principal Component Analysis(PCA) were applied to be able to project and classify and cluster components on the lower, 2-d plots. To find which components have a bigger influence on diversification, loading scores in PCA were used. Another fascinating method to find the most important components is a Random forest, and that is the next method used in the analysis. Random forest is an improved version of the Decision tree. But how Decision tree stand out in comparison with Random forest? To find an answer, a Decision tree was also added to the methodology. PCA is an unsupervised clustering method, but to test its accuracy Multinomial Logistic Regression (MLR) was used, and dependent variables for MLR are first Principal Components (PCs) with the highest variations obtained earlier in PCA. To compare PCA with another unsupervised clustering method, k-means clustering was used in the analysis. Since we have included LDA and MLR methods, which assume that categorical variables, in our case it is a Food Group, is normally distributed, it would be relevant to compare results with a method that does not make any assumptions on food group distribution. This method is the Support Vector Machine (SVM).

The final list of applied methods: LDA, PCA, Random Forest, Decision Tree, MLR, k-means, and SVM. In addition, all accuracy rates will be calculated and compared in one table.

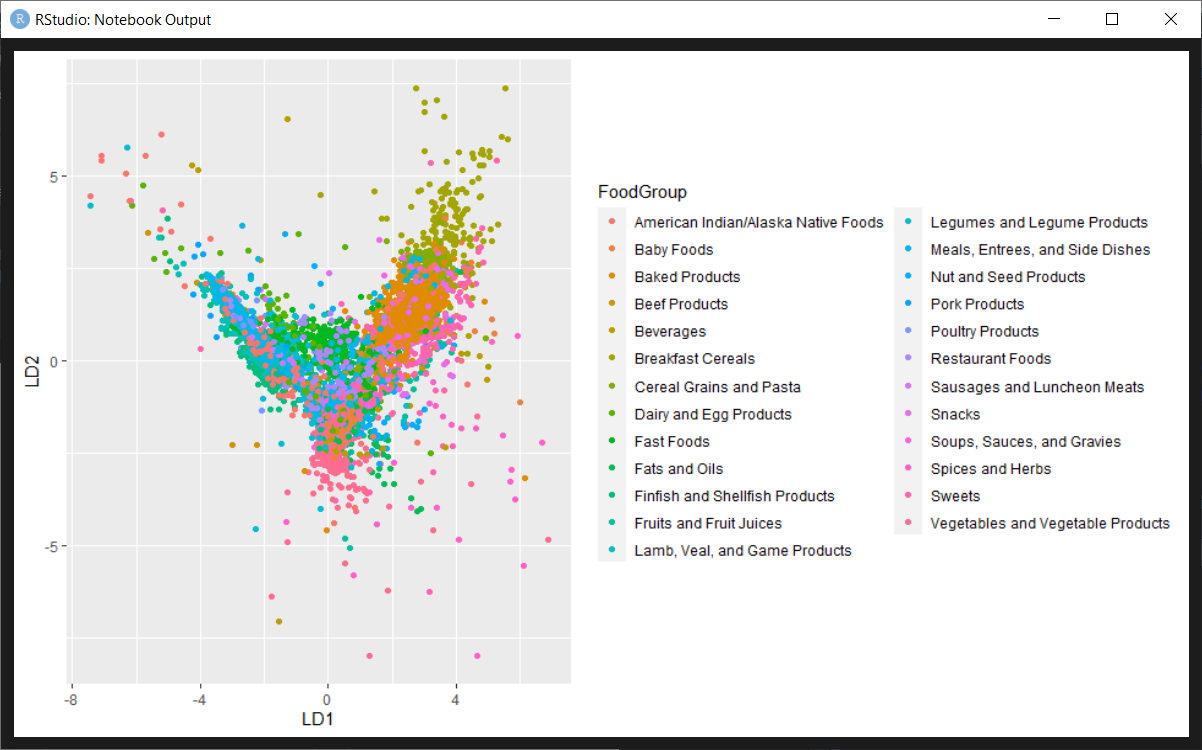
***Which method in dimensionality reduction will provide better classification/clustering and project on the two-dimensional graph?***

We are going to compare supervised LDA and unsupervised PCA methods.

### LDA

Linear Discriminant Analysis (LDA) is a statistical method similar to PCA in terms of dimensionality reduction, but very different in terms of calculation approach. First of all, it is a supervised learning method unlike PCA, LDA seeks features, which separate classes. LDA is widely used in the Machine Learning field since it is a powerful classification tool with robust performance. In this research, we are going to compute LDA along with its cross-validation method and compare their performance.

**library**(MASS)  
**library**(ggplot2)  
lda\_model <- **lda**(FoodGroup**~**., data = training\_data)  
  
*#prediction on testing set*  
prediction\_lda\_test <- **predict**(lda\_model, newdata = testing\_data)  
lda\_accuracy <- **mean**(prediction\_lda\_test**$**class**==**testing\_data**$**FoodGroup)  
lda.data <- **cbind**(training\_data, **predict**(lda\_model)**$**x)  
**ggplot**(lda.data, **aes**(LD1, LD2)) **+**  
 **geom\_point**(**aes**(color = FoodGroup))



As we can observe in the results, the accuracy rate of LDA on the testing set is 58 %. Although separation on the graph is not crystal clear, LDA was capable to separate and plot classes on lower dimensions.

***How accurate LDA would be with Cross-Validation method compared to regular LDA?***

### LDA Cross-Validation

Cross-Validation(CV) was included in this analysis to compare accuracy with regular LDA. CV was done on the training set and validated on the testing set.

*#lda\_cross-val*  
lda\_model\_cross\_val <- **lda**(FoodGroup**~**., data = training\_data,  
 trControl = **trainControl**(method = 'cv',  
 number = 10,  
 verboseIter = TRUE))  
*#prediction of test*  
lda\_model\_cross\_val\_predict = **predict**(lda\_model\_cross\_val, newdata = testing\_data)  
lda\_cv\_accuracy <- **mean**(lda\_model\_cross\_val\_predict**$**class**==**testing\_data**$**FoodGroup)

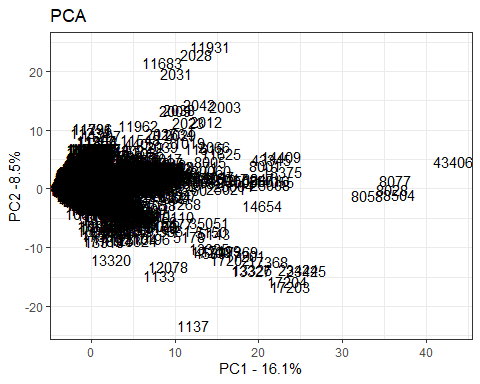
The accuracy rate of LDA CV was identical to regular LDA.

### Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method of reducing the dimensionality of the variables space without loss of important information that uses an orthogonal transformation to separate correlated and uncorrelated observations and captures its variability. In other words, PCA is a coordinate transformation, where data is plotted on a two-dimensional graph. More observations lead to an increase in graph dimensions but since it is impossible to plot a graph with four and more axes, PCA manages to put all observations on a simple two-dimensional graph with the direction of most variation and second most variation axes. It turns out that a set of other axes are related to eigenvalues and eigenvectors of the covariance matrix (covariance of dataset matrix). Each eigenvector is a vector unit that points in the direction of a coordinate axis that was impossible to plot along with previous ones. The axis with the highest eigenvalue score is the axis that explains most variation, and it is usually Principal Component 1 (PC1) and Principal component 2 (PC2).

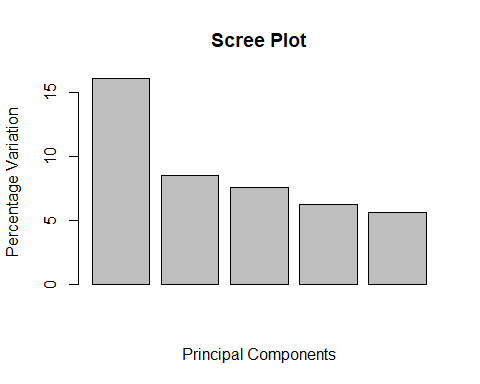
pca <- **prcomp**(**data.matrix**(data[**-length**(data)]), scale = F)  
pca.var <- pca**$**sdev**^**2 *#calculate variation in the data for each PC. if is more than 1 it explains better*  
pca.var.per <- **round**(pca.var**/sum**(pca.var)**\***100, 1) *#percentage of each variation for PC*  
cumvar <- **sum**(pca.var.per[1**:**5]) *#first 5 PCs explanation share*

**library**(ggplot2)  
pca.data <- **data.frame**(Sample= **rownames**(pca**$**x),  
 X= pca**$**x[,1],  
 Y = pca**$**x[,2])  
**ggplot**(data=pca.data, **aes**(x=X, y=Y, label=Sample))**+**  
 **geom\_text**() **+**  
 **xlab**(**paste**('PC1 - ', pca.var.per[1], '%', sep=''))**+**  
 **ylab**(**paste**('PC2 -', pca.var.per[2], '%', sep=''))**+theme\_bw**()**+**  
 **ggtitle**('PCA')



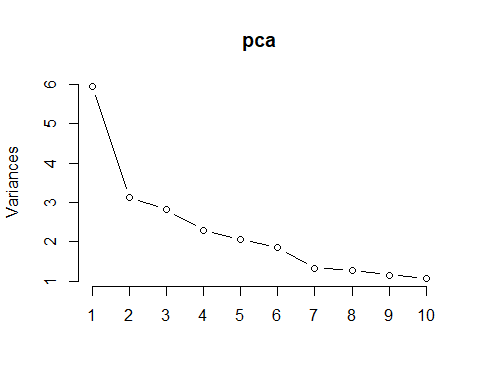
The plot shows the data distribution on the two-dimensional graph, where PC1 and PC2 are axis of it.

**barplot**(pca.var.per[1**:**5], main = 'Scree Plot', xlab = 'Principal Components', ylab = 'Percentage Variation') *#plotting the PC percentage*



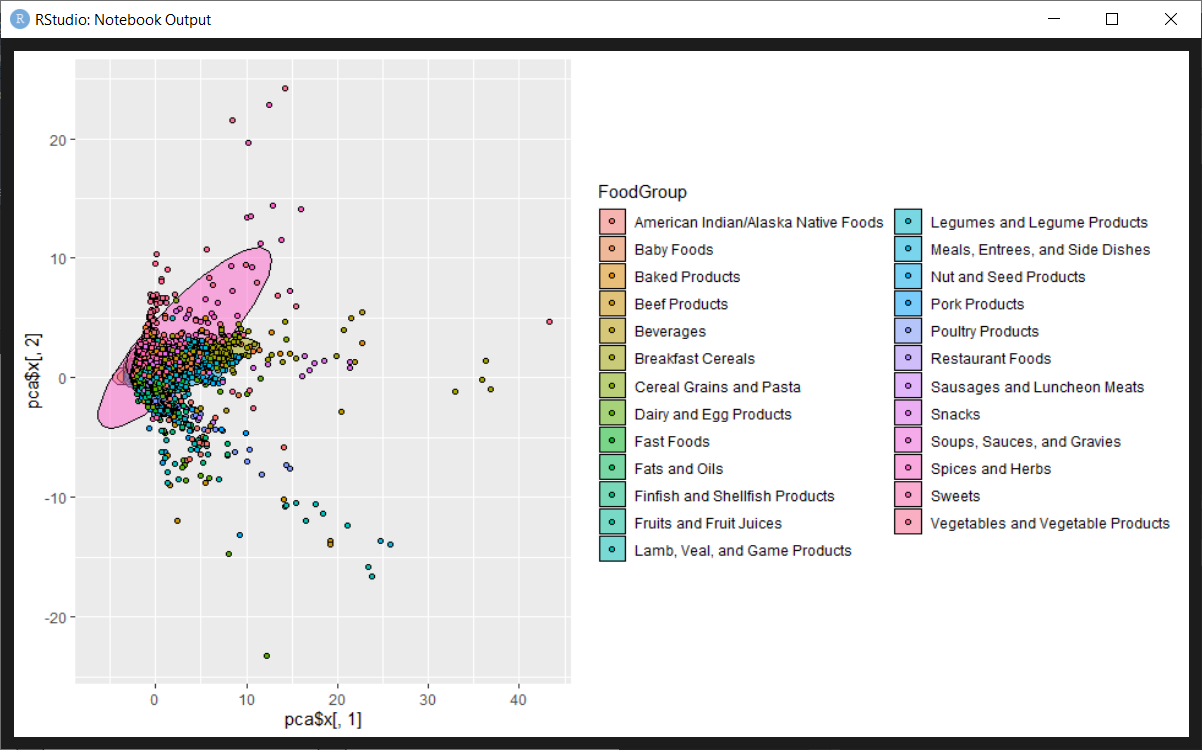
The picture above shows the percentage of variation of each PC (Also known as Factor Loading Analysis). As it was expected, PC1 accounts for most of the data variation, which is 16.1%. Next follows PC2 with 8.5%. Detailed percentage distribution and variation scores are indicated in the below

**plot**(pca, type = 'l') *#plotting eigenvalues per pc*



The plot shows how much variation in original data each Principal Component accounts for. As we can observe here, PC1 has the highest variation score. Variation accounts as “valuable” if the score (or eigenvalue) of it is more than 1.

**ggplot**(data, **aes**(pca**$**x[,1], pca**$**x[,2], col = FoodGroup, fill =FoodGroup))**+**  
 **stat\_ellipse**(geom = 'polygon', col = 'black', alpha = 0.5)**+**   
 **geom\_point**(shape = 21, col = 'black')



Last, but not least graph is an illustration of all samples with color-coding and grouping (25 determined groups from the initial dataset). We can observe here that there are distinguishing groups and PCA did a decent job in classification, but also, we can see knocked out samples scattered on the downside of the left-hand panel of the graph.

***Which components have the biggest impact on classification?***

To answer this question, we are going to compare PCA’s top ten loading scores on PC1 and PC2, and the Random Forest method.

### PCA loading scores

**for** (PC **in** 1**:**2){  
 component\_scores\_1 <- **names**(**sort**(**abs**(pca**$**rotation[,PC]), decreasing = T)[1**:**10])  
 **print**(PC)  
 **print**(pca**$**rotation[component\_scores\_1, PC])  
}

## [1] 1  
## Riboflavin\_.mg. Niacin\_.mg. Vit\_B6\_.mg. Iron\_.mg.   
## 0.3243723 0.3098795 0.2888527 0.2699890   
## Panto\_Acid\_mg. Thiamin\_.mg. Zinc\_.mg. Magnesium\_.mg.   
## 0.2528039 0.2394946 0.2261306 0.2250252   
## Folic\_Acid\_.Âµg. Copper\_mg.   
## 0.2169262 0.2011578   
## [1] 2  
## Carbohydrt\_.g. Protein\_.g. Cholestrl\_.mg. Fiber\_TD\_.g.   
## 0.3290625 -0.3216176 -0.2937984 0.2916753   
## Choline\_Tot\_..mg. Beta\_Carot\_.Âµg. Vit\_B12\_.Âµg. Sugar\_Tot\_.g.   
## -0.2830927 0.2560961 -0.2519450 0.2312374   
## Lut.Zea\_..Âµg. Vit\_K\_.Âµg.   
## 0.2112488 0.2030230

On the table above we can see the top ten food components on each axis that derives into clusters, Food Groups. On a PC1 axis, components with positive values push samples to the right-hand side, with negative values push to the left-hand side. The same logic goes to the PC2 axis, where components with positive values push samples up, and with negative values push samples down.

For instance, Riboflavin\_.mg. has a positive value, thus samples with high value in this component are pushed to the right on the PC1 axis. Simultaneously, samples with low values in the very same component are pushed to the left-hand side on the PC1 axis. Samples with high value in Protein\_.g. are pushed to the downside on the PC2 axis, and low value is pushed to the upside.

### Random Forest

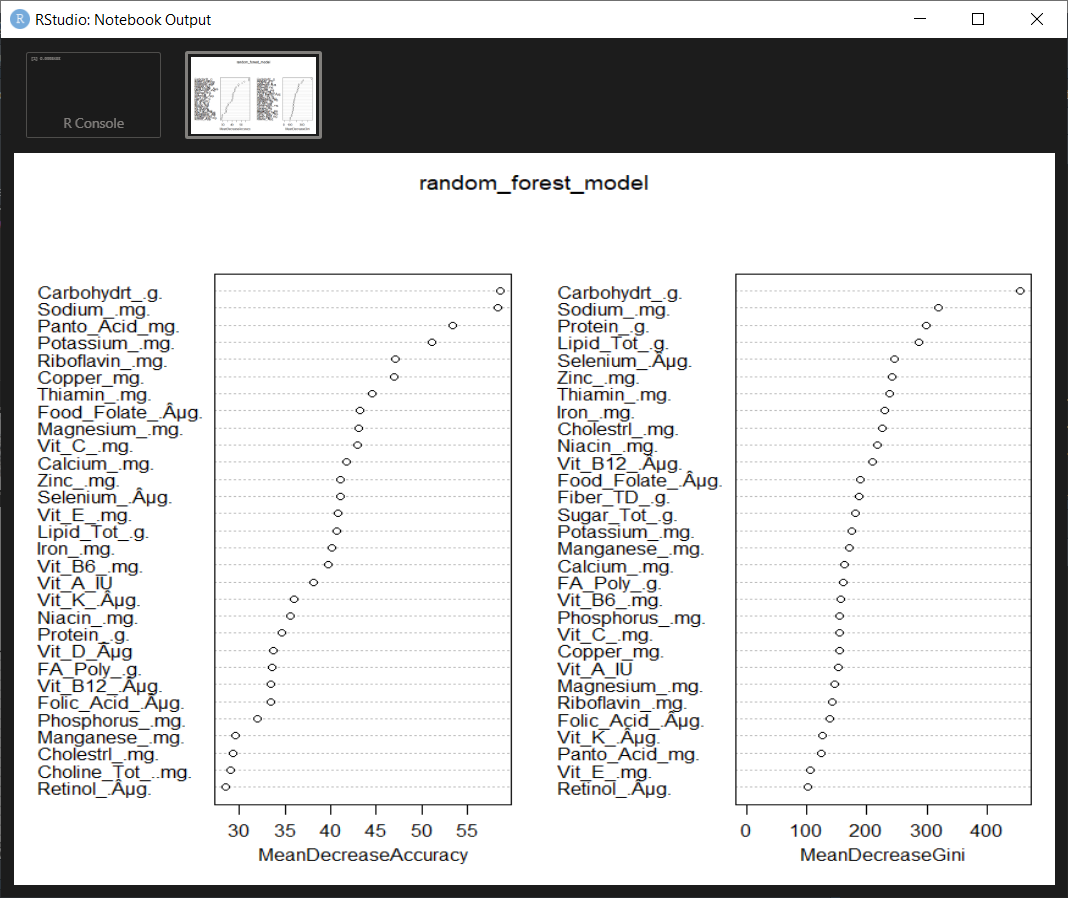
Random Forest is a relatively new method for the classification/regression approach. The algorithm uses a general method of bootstrap aggregation with a decision tree idea. Bootstrapping tends to decrease the variance of the model without increasing the bias. In other words, Random Forest is a more developed version of the Decision Tree with a very robust performance.

**library**(randomForest)

random\_forest\_model <- **randomForest**(FoodGroup**~**.,   
 data = training\_data, importance = T)  
*#prediction of training set*  
prediction\_forest\_train <- **predict**(random\_forest\_model, training\_data,   
 type = 'class')  
**mean**(prediction\_forest\_train **==** training\_data**$**FoodGroup)

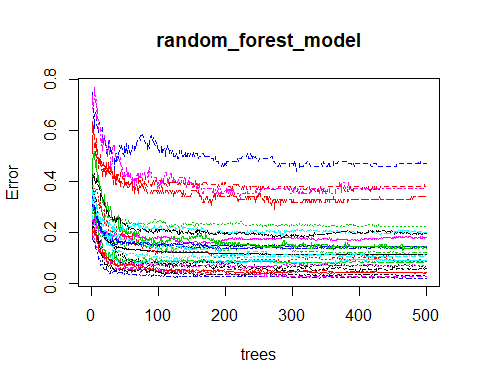
## [1] 0.9998483

*#prediction of testing set*  
prediction\_forest\_test <- **predict**(random\_forest\_model, newdata = testing\_data,   
 type = 'class')  
random\_forest\_accuracy <- **mean**(prediction\_forest\_test **==** testing\_data**$**FoodGroup)  
  
**varImpPlot**(random\_forest\_model)



The accuracy rate of Random Forest is 88% on testing data, which is remarkable. The random forest method allows us to plot the most important components for classification. As we can see in the table above (mean decrease accuracy), the most important components, in general, that helps us to classify food groups listed in there. In another table (mean decreased Gini), food components that are important in local terms (out of the bag) are listed. Mean Decrease accuracy table is more valuable for us since it indicates the most important components in general that play a significant role in classification.

**plot**(random\_forest\_model, type="l", main=**deparse**(**substitute**(random\_forest\_model)))



The picture above indicates the error rate and fluctuation for each Food Group.

***How does Decision Tree stand out in comparison with Random Forest, and which are deciding components in classification for this method?***

To answer those questions, let us run the Decision Tree method.

### Decision Tree

Decision Tree learning is a tool in statistics that uses tree-like moles with all possible consequences as leaves. The method is frequently used in data mining. It puts as a root node the most valuable variable and recursively does the same thing with the following nodes. It has a simple idea but the performance strongly depends on data type and types of variables.

**library**(rpart)

**library**(caret)  
decision\_tree\_model <- **rpart**(FoodGroup**~**., data = training\_data)

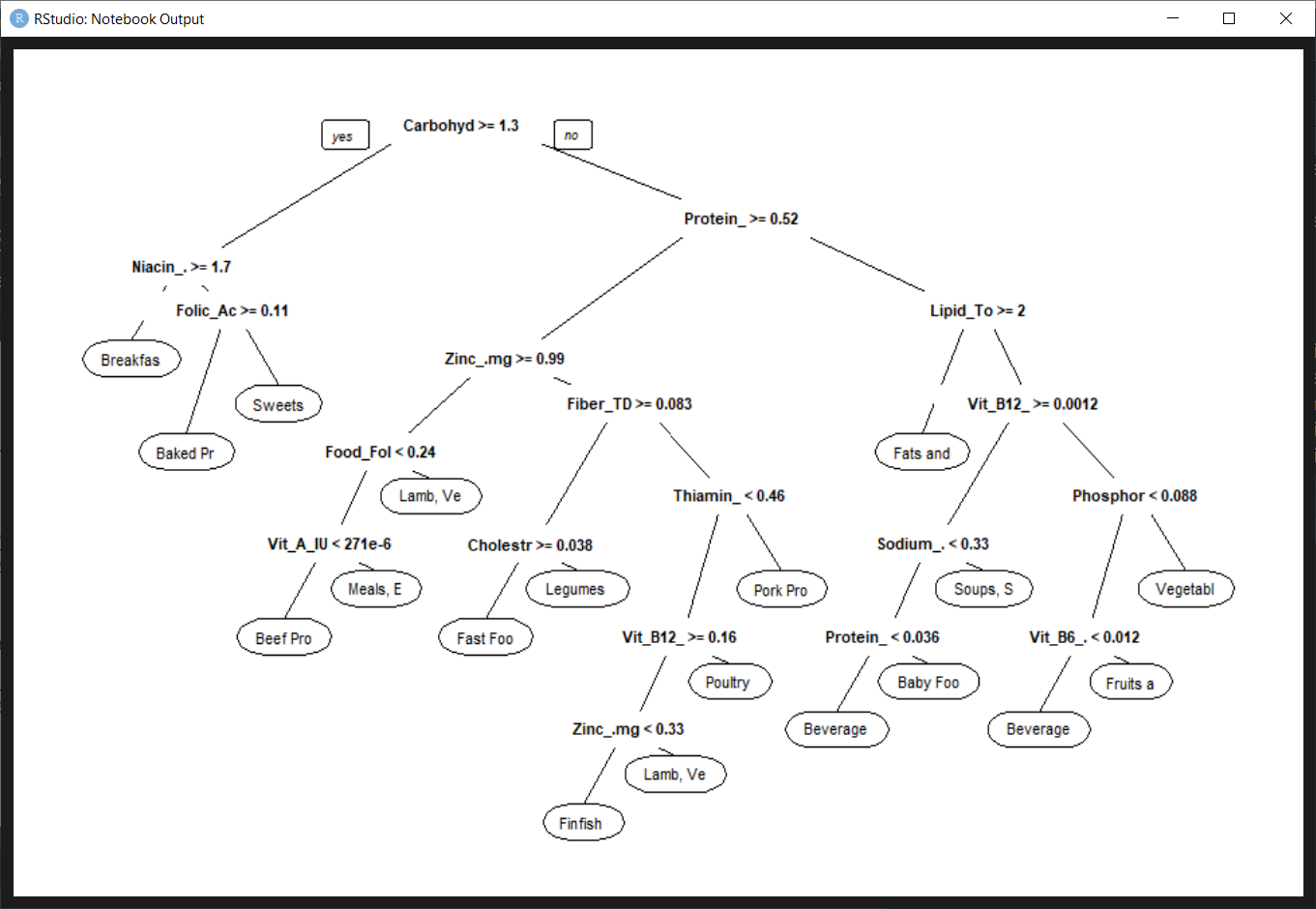
*#prediction on train data*  
prediction\_decision\_train <-**predict**(decision\_tree\_model, training\_data, type = 'class')   
**mean**(prediction\_decision\_train**==**training\_data**$**FoodGroup)

## [1] 0.5680267

*#prediction on testing data*  
prediction\_decision\_test <-**predict**(decision\_tree\_model, newdata = testing\_data, type = 'class')  
decision\_tree\_accuracy <- **mean**(prediction\_decision\_test**==**testing\_data**$**FoodGroup)   
**library**(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.6.3

**prp**(decision\_tree\_model)



The accuracy rate of Decision tree classification is 54%. In the plot above we can see classification trees with food components and conditional values, and leaves as a food group. For instance, if a sample has a value in the carbohydrates more or equal than 1.4 and has value in niacin more than 1.7, it would be classified as a Breakfast group.

***How accurate was the PCA method and how does it stand out compared to another unsupervised clustering method as k-means?***

To answer these questions, first of all, we have to measure how well principal components obtained in PCA could perform in classification, and in order to do that, Multinomial Logistic Regression is used in this analysis. Afterward, we are going to run k-means clustering.

### Multinomial Logistic Regression

Multinomial Logistic Regression (MLR) is a predictive regression analysis for a dependent variable with more than 2 levels. In this analysis, MLR is used to test Principal Components obtained before.

*#preparation for MLR*  
pca\_train <- **predict**(pca, training\_data)  
pca\_train <- **data.frame**(pca\_train, training\_data**$**FoodGroup)  
  
pca\_test <- **predict**(pca, testing\_data)  
pca\_test <- **data.frame**(pca\_test, testing\_data**$**FoodGroup)  
*#multinom reg*  
*#levels(training\_data$FoodGroup)*  
**library**(nnet)  
pca\_train**$**training\_data.FoodGroup <- **relevel**(pca\_train**$**training\_data.FoodGroup,   
 ref = 'Sweets')  
mnom\_model <- **multinom**(training\_data.FoodGroup**~**PC1**+**PC2**+**PC3**+**PC4**+**PC5, data = pca\_train)

*#testing on train*  
pca\_predict\_train <- **predict**(mnom\_model, pca\_train)  
**mean**(pca\_predict\_train**==**pca\_train**$**training\_data.FoodGroup)

## [1] 0.4239345

*#testing on test*  
pca\_predict\_test = **predict**(mnom\_model, newdata = pca\_test)  
mnom\_accuracy <- **mean**(pca\_predict\_test**==**pca\_test**$**testing\_data.FoodGroup)

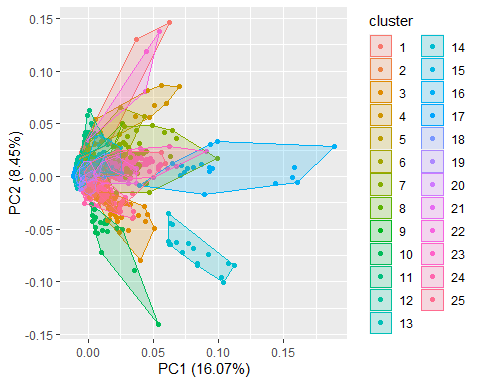
MLR was run on the first five PCs, since these components have the highest variation. The accuracy on the testing set is 42%. As a reference type, ‘Sweets’ was indicated for this analysis. We can indicate any other food group. We can conclude that the accuracy rate of PCA was mediocre.

### K-Means

K-means is another unsupervised clustering method frequently used in machine learning and data mining, where k stands for the number of clusters indicated by a user. The algorithm sets those selected clusters on data as ‘marked’ points and related other data points depending on the distance.

**library**(latexpdf)  
**library**(ggfortify)

**set.seed**(123)  
k\_means\_clust <- **kmeans**(data[**-length**(data)], 25)  
**autoplot**(k\_means\_clust, data, frame= T)



As we can see in the plot, k-means was able to perform decent clustering, but we still can observe an overlap on null points as we already saw on PCA and LDA as well.

***How does Support Vector Machine, optimization classification method that does not make any assumptions on the distribution of food groups stand out in comparison with other classification methods such as LDA or MRL in terms of accuracy and performance in general?***

### SVM

Support Vector Machine (SVM) is a supervised learning method perfectly suitable for classification and regression analysis. SVM is a very fast and well-performing algorithm even on multidimensional datasets. In current analysis the “Radial” kernel was used to separate samples.

**library**(e1071)

svm\_classificator = **svm**(formula = FoodGroup**~**.,  
 data = training\_data,  
 type = 'C-classification',  
 kernel = 'radial',  
 cost = 10)

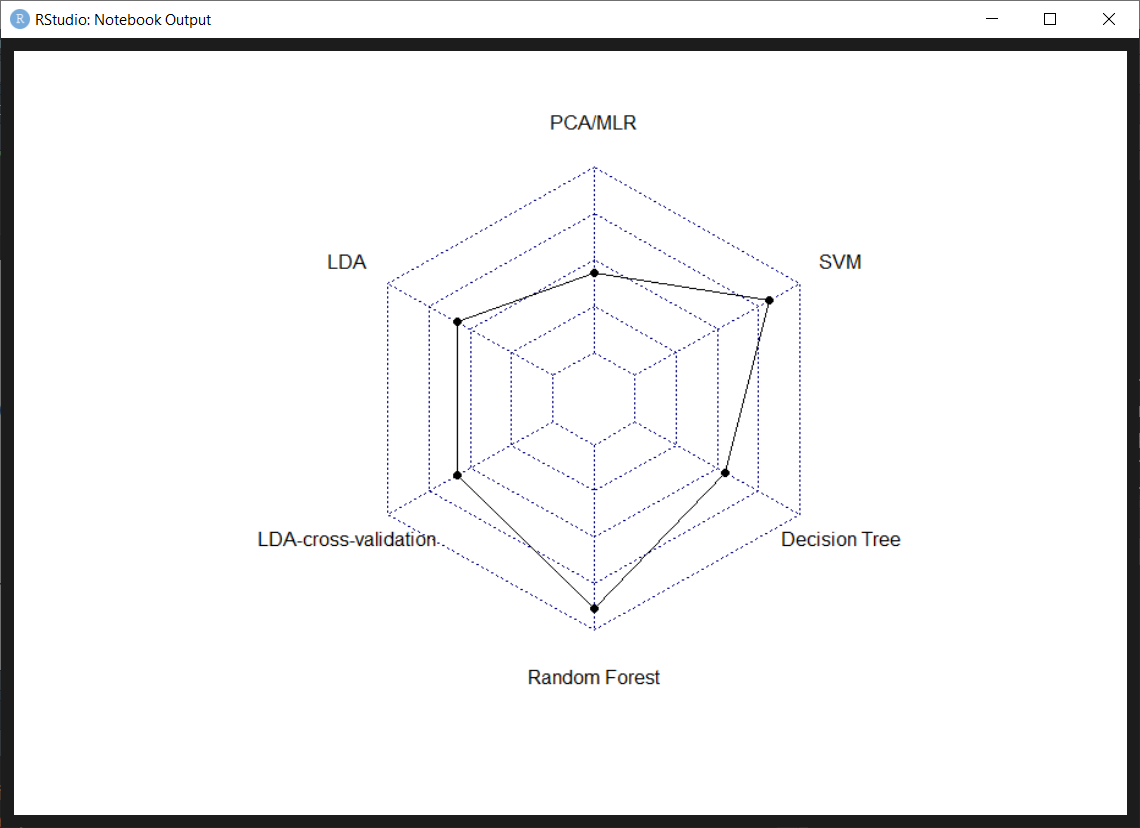
*#prediction on training set*  
prediction\_svm\_classif\_train = **predict**(svm\_classificator, training\_data)  
**mean**(prediction\_svm\_classif\_train**==**training\_data**$**FoodGroup)

## [1] 0.8901866

*#prediction of test data*  
prediction\_svm\_classif\_test = **predict**(svm\_classificator, newdata = testing\_data)  
*#table(prediction\_svm\_classif\_test, testing\_data$FoodGroup)*  
svm\_accuracy <- **mean**(prediction\_svm\_classif\_test**==**testing\_data**$**FoodGroup)

The performance of SVM also was good with decent accuracy at 81%.

## Final Table and Conclusion



|  |  |
| --- | --- |
|  |  |
| **Method** | **Accuracy\_percent** |
| PCA/MLR | 42.9 |
| LDA | 58.2 |
| LDA-CV | 58.2 |
| Random Forest | 88.5 |
| Decision Tree | 54.8 |
| SVM | 81.6 |
|  |  |

The table above along with the radar chart indicate accuracy rate of all used methods in this analysis. As we can see there, one of the most accurate methods for the food composition dataset are Random Forest and Support Vector Machine. Other methods have relatively poor accuracy results, but the reason for that kind of performance could be “socially classified” food groups that have nothing to do with actual food composition clustering. Some groups may contain a variety of components that may mislead unsupervised methods as we can see in PCA or k-means clustering.

## Bibliography

[1] (2020, April 1). SR11-SR28 : USDA ARS. Retrieved June 4, 2020, from <https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/methods-and-application-of-food-composition-laboratory/mafcl-site-pages/sr11-sr28/>

[2] Granato, D., Santos, J. S., Escher, G. B., Ferreira, B. L., & Maggio, R. M. (2018). Use of principal component analysis (PCA) and hierarchical cluster analysis (HCA) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective. Trends in Food Science & Technology, 72, 83-90.

[3] Giansante, L., Di Vincenzo, D., & Bianchi, G. (2003). Classification of monovarietal Italian olive oils by unsupervised (PCA) and supervised (LDA) chemometrics. Journal of the Science of Food and Agriculture, 83(9), 905-911.

[4] Oppel, S., & Huettmann, F. (2010). Using a random forest model and public data to predict the distribution of prey for marine wildlife management. In Spatial Complexity, Informatics, and Wildlife Conservation (pp. 151-163). Springer, Tokyo.

[5] Wie, S., Shanklin, C. W., & Lee, K. E. (2003). A decision tree for selecting the most cost-effective waste disposal strategy in foodservice operations. Journal of the American Dietetic Association, 103(4), 475-482.

[6] Santhanam, T., & Padmavathi, M. S. (2015). Application of K-means and genetic algorithms for dimension reduction by integrating SVM for diabetes diagnosis. Procedia Computer Science, 47, 76-83.

1. (2020, April 1). SR11-SR28 : USDA ARS. Retrieved June 4, 2020, from <https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/methods-and-application-of-food-composition-laboratory/mafcl-site-pages/sr11-sr28/> [↑](#footnote-ref-0)
2. (n.d.). Use of principal component analysis (PCA) and hierarchical .... Retrieved June 4, 2020, from <https://www.sciencedirect.com/science/article/pii/S0924224417306362> [↑](#footnote-ref-1)
3. (n.d.). (PCA) and supervised (LDA) chemometrics - Wiley Online .... Retrieved June 4, 2020, from <https://onlinelibrary.wiley.com/doi/abs/10.1002/jsfa.1426> [↑](#footnote-ref-2)
4. (n.d.). Using a Random Forest Model and Public Data to Predict the .... Retrieved June 4, 2020, from <https://link.springer.com/chapter/10.1007/978-4-431-87771-4_8> [↑](#footnote-ref-3)
5. (n.d.). A decision tree for selecting the most cost-effective waste .... Retrieved June 4, 2020, from <https://www.sciencedirect.com/science/article/pii/S0002822303000312> [↑](#footnote-ref-4)
6. (n.d.). Application of K-Means and Genetic Algorithms for Dimension .... Retrieved June 4, 2020, from <https://www.sciencedirect.com/science/article/pii/S1877050915004536> [↑](#footnote-ref-5)
7. (2020, April 1). SR11-SR28 : USDA ARS. Retrieved June 4, 2020, from <https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/methods-and-application-of-food-composition-laboratory/mafcl-site-pages/sr11-sr28/> [↑](#footnote-ref-6)