Visual Data Science
Assignment 1

Fabian Pechstein
00726104

November 2020

1 Clustering

This task's goal was to cluster food types according to two nutrition elements. The following combinations were suggestions, although finding a different preferred combination of nutrition elements was also an option. The suggested combination were the following:

- Energy_(kcal) and Protein_(g)
- Energy_(kcal) and Carbohydrt_(g)
- Energy_(kcal) and Water_(g)
- Energy_(kcal) and FA_Sat_(g)
- Water_(g) and Zinc_(mg)
- Water_(g) and Iron_(mg)
- Water_(g) and Phosphorus_(mg)
- Water_(g) and Sugar_(g)
- Sugar_(g) and Protein_(mg

when combining each element from the list with each other, we get 36 unique (= x and y axes can be swapped) combinations. To visualize the combinations I chose a simple scatter plot and tried to apply suitable coloring to show affiliation between data point and food category. With 25 categories there is unfortunately a repeat in colors, as I found no suitable color-map to solve this issue, instead I settled with the the available 10 colors as in theory clusters would be overlap categories in any case. Resulting visualisations can be seen in figures **??** and **??**.

The remaining plots have not been included in this report as I would focus on the combination of 'Energy_(kcal)' and 'Water_(g)' (see figure 1f; also not

from the suggestions) for the rest of the task. In my opinion this is well suited for clustering as already two more less or distinctive blobs are formed that are almost linearly separable. Adding more blobs would also cover the spaces in between and describe the data groups better (see. figure 3), although one should be careful not to overfit the cluster blobs.

Additionally this plot also tells a story of correlated features, as values with high energy usually have less water and vice versa (more on this in section 2)

1.1 Statistical analysis for clustering

For clustering using an statistical approach I first tried to reduce the the dimensionality of the original data-set using PCA on a scaled and a missing value replacement strategy however as a principal components where 'Lipid_Tot_(g)' and 'Lipid_Tot_(g)' I suspect, that my replacement strategy to deal with NAN values was erroneous in my opinion ans seems to have introduced an unintended bias through missing measurements. Figure 4 shows the

Kmeans clustering applied to my selected combination of 'Energy_(kcal)' and 'Water_(g)' can be found in figure 5. As we already saw in 4 kmeans is not well suited to model the underlying distributions that have generated our data, probably Gaussian Mixture Models would have been a better choice, but I assume this dilemma highlights the benefits of having meaningful visualisation at first to gain insights into the data at hand and then think about proper modeling.

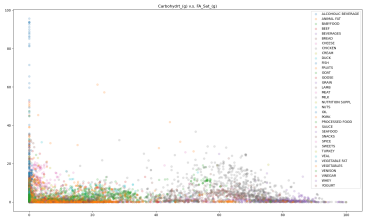
The right number of components is a tricky questions and depends on the used algorithm, as some will learn the number themselves, others like kmeans need the number of centroids set beforehand. In my case I used the ellbow method to find the sweet-spot for the number of components (see figure 6). I this method clustering is done for a range of number of components and the SSE is recorded. the SSE is the sum of the squared Euclidean distances of each point to its closest centroid. By choosing the setting as described by the ellbow point, we can avoid over fitting.

1.2 Summary

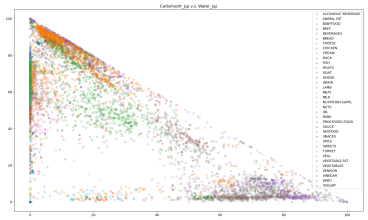
In summary must say, it felt easier to work with the visualizations to begin with, to familiarize with the data. Statistical methods greatly help in some cases, in other the model put to use by me were too simple.

1.3 Python functionalities used

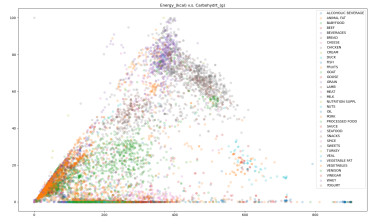
- pandas
- numpy
- sklearn.cluster.KMeans
- sklearn.decomposition.PCA



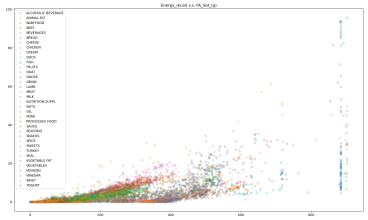
(a) Carbohydrt_(g) and FA_Sat_(g)



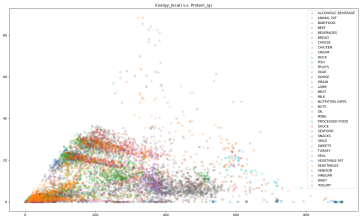
(b) Carbohydrt_(g) and Water_(g)



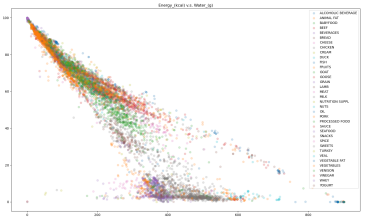
(c) Energy_(kcal) and Carbohydrt_(g)



(d) Energy_(kcal) and FA_Sat_(g)

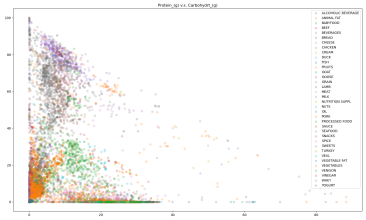


(e) Energy_(kcal) and Protein_(g)

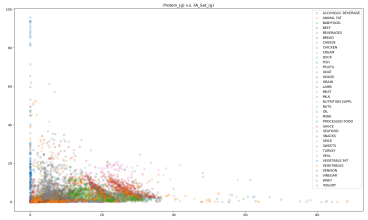


(f) Energy_(kcal) and Water_(g)

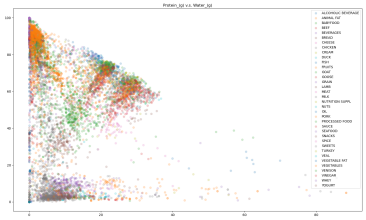
Figure 1: Task1, cluster visualisations using scatter plots



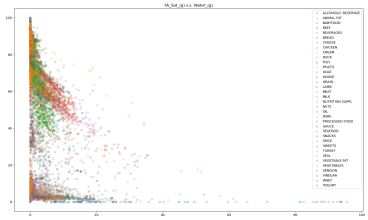
(a) Protein_(g) and Carbohydrt_(g)



(b) Protein_(g) and FA_Sat_(g)



(c) Protein_(g) and Water_(g)



(d) Water_(g) and FA_Sat_(g)

Figure 2: Task1, cluster visualisations using scatter plots

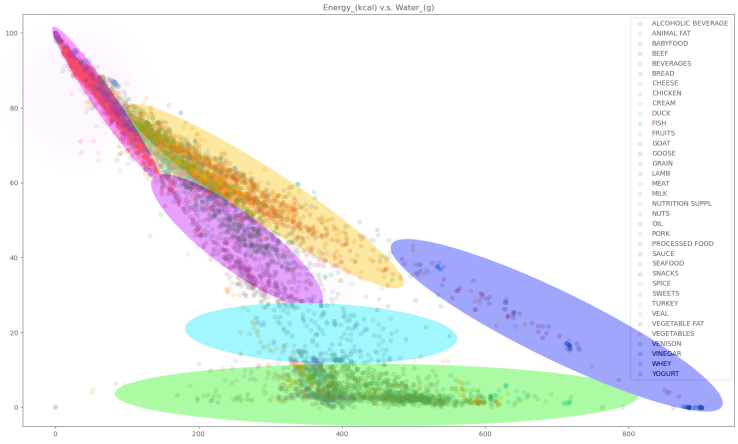


Figure 3: Clustering done by hand, compare to figure 5

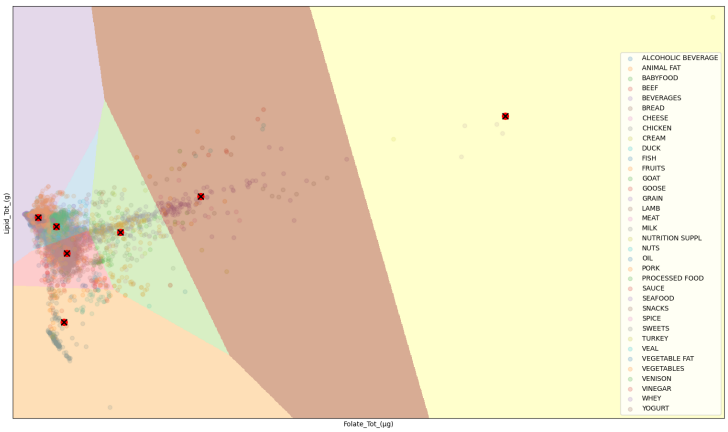


Figure 4: kmeans clustering and space tesselation of PCA reduced scaled data with number of components = 7, centroids marked as red dots with black X

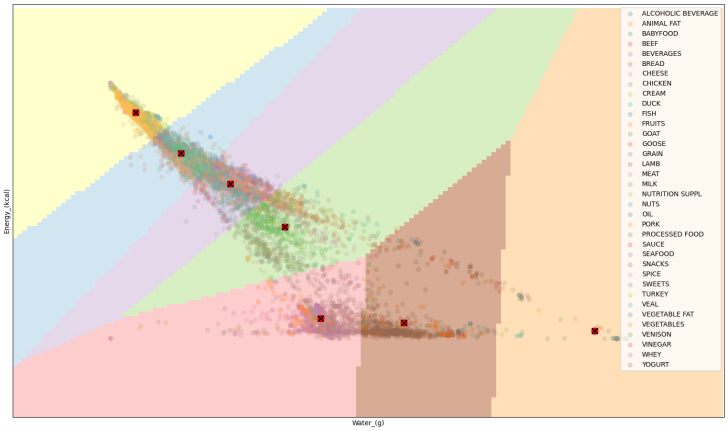


Figure 5: kmeans clustering and space tesselation of 'Energy_(kcal)' and 'Water_(g)' with number of components = 7, centroids marked as red dots with black X

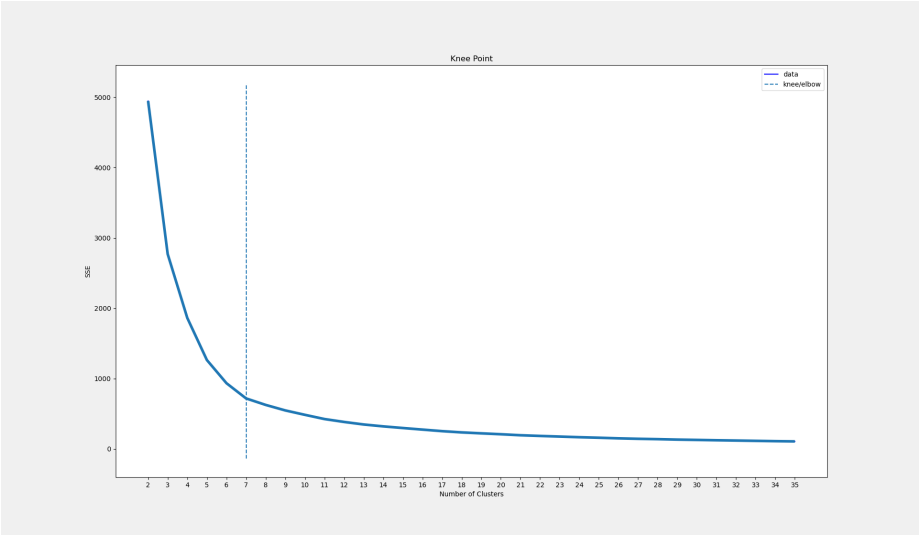


Figure 6: number of kmeans components against their respective SSE with highlighted ellbow point

- sklearn.impute.SimpleImputer
- sklearn.preprocessing.scale
- matplotlib.pyplot
- kneed.KneeLocator

2 Correlation

The goal here is to identify three strong correlations within the data. Using the samples co-variance matrix from the pandas.DataFrame.corr() function it's very easy to extract strong correlations (see table 1). A quite usable visual representation of the information from the matrix can be seen in figure 7.

I had another go to visualize correlations employing parallel coordinates, however although it is easy to spot positive and negative ones in a scaled dataset, the order of the axes have a great influence as this visualization simple lacks the additional dimension to show more inter-relations between features. Figure **??** shows an example of such a case:

By moving 'Carbohydrt_(g)' to the left we no longer can see the strong positive correlation between 'Folate_Tot_(µg)' and 'Folate_DFE_(µg)' and likewise the negative correlation between 'Carbohydrt_(g)' and 'Water_(g)'.

Consequently, by using only parallel coordinates as a method to find correlations, it's easy to miss some.

Folate_Tot_(µg)	Folate_DFE_(µg)	0.982891
Folic_Acid_(µg)	Folate_DFE_(µg)	0.952292
Lipid_Tot_(g)	FA_Mono_(g)	0.885439
Folate_Tot_(µg)	Folic_Acid_(µg)	0.879797
Ash_(g)	Sodium_(mg)	0.825681
Water_(g)	Energy_(kcal)	-0.900554
Water_(g)	Carbohydrt_(g)	-0.773920
Water_(g)	Sugar_Tot_(g)	-0.506365
Water_(g)	Lipid_Tot_(g)	-0.489781
Water_(g)	FA_Poly_(g)	-0.405290

Table 1: Top 5 positive and negative correlations

2.1 Python functionalities used

- pandas.DataFrame
- numpy
- matplotlib.pyplot
- seaborn (for the heatmap)
- sklearn.preprocessing.MinMaxScaler

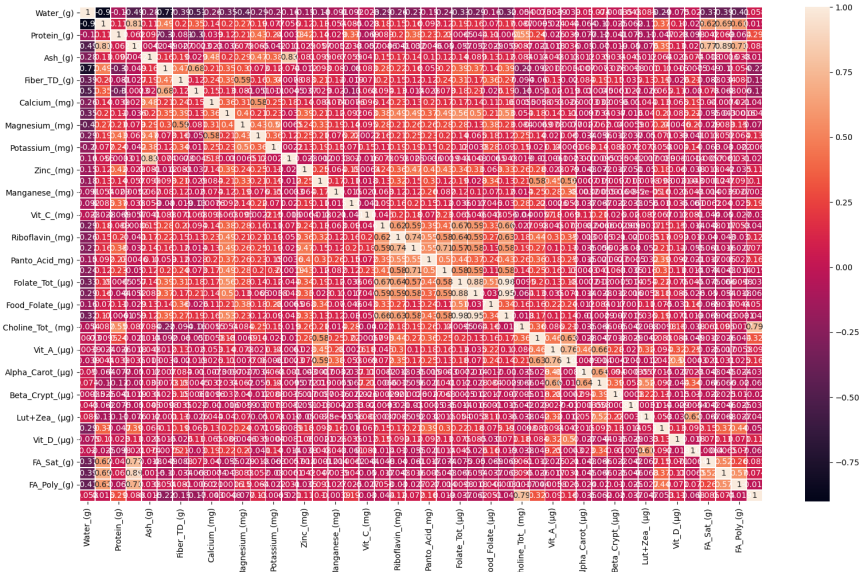


Figure 7: Heatmap visualisation of the co-variance matrix

3 Comparison

For the last sub-task I chose to take a closer look at items with the keyword 'PROCESSED FOOD' and split the resulting set in 7 sub groups of popular fast food restaurants. group affiliation was based on regular expression matches with the name column of available data records. The resulting groups are as follows:

- McDonald's (n=64)
- Burger King (n=18)
- Wendy's (n=12)
- Taco Bell (n=11)
- Subway (n=11)
- Pizza Hut (18)
- Domino's (n=10)

With these groups I hope to gain some insights into (1) general fast food options available, (2) a comparison between McDonald's and Burger King and finally (3) a comparison between Pizza Hut and Domino's.

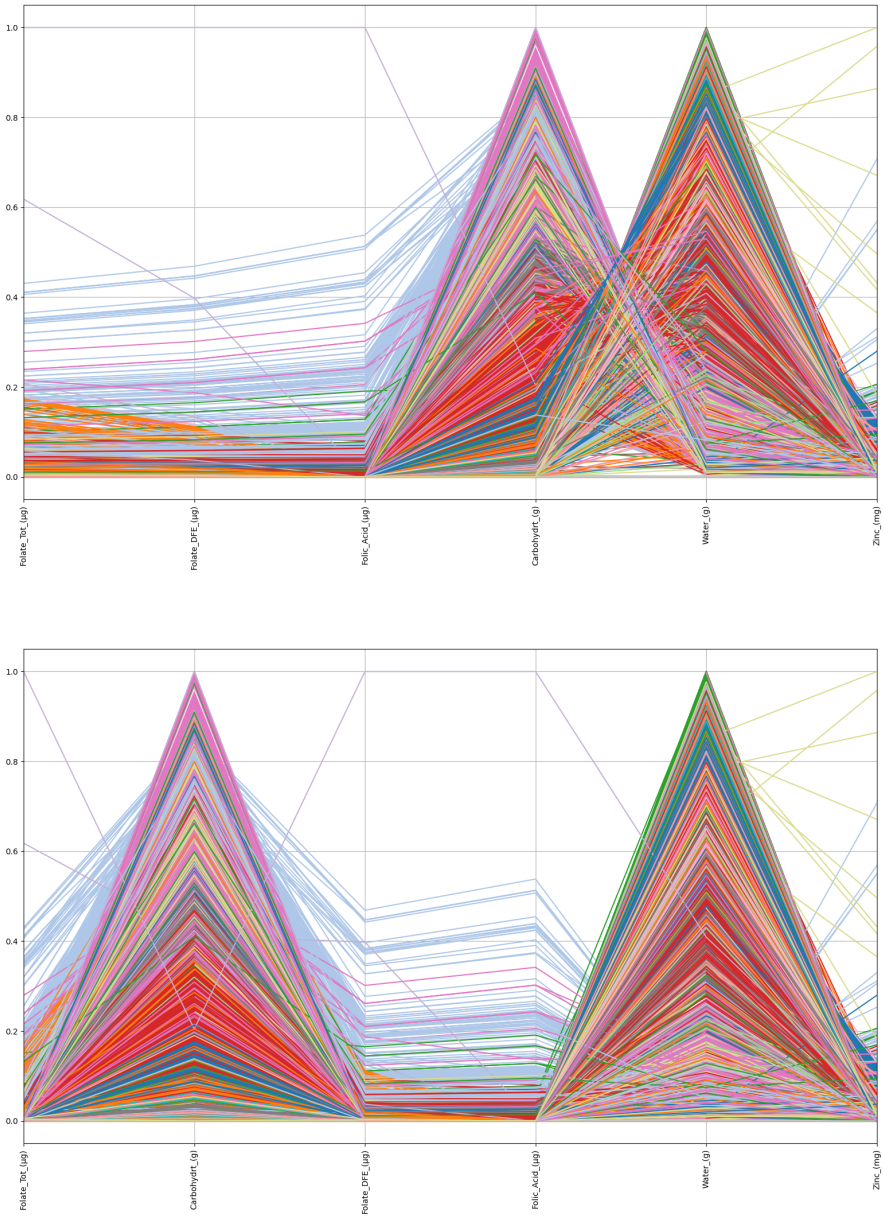


Figure 8: One issue with parallel coordinates is that the order of the axes matter.

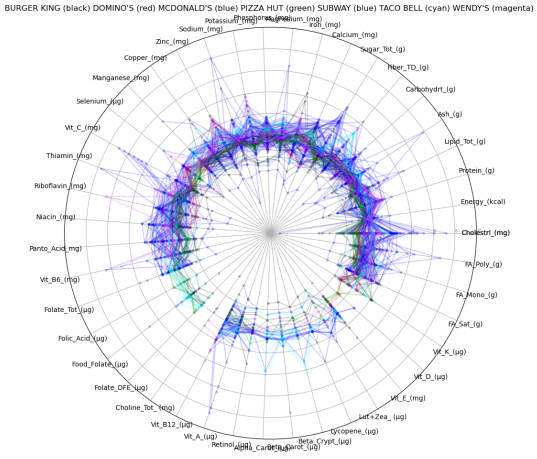


Figure 9: All groups in a radar chart

Figure 10 shows a cumulative figure consisting of box plots for all dimensions and each group side by side, if measurements were available. As we can see for some groups the number of available items is quite low, in case of 'MCDONALDS' items like sauces and deserts are also part of this group, which explains the high variance and outliers e.g for "Energy_(kcal) and 'Protein_(g)'.

Figures 11 to 17 show a radar chart respectively for each fast food corporation. Here all features axes are available in the charts, although some are unfortunately empty. Unfortunately, some scaling has been applied to the respective figures, thus making direct comparison near impossible. For a directly comparable radard chart, see figure 9, which in turn is more confusing due to the overlaps.

3.1 Fast Food Insights

On an overall level it is quite hard to get a real insights from the fast food industry, as some feature dimensions have missing values. In general Domino's, Subway and Taco seem to be the lesser evils, if we assume that the lower the area the plots the 'healthier' the dish is. However without a comparison to non-fast-food meals this hypothesis require further evaluation. Alltough, if we take a look at figure 9 containing all groups, there is no overall healthy.

Both Pizza Hut and Domino's meals contain a higher level of calcium in comparison to the other groups, followed by Subway.

When comparing MacDonald's and Burger King, what immediately pops up in the box-plots that McDonald's has incredible high amount of 'Sugar_tot_(g)', which might be due the deserts and sauces in the data set. On the other hand

Burger King items seem to contain more 'FA_Sat_(g)' and slightly more 'Protein_(g)' and surprisingly more 'Lipid_(g)'. When comparing the radar charts (figures 12 and 11), again McDonald's cover's more area.

In the pizza war, we can see that both contenders behave rather similar in the box-plots. However, again the radar charts tell a different story and let us assume that Domino's is the healthier choice. (based on the area thesis again).

Although the box-plots are easier to compare groups in one dimension, the radar charts offer a more spacious overview, with all axes scaled identically for all charts, we can compare groups rather easily. Reduction of available axes could help to paint a more clearer picture of the underlying data.

3.2 Python functionalities used

- pandas.DataFrame
- numpy
- matplotlib.pyplot
- re
- sklearn.preprocessing.MinMaxScaler

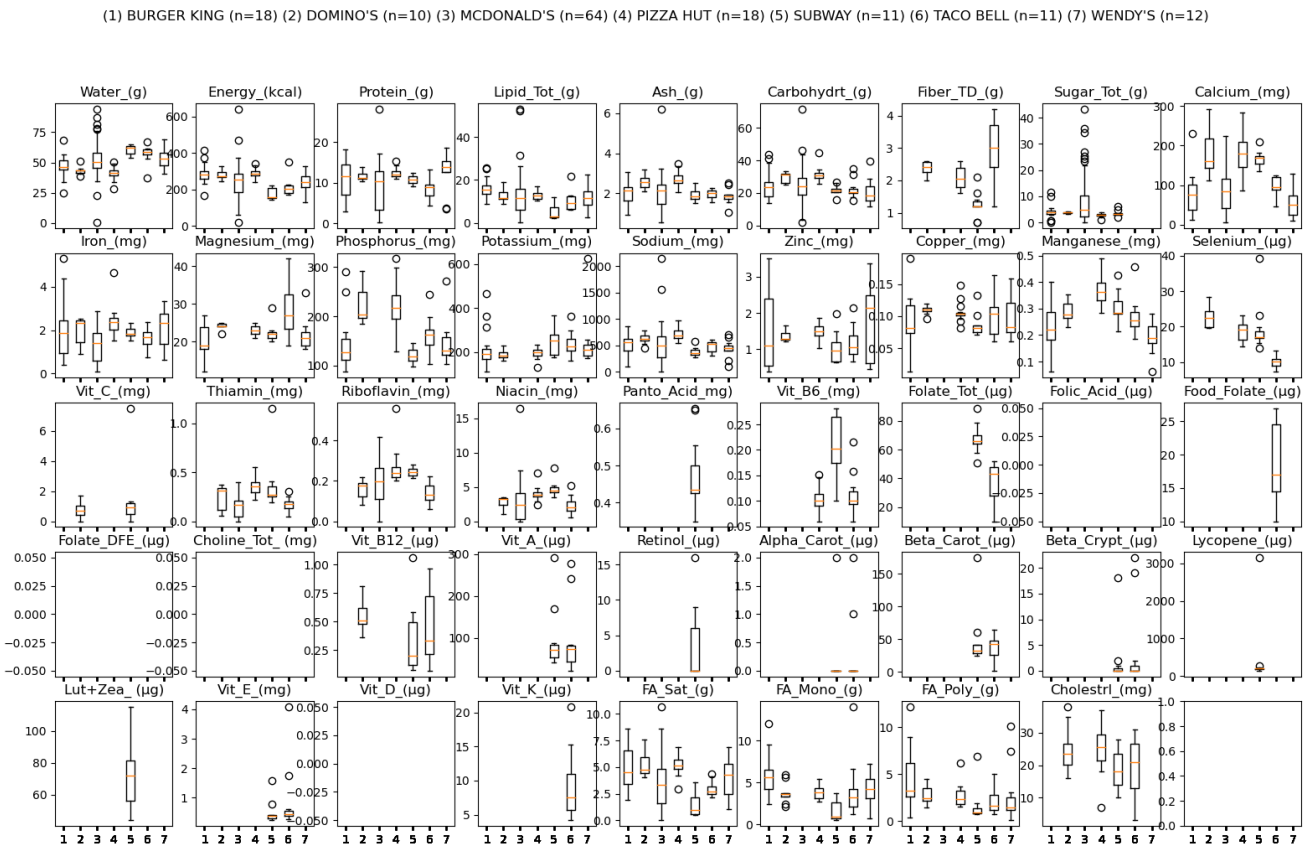


Figure 10: box blots for each feature dimension and each group (x-tick is group affiliation)

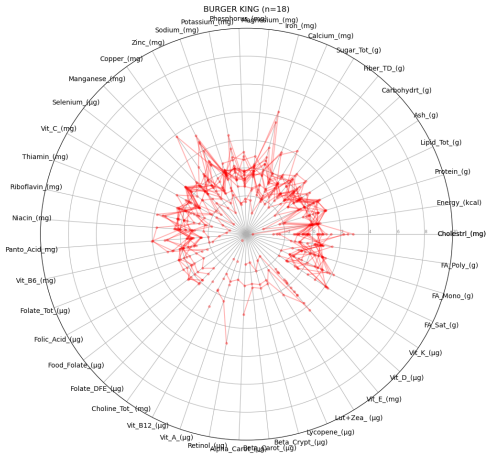


Figure 11: Burger King

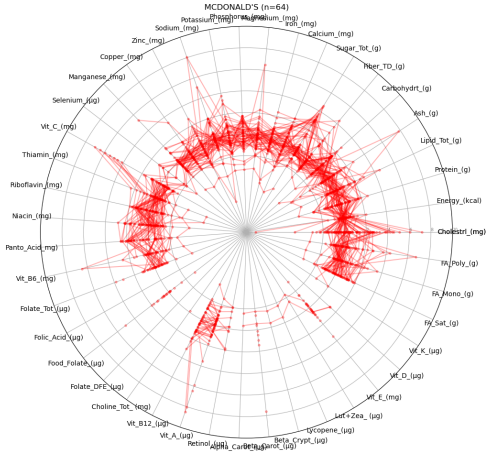


Figure 12: McDonald's

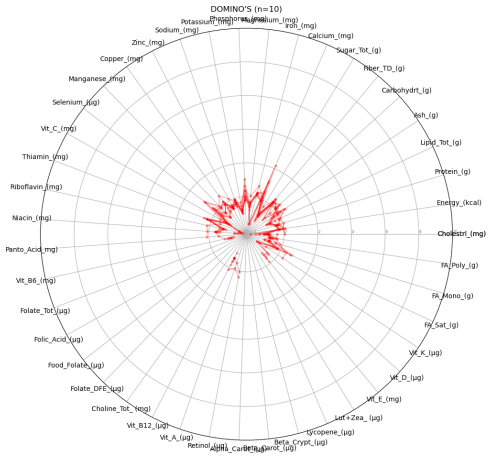


Figure 13: Domino's

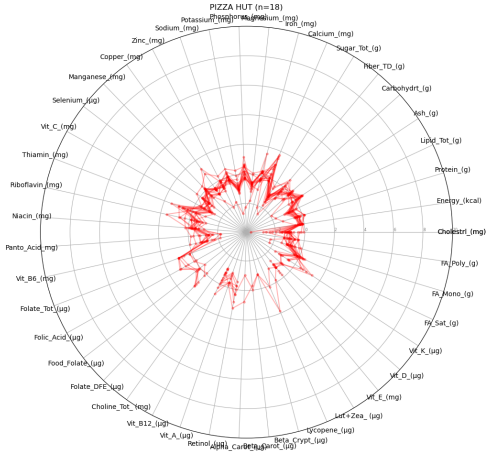


Figure 14: Pizza hut

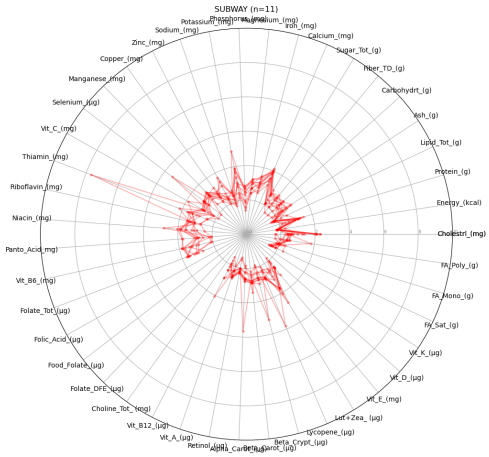


Figure 15: Subway

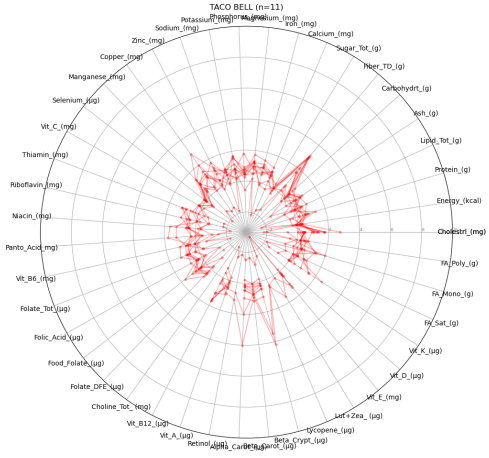


Figure 16: Taco Bell

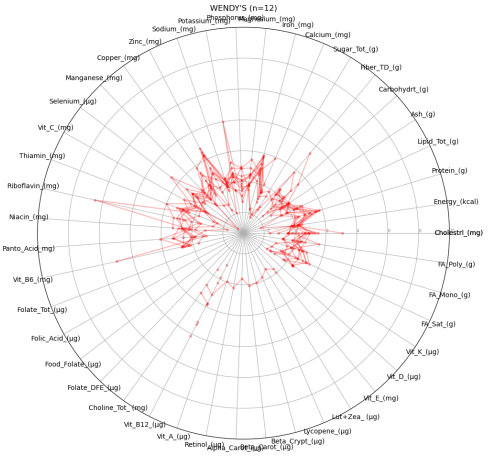


Figure 17: Wendy's