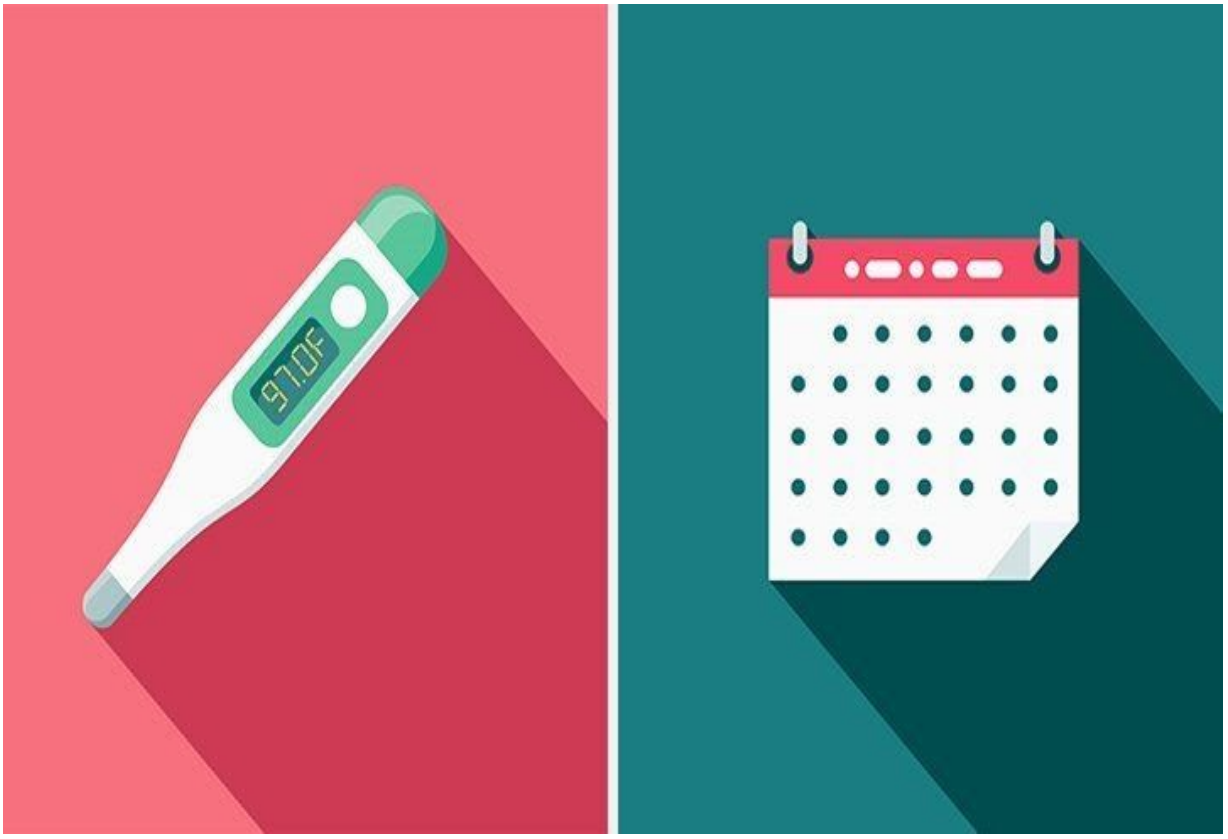
Data Mining
Contraception Methods



Lecturer: FALIH Issam

He NI
Bader GUETARI

BI – 2

SUMMARY

Introduction 3

I – Characteristics of the data set 4

II – Attribute selection/extraction 7

1) Feature Selection 7

2) Pre-processing 7

III – Best results/best parameters 9

Conclusion 14

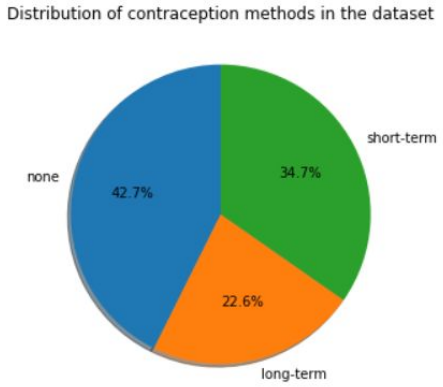
Introduction

For this project, we have chosen the cmc dataset. This dataset is relevant to how women in Indonesia handle contraception. The aim of this project is to analyze this dataset by using pre-processing steps and clustering algorithms. Indeed, we suppose that we do not dispose of the features given. To answer this project proposal, we will first present the dataset characteristics, we will select and extract attributes and finally we use unsupervised methods for clustering this dataset.

I – Characteristics of the data set

The data presents a total of 1473 observations with 10 features computed. The repartition of the different methods used by women is the following:

```
628 women are not using any contraception method, 333 women using long-term contraception method and 511 women areusing shor
t-term contraception method.
```



There are no missing values as we can see in the screenshot of the result of a Python command to get information on the dataset :

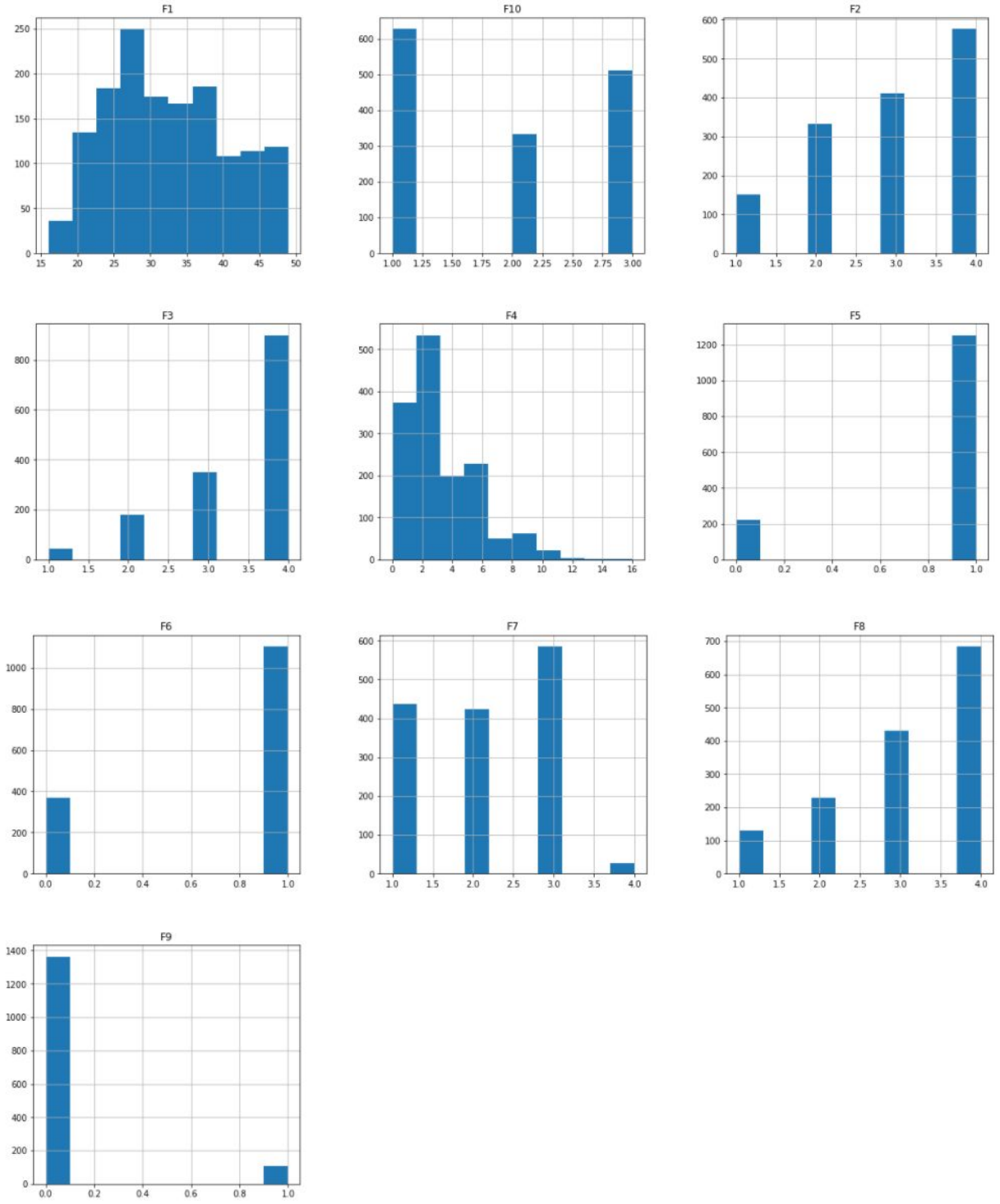
General information about dataset

With pandas `info()` method, we can see some basic information about our data. We can see that there are no missing values.

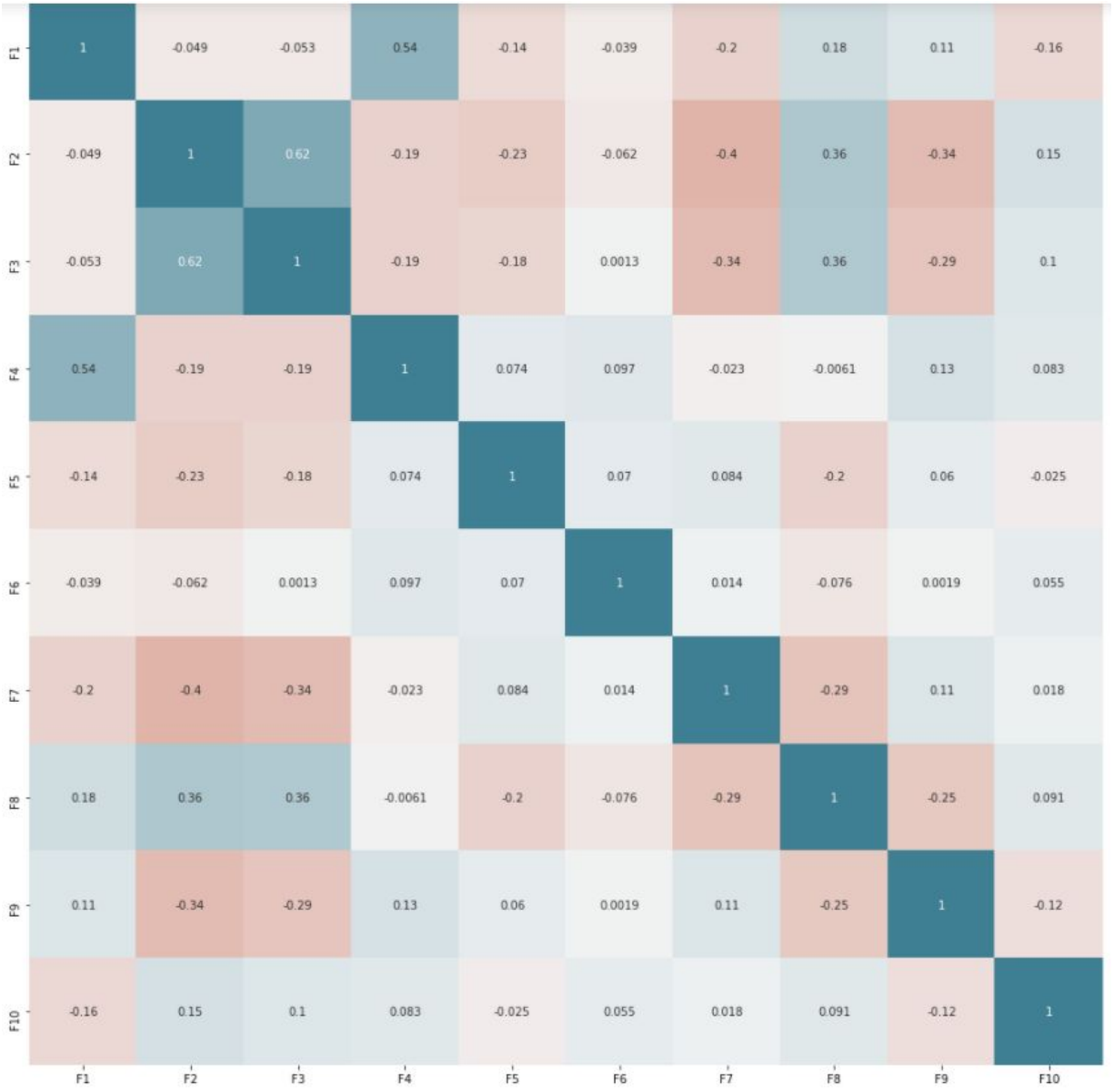
```
Entrée [95]: print(contraceptive_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1472 entries, 0 to 1471
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Age            1472 non-null   int64
 1   W_Education    1472 non-null   int64
 2   H_Education    1472 non-null   int64
 3   Childrens      1472 non-null   int64
 4   Religion       1472 non-null   int64
 5   Employed       1472 non-null   int64
 6   H_Occupation   1472 non-null   int64
 7   SOL            1472 non-null   int64
 8   MediaExp       1472 non-null   int64
 9   Method         1472 non-null   int64
dtypes: int64(10)
memory usage: 115.1 KB
None
```

In the notebook associated with this report, we also give some basic statistics about the data (mean, standard deviation, minimum values…). We also plot some histogram to visualize if the data presents some outliers or not.



We also compute the correlation between each feature and the correlation between the target feature and the rest of the features. We plot the correlation. We can see that most of the featured are not really correlated :


F1 1 -0.049 -0.053 0.54 -0.14 -0.039 -0.2 0.18 0.11 -0.16
F2 -0.049 1 0.62 -0.19 -0.23 -0.062 -0.4 0.36 -0.34 0.15
F3 -0.053 0.62 1 -0.19 -0.18 0.0013 -0.34 0.36 -0.29 0.1
F4 0.54 -0.19 -0.19 1 0.074 0.097 -0.023 -0.0061 0.13 0.083
F5 -0.14 -0.23 -0.18 0.074 1 0.07 0.084 -0.2 0.06 -0.025
F6 -0.039 -0.062 0.0013 0.097 0.07 1 0.014 -0.076 0.0019 0.055
F7 -0.2 -0.4 -0.34 -0.023 0.084 0.014 1 -0.29 0.11 0.018
F8 0.18 0.36 0.36 -0.0061 -0.2 -0.076 -0.29 1 -0.25 0.091
F9 0.11 -0.34 -0.29 0.13 0.06 0.0019 0.11 -0.25 1 -0.12
F10 -0.16 0.15 0.1 0.083 -0.025 0.055 0.018 0.091 -0.12 1
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10

II – Attribute selection/extraction

1. Feature Selection

There are 10 features in the CMC dataset.

	Age	W_Education	H_Education	Childrens	Religion	Employed	H_Occupation	SOL	MediaExp	Method
0	45	1	3	10	1	1	3	4	0	1
1	43	2	3	7	1	1	3	4	0	1
2	42	3	2	9	1	1	3	3	0	1
3	36	3	3	8	1	1	3	2	0	1
4	19	4	4	0	1	1	3	3	0	1

We put Method as label, into the target data frame, and the remaining 9 features into the data frame.

Since we are using unsupervised learning algorithms, we do not need to segment the data.

2. Pre-processing

a. Standardization

We normalize scale the data to get the DataScaler

```
DataScaler
```

```
array([[ 1.51452951, -1.93088138, -0.52681869, ...,  0.99681168,
         0.88732745, -0.28279083],
       [ 1.27134571, -0.94535524, -0.52681869, ...,  0.99681168,
         0.88732745, -0.28279083],
       [ 1.14975382,  0.0401709 , -1.751901  , ...,  0.99681168,
        -0.13710079, -0.28279083],
       ...,
       [ 0.78497812,  0.0401709 , -0.52681869, ..., -1.31572858,
         0.88732745, -0.28279083],
       [ 0.05542674,  0.0401709 , -0.52681869, ..., -0.15945845,
        -1.16152902, -0.28279083],
       [-1.89004361,  0.0401709 , -0.52681869, ..., -0.15945845,
         0.88732745, -0.28279083]])
```

b. PCA Dimensionality Reduction

In a normal data analysis situation, we need to retain 95% of the information, i.e. the cumulative of the exploded variance ratio of the features should be greater than 95%

```
pca=PCA(n_components=0.95).fit(DataScaler)
DataPca=pca.transform(DataScaler)


print('after pca.transform:')
print("DataPca.shape",DataPca.shape)
print('explained variance ratio (comulative > 0.95): %s'% str(pca.explained_variance_ratio_))
```

```
after pca.transform:
DataPca.shape (1472, 8)
explained variance ratio (comulative > 0.95): [0.27999379 0.18554707 0.11840498 0.10056442 0.09410836 0.07160332
 0.06633528 0.04347225]
```

It can be seen that the explained variance ratio of the first 8 features together is greater than 95%, so the effect of dimensionality reduction is not good, and in this case the accuracy of the K-mean algorithm is also very low, at 32%:

```
correct_amount_km = km_all[km_all["Predict"] == km_all["Origin"]].shape[0]
print(correct_amount_km)
print("Accuracy is {0} % ".format(correct_amount_km / km_all.shape[0] * 100))
```

```
476
Accuracy is 32.33695652173913 %
```

After iteration, we found that the accuracy of the K-mean algorithm was higher when pca was reduced to 2 dimensions, at 42%:

```
correct_amount_km = km_all[km_all["Predict"] == km_all["Origin"]].shape[0]
print(correct_amount_km)
print("Accuracy is {0} % ".format(correct_amount_km / km_all.shape[0] * 100))
```

```
623
Accuracy is 42.32336956521739 %
```

So we decided to drop to 2 dimensions.

III – Best results/best parameters

Since we have chosen Method as the Label for our data set, then we already know that our data will be divided into 3 categories.

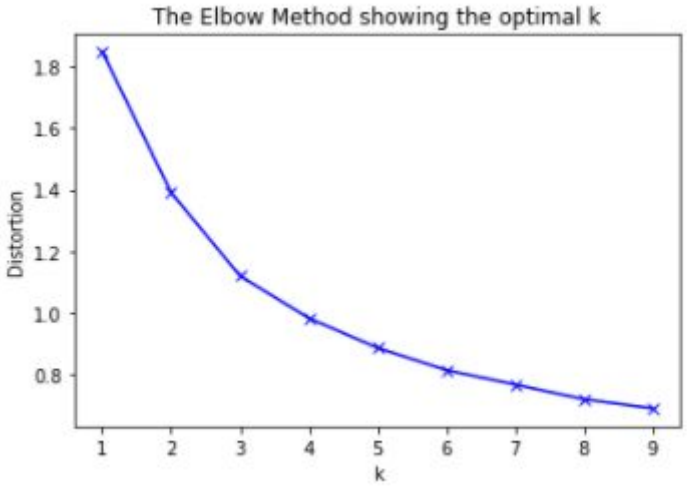
We are going to use the K-mean algorithm, so even though we already know that K=3, we still need to make a determination of the K value.

First Method: The Elbow Method

We need to calculate the SSE (sum of the squared errors), which is the clustering error of all the samples and represents how well the clustering works.

As the number of clusters k increases, the samples will be more finely divided and the degree of aggregation of each cluster will gradually increase, so the sum of the squared errors and SSE will naturally become smaller.

When k is less than the true number of clusters, the SSE decreases significantly because increasing k increases the degree of aggregation of each cluster, while when k reaches the true number of clusters, increasing k yields a rapidly decreasing return on the degree of aggregation, so the SSE decreases abruptly and then flattens out as k continues to increase, i.e. the graph of SSE versus k is an elbow shape, and the value of k corresponding to this elbow is the true number of clusters.



From the graph we can see that the curvature is highest for a k value of 3. Therefore, for the clustering of this dataset, the optimal number of clusters should be 3.

Second Method: Fowlkes-Mallows Index

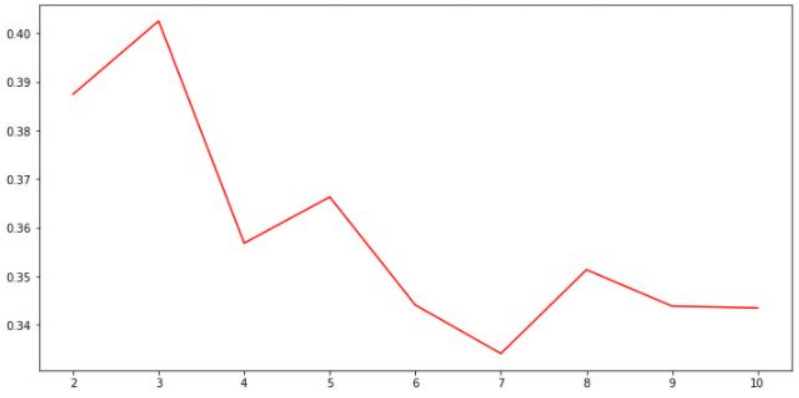
The FMI is the result of a geometric average of the recall and precision calculated from the clustering results and the true values, taking values in the range [0,1], the closer to 1 the better.

```
The FMI score for the 2th clustering model is: 0.4374589601882109
The FMI score for the 3th clustering model is: 0.3721814271383299
The FMI score for the 4th clustering model is: 0.32086887660429825
The FMI score for the 5th clustering model is: 0.29549096447298895
The FMI score for the 6th clustering model is: 0.2694789698509319
The FMI score for the 7th clustering model is: 0.2536480174143609
The FMI score for the 8th clustering model is: 0.2370168918768751
The FMI score for the 9th clustering model is: 0.22645691015890743
The FMI score for the 10th clustering model is: 0.21345601225714977
```

The FMI score is also maximum for k=2, but it does not match

Third Method: Silhouette Coefficient

The Silhouette Coefficient can be used to select the appropriate number of clusters. The points where the coefficients vary the most can be found visually according to the line graph, and the point where the greatest distortion is considered to occur is the best number of clusters.



We can see that the contour coefficient changes the most from 3 -> 4, from 0.4 -> 0.355. so we can be sure it is 3.

Fourth Method: Calinski-Harabasz index

The Calinski-Harabaz index can also be used to select the best number of clusters and is much faster than the contour coefficients. The Calinski-Harabasz score is higher when the covariance of the internal data is smaller and the covariance between categories is larger.

```
The Calinski-Harabasz index score for the 2th clustering model is: 1046.5344776171557
The Calinski-Harabasz index score for the 3th clustering model is: 1198.749505481909
The Calinski-Harabasz index score for the 4th clustering model is: 1194.8838538205362
The Calinski-Harabasz index score for the 5th clustering model is: 1194.0384248979258
The Calinski-Harabasz index score for the 6th clustering model is: 1186.1220946979495
The Calinski-Harabasz index score for the 7th clustering model is: 1146.402169792725
The Calinski-Harabasz index score for the 8th clustering model is: 1131.6732505880218
The Calinski-Harabasz index score for the 9th clustering model is: 1136.0965681036876
The Calinski-Harabasz index score for the 10th clustering model is: 1138.0593610166247
```

Obviously maximum at k = 3

K-mean Algorithm And Its Results:

We use the K-mean for DataPca to obtain the predicted values

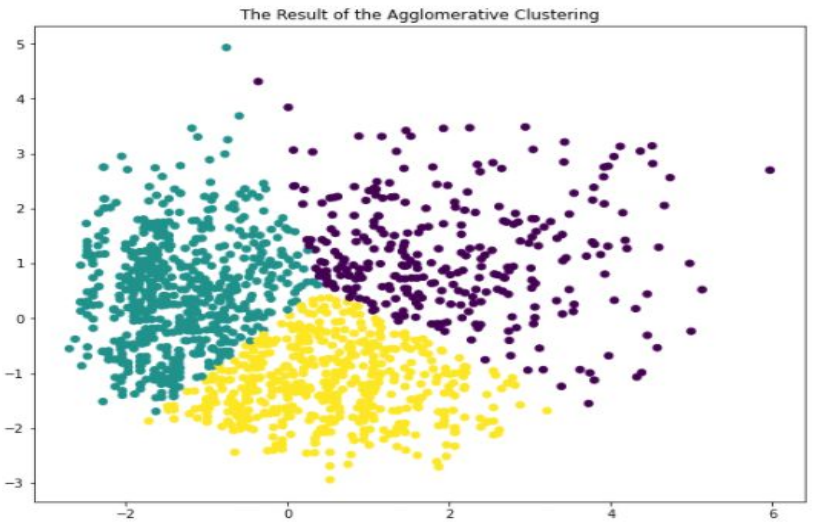
```
#K-mean
from sklearn.cluster import KMeans
kmean=KMeans(n_clusters=3,random_state=23).fit(DataPca)
print("The model\n",kmean)
```

```
The model
 KMeans(n_clusters=3, random_state=23)
```

We compared the predicted values with the original data and calculated a correct rate of 42.3%

```
correct_amount_km = km_all[km_all["Predict"] == km_all["Origin"]].shape[0]
print(correct_amount_km)
print("Accuracy is {0} % ".format(correct_amount_km / km_all.shape[0] * 100))
```

```
623
Accuracy is 42.32336956521739 %
```



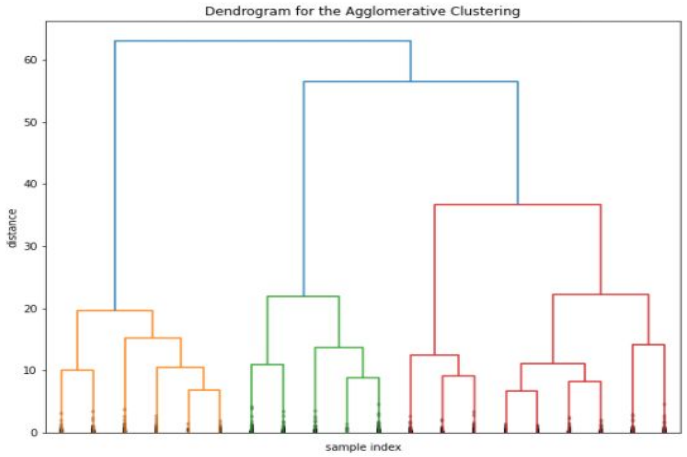
When PCA(n_components = 2),Kmeans(n_cluster = 3), we calculated the highest correct rate.

But this correct rate is a bit unsatisfactory, so we want to try another clustering method, HCA

The HCA Algorithm And Its Results:

Drawing a tree diagram.

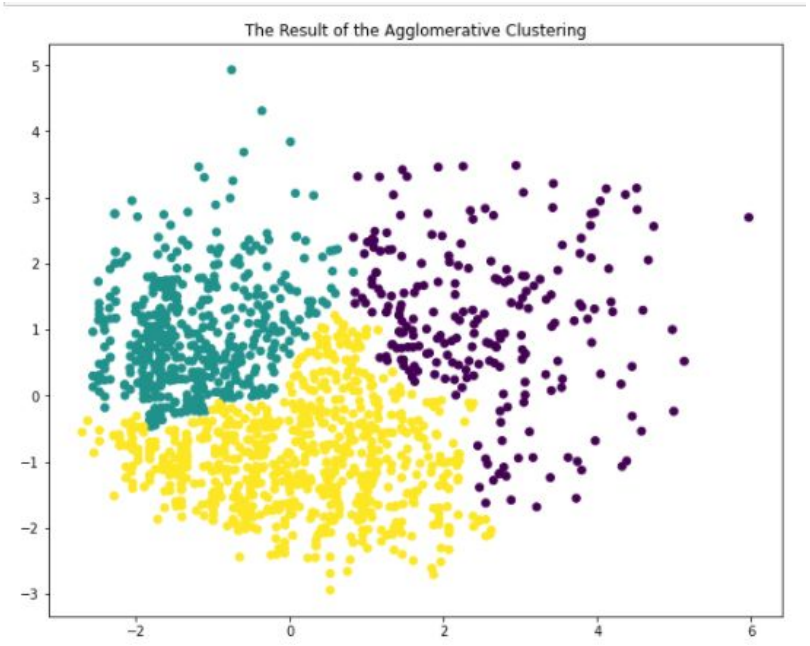
The function dendrogram is given in SciPy for drawing a tree diagram based on the results of hierarchical clustering Z. We thus draw the last 20 merges in this experiment.



We then calculated the accuracy of the clustering results to give 43%.

```
correct_amount_HCA = H_all[H_all["Predict"] == H_all["Origin"]].shape[0]
print(correct_amount_HCA)
print("Accuracy is {0} % ".format(correct_amount_HCA / H_all["Origin"].shape[0] * 100))
```

```
634
Accuracy is 43.07065217391305 %
```



So the accuracy rate should be right around 43%.

Conclusion

Here for parameter tuning, we have only dealt with the number of dimensions in PCA, and the number of clusters in the k-mean.

It is actually possible to tweak the hyperparameters of the k-mean to obtain higher accuracy.

But given the somewhat heavy workload, we did not do it (we used Grid in the Big Data Analysis project).

We also did a project last semester to determine the benignity and malignancy of cancer cells, there were several features in this dataset that had very high exploded variance ratios, and two of them had exploded variance ratios that added up to over 95%, so we let the PCA parameter of cancer dataset takes 2. The final result was 92% correct using k-mean.

Reasons for the low accuracy rates ultimately obtained for this project.

1. We believe that this is because all the other features have low correlation coefficients with Method, none of which is greater than 0.2 (either positive or negative).
2. The exploded variance ratio of each feature is very low, so the PCA parameter taken as 2 does not retain enough information to perform a dimensionality reduction.
3. However, even if enough information is retained (the PCA parameter is 8), the final accuracy is even lower, which is an interesting problem, we guess because the correlation coefficient between Method and other features is too low.