Analysis of Difference in Supplier Clustering Using Matlab And Phyton by Implementing K-Means Clustering Method (Case Study of Company X)

Arranged By:

Aprilia Indah Saraswanti 02411740000010

Merisa Khristanti Febrianka Hanka 02411740000094

SUPERVISOR 1

Nani Kurniati, S.T., M.T., Ph.D.

NIP. 197504081998022001

SUPERVISOR 2

Dewanti Anggrahini, S.T., M.T.

NIP. 198805022019032014

Company's Profile

Company X is a branch of a global manufacturing company engaged in automation. Company X works to produce products such as components, tools and automation systems.

- Three types of products produced by Company X, there are Relays, Switches, and Industrial Automation Business (IAB).
- The core process of Company X is assembling the components obtained from suppliers. Suppliers owned by Company X have contributed as much as 70% of all components and products assembled by Company X.

Background

Currently, there are 62 suppliers both located within and outside Indonesia.

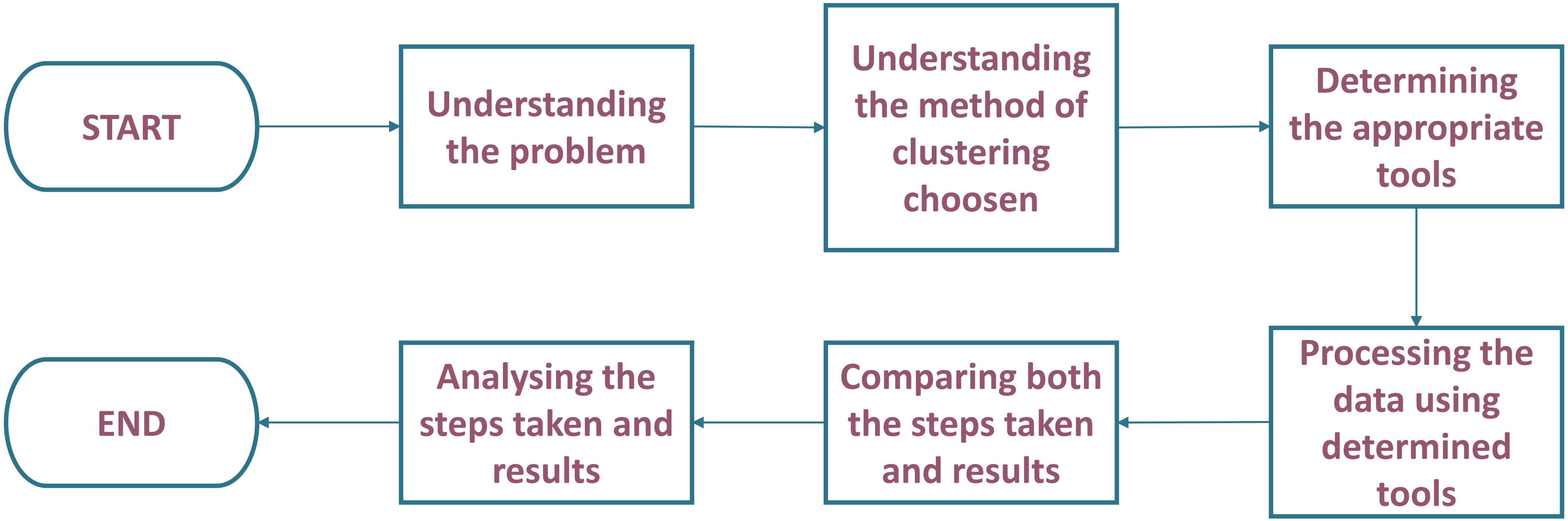
- To maintain product quality and the relationship between the company and suppliers, a supplier's performance evaluation is carried out every month
- The supplier performance target is set at 99.5%
- There are still suppliers who have not reached the target
- The process of grouping suppliers based on performance that has been applied by company X still has not fully considered every criterion of the performance appraisal

Problem Identification

Analyzing the difference in both steps to conduct & the result gained from utilizing MATLAB and Python as Data Analytics tools by using K-Means Clustering Methods.

Objectives

- To gain understanding how clustering may help the Company X in managing its suppliers.
- To find the characteristics of supplier with similar performances.
- To gain understanding how different tools can be used in processing same data and its impact



Methodology

Industrial and System Engineering Knowledge

That is being implemented during practical works

Statistics

- Understanding how the data should be made so it is apple-to-apple by using normalization
- Understanding how the data should be visualized
- Understanding how the data is to be interpreted

Quality Control Engineering

- Understanding what are the quality to be inspected
- Understanding what are the problems in each cluster
- Understanding why the supplier performance should be improved

Data Mining

- Understanding the methodology and concept of K-Means Clustering
- Understanding why Silhouette Index and SSE should be used to evaluate the model
- Understanding the difference between Data Analytic Tools

Comparison Between Tools for Data Analytics

Comparison between Matlab, Python and R as data processing tools in Data Analytics

Matlab

Developer : Mathworks (1984)

- Data visualization is more attractive
- Commercial product
- Excellent when used for linear matrix algebra
- Bias is hard to find
- Iteration repetition takes more time

Phyton

Developer: Guido van Rossum (1990)

- Used both for developing platform and data analytics
- Open Source and has large collection libraries
- Data is processed very fast
- Standard data visualization
- General purpose language and less difficult than R and MATLAB

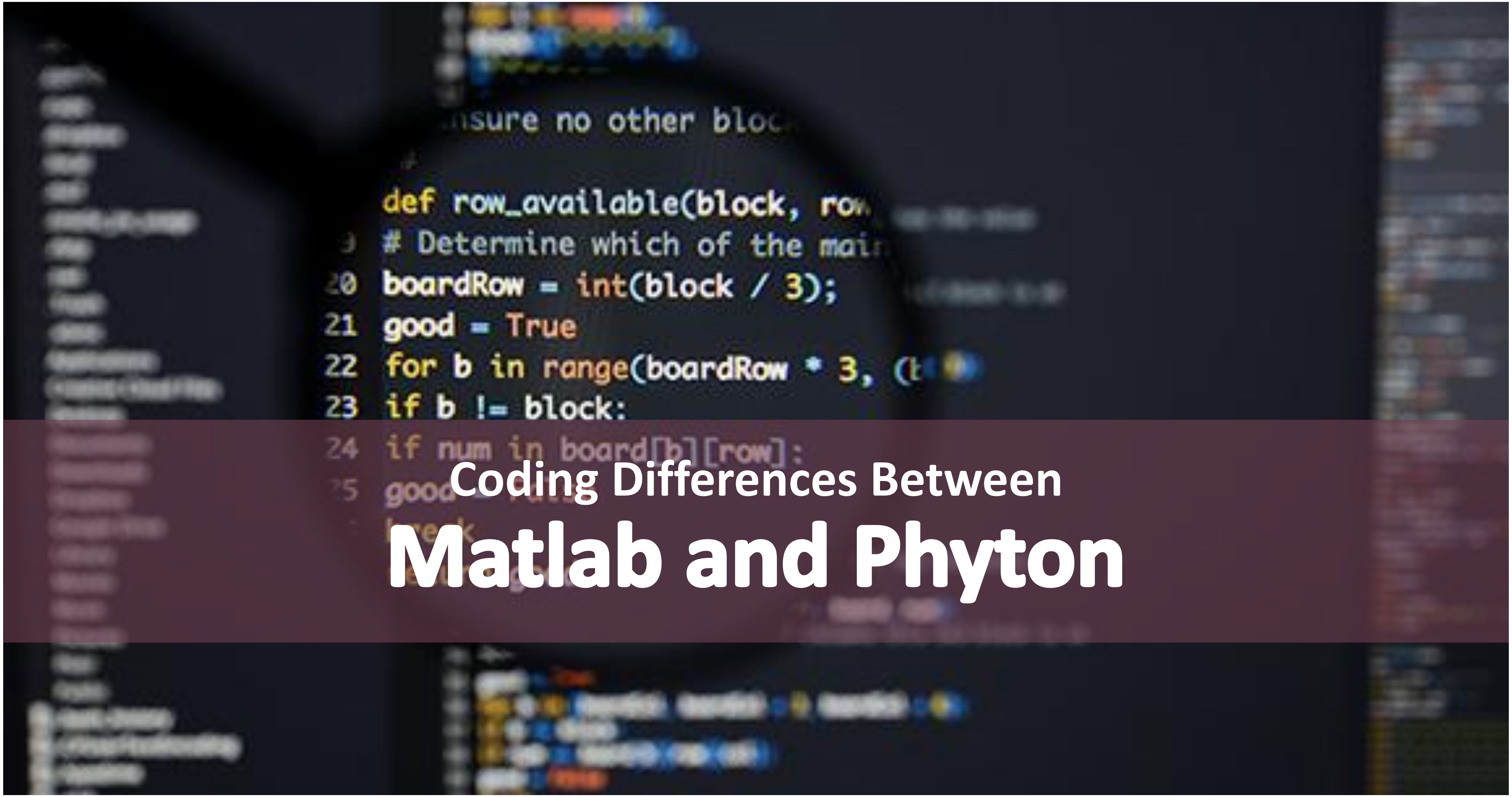
R

Developer : R Core Team (1993)

- Only used in statistical analysis
- Open source and has large collection of packages
- Connecting with source data is easier
- Data processing takes more time
- a language designed by statisticians, and difficult for understanding than MATLAB and Python

Coding Differences Between

Matlab and Phyton



Understanding Data

Supplier Name	Month	Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit
A	Apr-19	14	0	0	0	0	0	0	0	0
B	Apr-19	409	0	0	0	0	0	0	0	0
C	Apr-19	2066	0	2	3	1	11	1	0	0
D	Apr-19	0	0	0	0	0	0	0	0	0
E	Apr-19	347	0	0	1	0	0	0	0	0
….	….	….	….	….	….	….	….	….	….	….
BF	Feb-20	0	0	0	0	0	0	0	0	0.03
BG	Feb-20	75	0	0	0	0	0	0	0	0
BH	Feb-20	2517	0	0	0	3	11	17	0	0
BI	Feb-20	1104	0	0	0	0	10	4	57	0.01
BJ	Feb-20	31	0	0	0	0	0	0	0	0

Quality Dimensions	Indicators
Performance	Lot Inspected
	Incoming rejection (NRS) type A, B, and C
	In-process rejection (QCI) type A, B, and C
	Special used parts (SAR)
Responsiveness	Demerit

The data used for data analysis or commonly referred as a dataset consists of ***682 data from 62 suppliers*** owned by Company X that was collected from ***April 2019 to February 2020.***

It is necessary to know the value of each attribute owned by the dataset. The value of each attribute needs to be known whether it is nominal, ordinal, interval, and ratio data types. This is necessary for the data pre-processing stage.

Data Preprocessing

Data pre-processing is done to get more accurate results in the use of data mining techniques. Pre-processing is also useful for reducing computation time.

Data cleaning : filling missing value (blank data), in the provided dataset there is no missing value because it has been confirmed to company X

Data Transformation: Very important for datasets before entering data processing, data transformation methods are standardization, centering, and scaling. Scaling is the procedure of changing the data so that it is on a scale of [0,1] or what is commonly called the **min-max normalization**.

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \times (UL - LL) + LL$$

Matlab

```
function dataNew = scale_normalization(data,LL,UL)
%input:
%data = data that will be pre-processed in the format of m x
n
%m = amount of data, n = data dimension
%LL is the lower limit, UL is the upper limit

[dataMax]=max(data);
[dataMin]=min(data);
[R,C]=size(data);
dataNew = (data-ones(R,1)*dataMin).*(ones(R,1)*(UL-
LL)*(ones(1,C)./(dataMax-dataMin)))+LL;
```

Supplier Name	Month	Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit
A	Apr-19	0.0041	0	0	0	0	0	0	0	0
B	Apr-19	0.1189	0	0	0	0	0	0	0	0
C	Apr-19	0.6006	0	0.4	0.375	0.167	0.423	0.059	0	0
D	Apr-19	0	0	0	0	0	0	0	0	0
E	Apr-19	0.1009	0	0	0.125	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
BF	Feb-20	0	0	0	0	0	0	0	0	0
BG	Feb-20	0.0218	0	0	0	0	0	0	0	0
BH	Feb-20	0.7317	0	0	0	0.5	0.423	1	0	0
BI	Feb-20	0.3209	0	0	0	0	0.385	0.235	0.147	0.333
BJ	Feb-20	0	0	0	0	0	0	0	0	0

Python

```
import numpy as np
import pandas as pd

from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import silhouette_score

import matplotlib.pyplot as plt
```

```
#Input Data
input_data = pd.read_excel("Data_SRM.xlsx")
input_data.head()
```

	Supplier Name	Month	Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit
0	NaN	NaT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	A	2019-04-01	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	B	2019-04-01	409.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	C	2019-04-01	2066.0	0.0	2.0	3.0	1.0	11.0	1.0	0.0	0.0
4	D	2019-04-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
In [4]: data_used = input_data.drop('Supplier Name', axis = 1)
        data_used = data_used.drop('Month', axis = 1)
        data_used = data_used.drop(0, axis = 0)
        data_used.head()
```

Out[4]:

	Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit
1	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	409.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2066.0	0.0	2.0	3.0	1.0	11.0	1.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	347.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

```
In [5]: #Mengubah Data Frame menjadi array
        x_array = np.array(data_used)
        print(x_array)

[[1.400e+01 0.000e+00 0.000e+00 ... 0.000e+00 0.000e+00 0.000e+00]
 [4.090e+02 0.000e+00 0.000e+00 ... 0.000e+00 0.000e+00 0.000e+00]
 [2.066e+03 0.000e+00 2.000e+00 ... 1.000e+00 0.000e+00 0.000e+00]
 ...
 [2.517e+03 0.000e+00 0.000e+00 ... 1.700e+01 0.000e+00 0.000e+00]
 [1.104e+03 0.000e+00 0.000e+00 ... 4.000e+00 5.700e+01 1.000e-02]
 [3.100e+01 0.000e+00 0.000e+00 ... 0.000e+00 0.000e+00 0.000e+00]]
```

```
In [28]: #Scaling dataset with MinMax (Standarisasi)
         scaler = MinMaxScaler()
         x_scaled = scaler.fit_transform(x_array)
         print(x_scaled)

[[0.00406977 0.         0.         ... 0.         0.         0.        ]
 [0.11889535 0.         0.         ... 0.         0.         0.        ]
 [0.6005814  0.         0.4        ... 0.05882353 0.         0.        ]
 ...
 [0.73168605 0.         0.         ... 1.         0.         0.        ]
 [0.32093023 0.         0.         ... 0.23529412 0.14652956 0.33333333]
 [0.00901163 0.         0.         ... 0.         0.         0.        ]]
```

```
#Array menjadi Data Frame
MinMax = pd.DataFrame(x_scaled,columns = ['Lot Inspected','NRS A','NRS B','NRS C',
                                          'QCI A', 'QCI B','QCI C','SAR','Demerit'])
MinMax.head()
```

	Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit
0	0.004070	0.0	0.0	0.000	0.000000	0.000000	0.000000	0.0	0.0
1	0.118895	0.0	0.0	0.000	0.000000	0.000000	0.000000	0.0	0.0
2	0.600581	0.0	0.4	0.375	0.166667	0.423077	0.058824	0.0	0.0
3	0.000000	0.0	0.0	0.000	0.000000	0.000000	0.000000	0.0	0.0
4	0.100872	0.0	0.0	0.125	0.000000	0.000000	0.000000	0.0	0.0

The results of normalization of matlab and python are the same when using MinMax scaling

Data Processing: K-Means Clustering

Clustering is a set of techniques used to partition data into groups, or clusters.
The K-Means Clustering method is a data processing method that clusters data in a particular cluster indicated by **Centroid as the center of the cluster**. Data is grouped based on the similarity of **data instances** while the number of clusters is predetermined as input.

Python

```
#K-Mean Clustering
Cluster=[]
k=[2,3,4,5,6,7,8,9,10]
j=[5,50,100]
for k in range (2,11):
    for y in j:
        Trial=KMeans(n_clusters=k, max_iter=y,init="random").fit(SRM)
        Predict=Trial.predict(SRM)
        P=print('Centroid of k = ',k,'\t with Maximum Iteration of ', y,'\n')
        print(pd.DataFrame(Trial.cluster_centers_,columns=['Lot Inspected','NRS A','NRS B',"NRS C",'QCI A',
                                                          'QCI B',"QCI C",'SAR','Demerit']),'\n')
        print('The Sum of Squared Error = ',Trial.inertia_)
        print('The Silhouette Score = ',silhouette_score(SRM,Predict),'\n','\n')
```

The K-Means Clustering as well as the Silhouette Index and SSE calculation can be conducted simultaneously with 5, 50, and 100 iterations.

MATLAB

1

```
function [cluster,centres] = kmeansa(k,data,niters)

%input:
%k = number of cluster
%data = data to be clustered
%niters = maximum iteration number
%Deskripsi

[ndata, data_dim]=size(data);
ncentres=k; %number of centres equals to number of cluster
if (ncentres > ndata)
    error('Too many clusters than data')
end

%determine random cluster centres
perm=randperm(ndata);
indpusat=perm(1:ncentres);
centres=data(indpusat,:);

%Loop utama
for n=1:niters
    %save old clusters
    old_centres=centres;
    %calculate distance between data and cluster centers
    d2=dist2(data,centres);
    %plot data to the nearest cluster
    [minvals,ind]=min(d2,[],2);
    post=accumarray(ind,1,[k,1]); %mencari banyak titik data
yg masuk kelas j
    cluster=ind;
```

2

```
    for j=1:ncentres
        if(post(j) > 0)
            centres(j,:)=sum(data(find(ind==j),:))/post(j);
 %cari pusat baru
        end
    end

    change=sum(sum(abs(old_centres-centres)));
    if change < 1e-10 %is it convergent
        break
    end
end


function d2=dist2(data,centres)
ndata=size(data,1);
ncentres=size(centres,1);


d2=zeros(ndata,ncentres);
for j=1:ncentres
    d2(:,j)=sum((data-repmat(centres(j,:),ndata,1)).^2,2);
end
```

Evaluation Model

Phyton

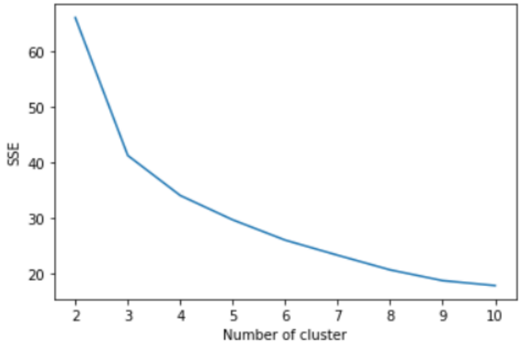
```
In [7]: #Elbow Criterion (SSE)

        sse = {}
        for k in range(2,11):
            kmeans = KMeans(n_clusters=k, max_iter=1000).fit(x_scaled)
            sse[k] = kmeans.inertia_

        print(sse)

{2: 66.0303590972852, 3: 41.223189701799654, 4: 34.02145010515328, 5: 29.64008914075026, 6: 25.990239782125688, 7: 23.269124252
165394, 8: 20.631929525201112, 9: 18.698709251765468, 10: 17.80814778007201}
```

```
In [8]: plt.figure()
        plt.plot(list(sse.keys()), list(sse.values()))
        plt.xlabel("Number of cluster")
        plt.ylabel("SSE")
        plt.show()
```



- **Elbow Method Evaluation**

The quality of the cluster assignments is determined by computing the sum of the squared error (SSE) after the centroids converge or match the previous iteration's assignment. The SSE is defined as the sum of the squared Euclidean distances of each point to its closest centroid. Since this is a measure of error, the objective of k-means is to try to minimize this value.

Using SSE to find out the optimal k. In evaluation model using **Elbow Method, optimal the number k of clusters is obtained when the difference in SSE value from the previous k is significantly reduced**

Silhouette Index Evaluation

Silhouette index is a parameter to evaluate whether the data placement in the cluster is correct or not. **The silhouette method computes silhouette coefficients of each point that measure how much a point is similar to its own cluster compared to other clusters.** If the index is close to 1, then the data placement in the cluster is accurate, while the farther from 1, the data placement in the cluster is not accurate

Phyton

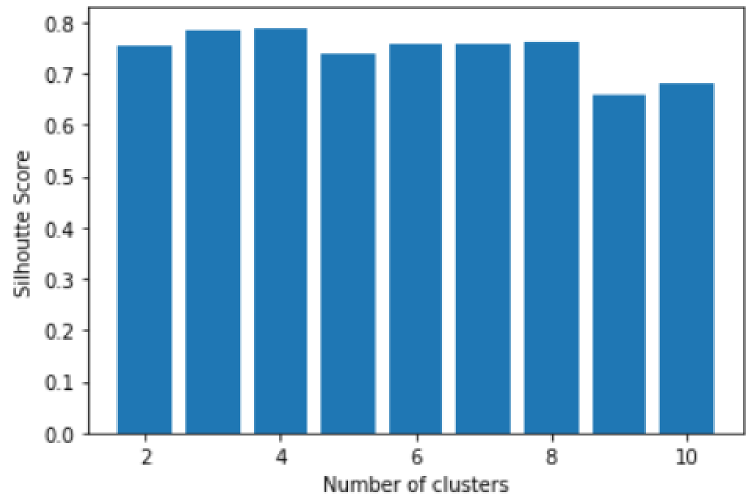
In [29]:
```
#Silhoutte Index

silhouette_scores = []

for k in range(2,11):
    silhouette_scores.append(
        silhouette_score(x_scaled, KMeans(n_clusters=k, max_iter=1000).fit_predict(x_scaled)))
k = [2,3,4,5,6,7,8,9,10]
print(k, silhouette_scores)
```

```
[2, 3, 4, 5, 6, 7, 8, 9, 10] [0.7529024625347353, 0.7826927621532846, 0.7887233888440417, 0.7663934685698124, 0.758103182538239
6, 0.7629593485980645, 0.7662219566487657, 0.6692631830748287, 0.676298712756818]
```

In [19]:
```
plt.bar(k,silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 10)
plt.ylabel('Silhoutte Score', fontsize = 10)
plt.show()
```



MATLAB

```
[s,h] = silhouette(dataNew,cluster,'sqEuclidean');
```

Differences in Data Processing Results Between

Matlab and Phyton

Cluster Centroid

Matlab

K = 2												
Iteration	Cluster	Centroid									Silhouette Index	SSE
		Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit		
5	1	0.3276	0.0102	0.0959	0.1339	0.0697	0.1593	0.1711	0.1818	0.3571	0.7868	67.4115
	2	0.0344	0.0000	0.0120	0.0139	0.0023	0.0059	0.0161	0.0028	0.0000		
50	1	0.3380	0.0109	0.0826	0.1168	0.0743	0.1664	0.1771	0.1898	0.3768	0.7927	67.2450
	2	0.0358	0.0000	0.0149	0.0178	0.0023	0.0063	0.0167	0.0034	0.0006		
100	1	0.0358	0.0000	0.0149	0.0178	0.0023	0.0063	0.0167	0.0034	0.0006	0.7927	67.2450
	2	0.3380	0.0109	0.0826	0.1168	0.0743	0.1664	0.1771	0.1898	0.3768		

K = 3												
Iteration	Cluster	Centroid									Silhouette Index	SSE
		Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit		
5	1	0.4913	0.0000	0.1367	0.1813	0.1194	0.2429	0.2324	0.2941	0.0222	0.8568	41.3294
	2	0.0340	0.0000	0.0123	0.0164	0.0014	0.0057	0.0168	0.0027	0.0023		
	3	0.0799	0.0286	0.0286	0.0214	0.0048	0.0319	0.0672	0.0059	0.9238		
50	1	0.0799	0.0286	0.0286	0.0214	0.0048	0.0319	0.0672	0.0059	0.9238	0.8634	41.2232
	2	0.5182	0.0000	0.1148	0.1713	0.1235	0.2607	0.2538	0.3195	0.0247		
	3	0.0361	0.0000	0.0155	0.0190	0.0022	0.0065	0.0171	0.0034	0.0022		
100	1	0.0799	0.0286	0.0286	0.0214	0.0048	0.0319	0.0672	0.0059	0.9238	0.8634	41.2232
	2	0.0361	0.0000	0.0155	0.0190	0.0022	0.0065	0.0171	0.0034	0.0022		
	3	0.5182	0.0000	0.1148	0.1713	0.1235	0.2607	0.2538	0.3195	0.0247		

Python

k=2											Silhouette Index	SSE
Iteration	Cluster	Centroid										
		Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit		
5	1	0.07637	0.00000	0.02380	0.03169	0.01237	0.02770	0.03682	0.02977	0.00412	0.75290	66.03036
	2	0.07989	0.02857	0.02857	0.02143	0.00476	0.03187	0.06723	0.00588	0.92381		
50	1	0.03719	0.00000	0.01518	0.01897	0.00253	0.00707	0.01756	0.00360	0.00056		
	2	0.33877	0.01124	0.08315	0.11236	0.07491	0.16681	0.17713	0.19474	0.38951		
100	1	0.33800	0.01087	0.08261	0.11685	0.07428	0.16639	0.17711	0.18976	0.37681	0.74176	67.24501
	2	0.03578	0.00000	0.01492	0.01780	0.00226	0.00632	0.01675	0.00340	0.00057		

k=3											Silhouette Index	SSE
Iteration	Cluster	Centroid										
		Lot Inspected	NRS A	NRS B	NRS C	QCI A	QCI B	QCI C	SAR	Demerit		
5	1	0.51085	0.00000	0.11786	0.17188	0.12202	0.25343	0.24475	0.31294	0.02381	0.78269	41.22319
	2	0.07989	0.02857	0.02857	0.02143	0.00476	0.03187	0.06723	0.00588	0.92381		
	3	0.03520	0.00000	0.01489	0.01840	0.00197	0.00631	0.01712	0.00294	0.00226		
50	1	0.07989	0.02857	0.02857	0.02143	0.00476	0.03187	0.06723	0.00588	0.92381		
	2	0.51820	0.00000	0.11482	0.17130	0.12346	0.26068	0.25381	0.31948	0.02469		
	3	0.03613	0.00000	0.01551	0.01897	0.00225	0.00649	0.01706	0.00339	0.00225		
100	1	0.51820	0.00000	0.11482	0.17130	0.12346	0.26068	0.25381	0.31948	0.02469		
	2	0.07989	0.02857	0.02857	0.02143	0.00476	0.03187	0.06723	0.00588	0.92381		
	3	0.03613	0.00000	0.01551	0.01897	0.00225	0.00649	0.01706	0.00339	0.00225		

Model of the Matlab and Python cluster

Coding

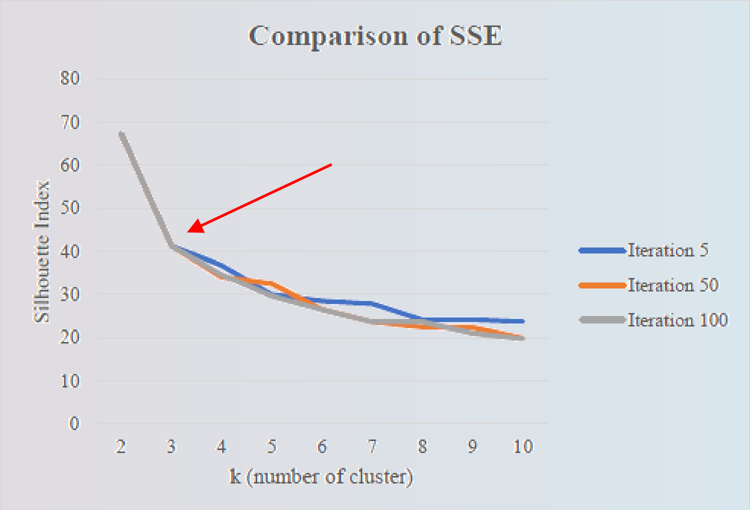
- Data ***processing tends to take longer*** in MATLAB, so iterations tend to be done in small amounts, whereas Python processes faster
- MATLAB use C/C++ language, while Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.
- There are also ***fewer functions required*** in Python compared to MATLAB

Results

- In MATLAB there is a ***need to check*** whether the ***centroid is convergent*** in the end of clustering while in Python the function will ensure the centroid is ***convergent as default***
- In ***clusters below 3*** and ***iterations below 50***, the ***centroids*** of the two tools are ***different***. The difference gets bigger with decreasing value of k and iteration.
- With the ***increasing number iteration***, the changes in the ***centroid gaps*** between each iteration should ***decrease*** since the ***stopping criteria for the K-Means is for minimizing the SSE*** given the maximum iteration to be conducted

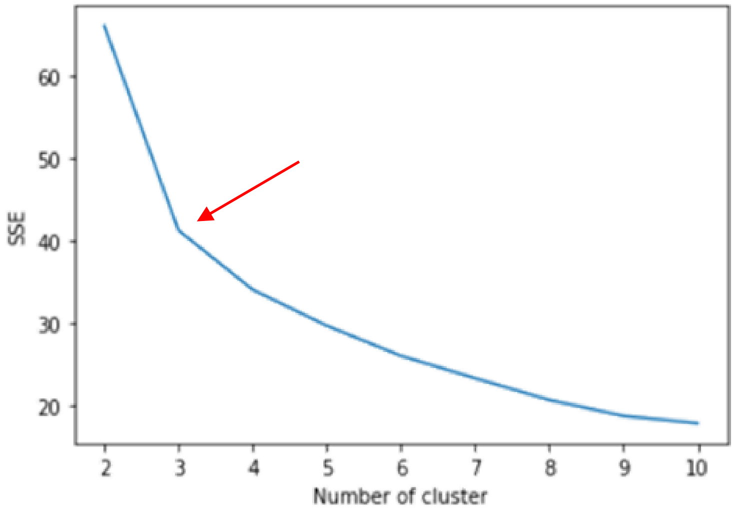
Evaluation for k clusters

Matlab



Matlab						
k	SSE			Silhouette Index		
	5	50	100	5	50	100
2	67.411	67.245	67.245	0.786	0.792	0.792
3	41.329	41.223	41.223	0.856	0.863	0.863
4	36.748	34.026	34.593	0.853	0.872	0.872
5	29.905	32.474	29.647	0.754	0.823	0.814
6	28.525	26.573	26.494	0.656	0.787	0.794
7	27.891	23.63	23.63	0.698	0.772	0.772
8	24.119	22.42	23.643	0.611	0.742	0.765
9	24.104	22.303	21.014	0.609	0.729	0.713
10	23.717	19.923	19.755	0.574	0.669	0.669

Python



Phyton						
K	SSE			Silhouette Index		
	5	50	100	5	50	100
2	67.268	66.03	66.03	0.742	0.753	0.753
3	41.223	41.223	41.223	0.783	0.783	0.783
4	36.466	34.026	34.021	0.759	0.786	0.789
5	30.361	29.558	29.551	0.756	0.747	0.747
6	28.408	25.969	25.962	0.692	0.758	0.752
7	26.407	23.122	23.122	0.691	0.756	0.755
8	22.452	20.915	20.835	0.594	0.686	0.686
9	23.994	19.625	18.862	0.577	0.763	0.646
10	19.404	17.93	17.78	0.559	0.65	0.651

How about 1000 iterations?

PYTHON		
k	SSE	Silhouette Index
	1000	
2	66.03	0.753
3	41.223	0.782
4	34.021	0.788
5	29.64	0.738
6	25.99	0.758
7	23.269	0.755
8	20.632	0.762
9	18.699	0.656
10	17.808	0.68

Model of the Cluster based on Model Evaluation

Elbow Method

In matlab, with 5, 50, and 100 iterations have different results, but in the end the optimal k is owned by k = 3 because it has the highest SSE reduction of k = 2.

In python, with 5, 50, 100, and 1000 iterations, it is found that k is also optimal at k = 3 because it has the highest SSE reduction of k = 2.

Matlab									
k	SSE						Silhouette Index		
	5	Gap	50	Gap	100	Gap	5	50	100
2	67.411		67.245		67.245		0.786	0.792	0.792
3	41.329	-26.082	41.223	-26.022	41.223	-26.022	0.856	0.863	0.863
4	36.748	-4.581	34.026	-7.197	34.593	-6.63	0.853	0.872	0.872
5	29.905	-6.843	32.474	-1.552	29.647	-4.946	0.754	0.823	0.814
6	28.525	-1.38	26.573	-5.901	26.494	-3.153	0.656	0.787	0.794
7	27.891	-0.634	23.63	-2.943	23.63	-2.864	0.698	0.772	0.772
8	24.119	-3.772	22.42	-1.21	23.643	0.013	0.611	0.742	0.765
9	24.104	-0.015	22.303	-0.117	21.014	-2.629	0.609	0.729	0.713
10	23.717	-0.387	19.923	-2.38	19.755	-1.259	0.574	0.669	0.669

Matlab

- ## Silhouette Index

In matlab, with 5, 50, and 100 iterations, optimal k is owned by k = 4.

In python, with 5, 100 and 1000 iterations, it is found that the optimal k is k = 4 because it is closest to the value 1.

Even though k = 4 has a value closest to 1, at k = 3 it also has a value close to 1 and is not much different from k = 4. So that k = 3 can still be tolerated.

Phyton									
k	SSE						Silhouette Index		
	5	Gap	50	Gap	100	Gap	5	50	100
2	67.268		66.03		66.03		0.742	0.753	0.753
3	41.223	-26.045	41.223	-24.807	41.223	-24.807	0.783	0.783	0.783
4	36.466	-4.757	34.026	-7.197	34.021	-7.202	0.759	0.786	0.789
5	30.361	-6.105	29.558	-4.468	29.551	-4.47	0.756	0.747	0.747
6	28.408	-1.953	25.969	-3.589	25.962	-3.589	0.692	0.758	0.752
7	26.407	-2.001	23.122	-2.847	23.122	-2.84	0.691	0.756	0.755
8	22.452	-3.955	20.915	-2.207	20.835	-2.287	0.594	0.686	0.686
9	23.994	1.542	19.625	-1.29	18.862	-1.973	0.577	0.763	0.646
10	19.404	-4.59	17.93	-1.695	17.78	-1.082	0.559	0.65	0.651

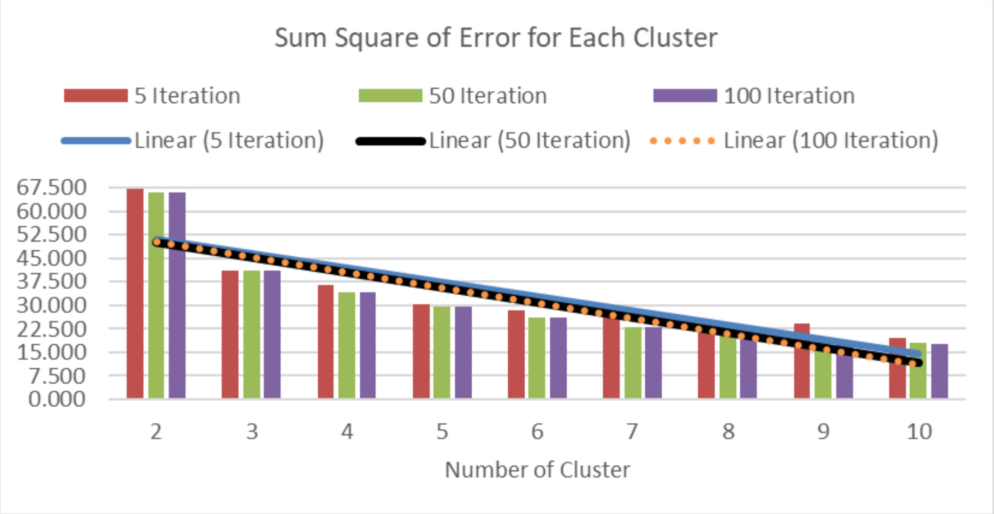
PYTHON			
k	SSE		Silhouette Index
	1000	Gap	1000
2	66.03		0.753
3	41.223	-24.807	0.782
4	34.021	-7.202	0.788
5	29.64	-4.381	0.738
6	25.99	-3.65	0.758
7	23.269	-2.721	0.755
8	20.632	-2.637	0.762
9	18.699	-1.933	0.656
10	17.808	-0.891	0.68

Python

Cluster k = 3

From the evaluation results, both the Silhouette and the Elbow Method produce the optimal k for the supplier quality characteristic data, which is 3 clusters.

MATLAB	Cluster 1	Cluster 2	Cluster 3	PHYTON	Cluster 1	Cluster 2	Cluster 3
Cluster Index	0	1	2	**Cluster Index**	0	1	2
Cover %	5%	8%	87%	**Cover %**	86.95%	5.13%	7.92%
Quality Indicators				**Quality Indicators**			
Indicator	**Average**	**Average**	**Average**	**Indicator**	**Average**	**Average**	**Average**
Lot Inspected	274.829	1782.593	124.3	**Lot Inspected**	124.3002	274.8286	1782.5926
NRS A	0.0286	0	0	**NRS A**	0.0000	0.0286	0.0000
NRS B	0.1429	0.5741	0.0776	**NRS B**	0.0776	0.1429	0.5741
NRS C	0.1714	1.3704	0.1518	**NRS C**	0.1518	0.1714	1.3704
QCI A	0.0286	0.7407	0.0135	**QCI A**	0.0135	0.0286	0.7407
QCI B	0.8286	6.7778	0.1686	**QCI B**	0.1686	0.8286	6.7778
QCI C	1.1429	4.3148	0.2901	**QCI C**	0.2901	1.1429	4.3148
SAR	2.2857	124.2778	1.317	**SAR**	1.3170	2.2857	124.2778
Demerit	0.0277	0.0007	0.0001	**Demerit**	0.0001	0.0277	0.0007



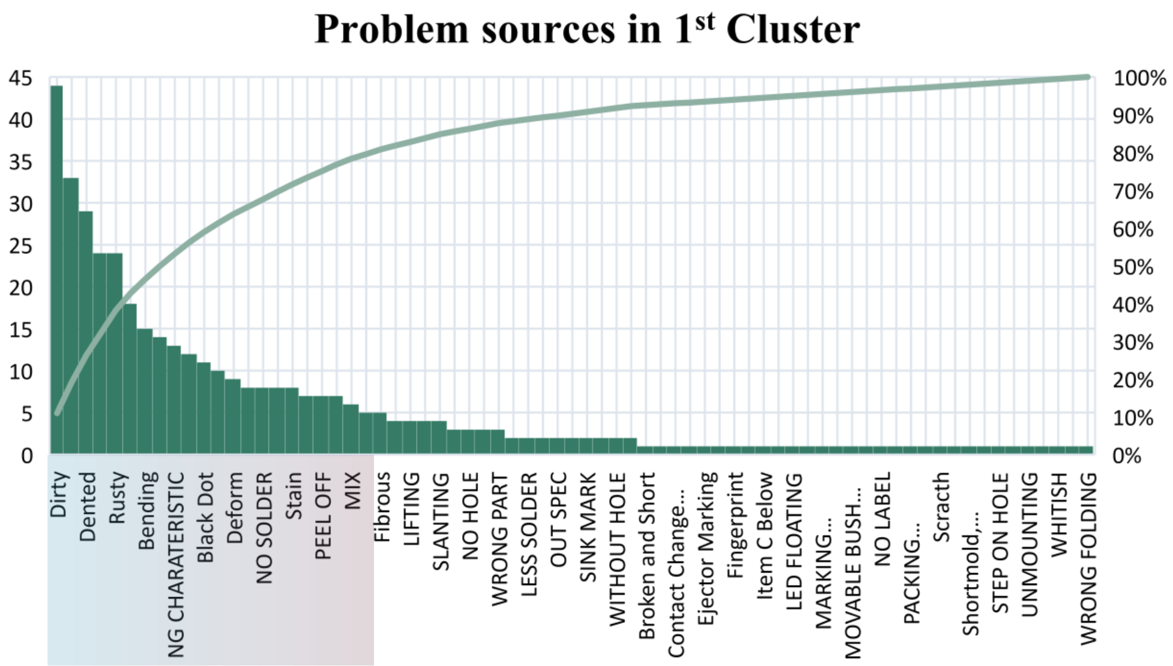
MATLAB	Cluster 1	Cluster 2	Cluster 3	PHYTON	Cluster 1	Cluster 2	Cluster 3
Cluster Index	0	1	2	Cluster Index	0	1	2
Cover %	5%	8%	87%	Cover %	86.95%	5.13%	7.92%
Quality Indicators				Quality Indicators			
Indicator	Average	Average	Average	Indicator	Average	Average	Average
Lot Inspected	274.829	1782.593	124.3	Lot Inspected	124.3002	274.8286	1782.5926
NRS A	0.0286	0	0	NRS A	0.0000	0.0286	0.0000
NRS B	0.1429	0.5741	0.0776	NRS B	0.0776	0.1429	0.5741
NRS C	0.1714	1.3704	0.1518	NRS C	0.1518	0.1714	1.3704
QCI A	0.0286	0.7407	0.0135	QCI A	0.0135	0.0286	0.7407
QCI B	0.8286	6.7778	0.1686	QCI B	0.1686	0.8286	6.7778
QCI C	1.1429	4.3148	0.2901	QCI C	0.2901	1.1429	4.3148
SAR	2.2857	124.2778	1.317	SAR	1.3170	2.2857	124.2778
Demerit	0.0277	0.0007	0.0001	Demerit	0.0001	0.0277	0.0007

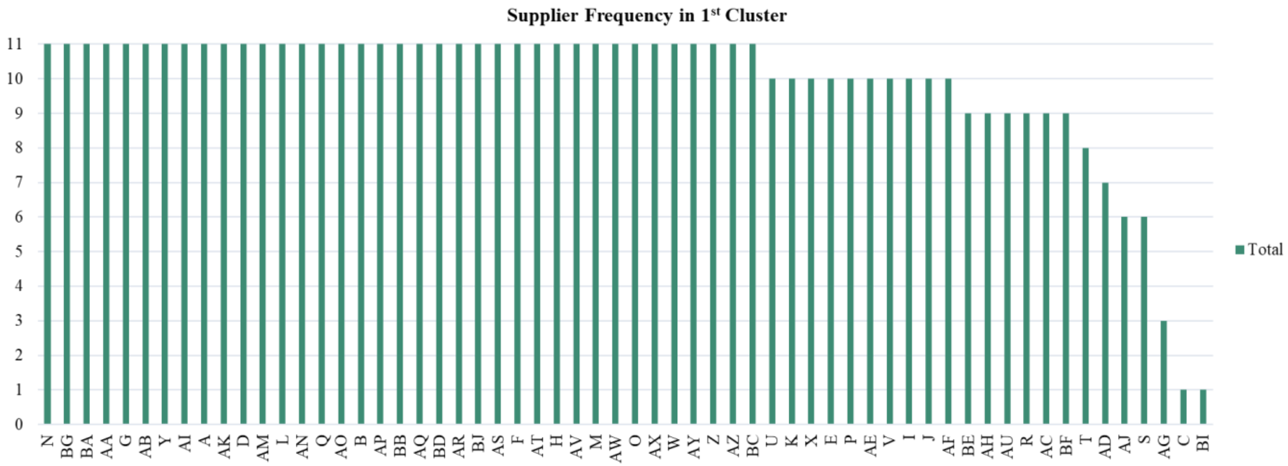

Cluster Analysis

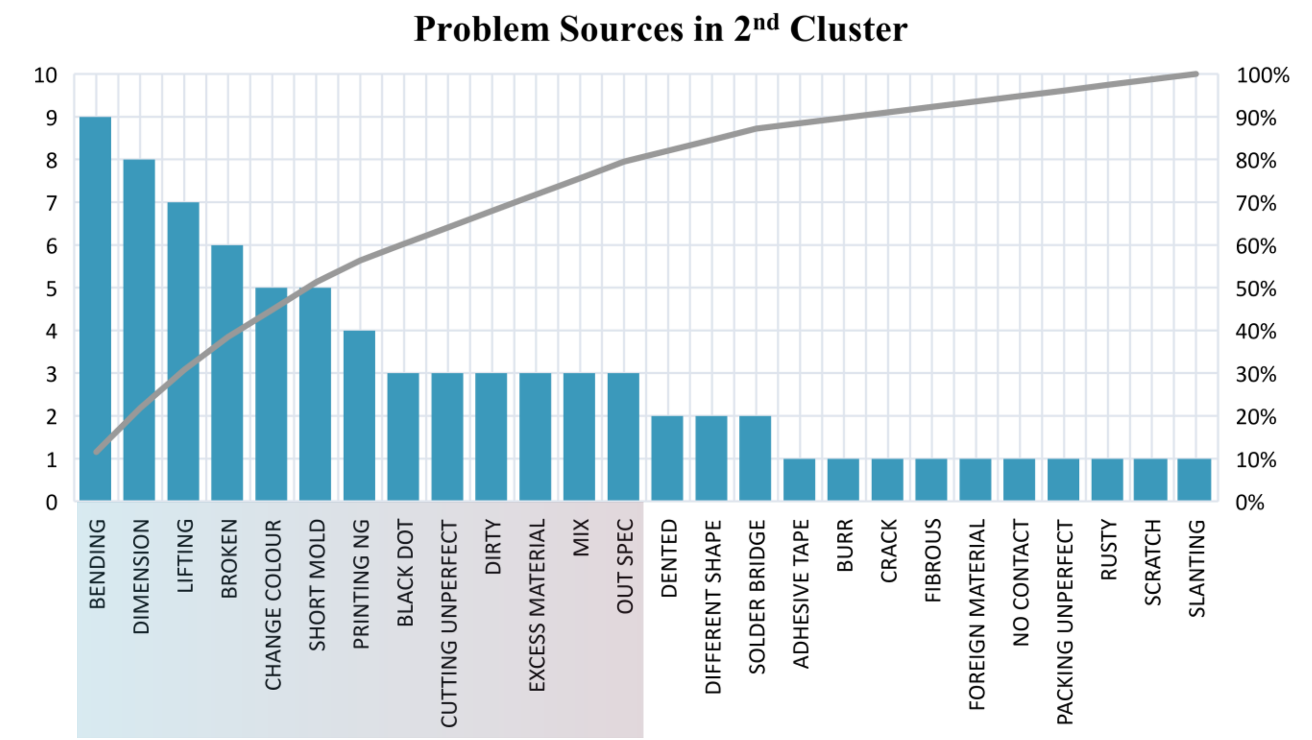
	Kluster 1	Kluster 2	Kluster 3
Index Kluster	0	1	2
Cover %	86.95%	5.13%	7.92%
Aspek Kualitas			
Aspek	Rata-rata	Rata-rata	Rata-rata
Lot Inspected	124.3002	274.8286	1782.5926
NRS A	0.0000	0.0286	0.0000
NRS B	0.0776	0.1429	0.5741
NRS C	0.1518	0.1714	1.3704
QCI A	0.0135	0.0286	0.7407
QCI B	0.1686	0.8286	6.7778
QCI C	0.2901	1.1429	4.3148
SAR	1.3170	2.2857	124.2778
Demerit	0.0001	0.0277	0.0007

Characteristics of 1st Cluster

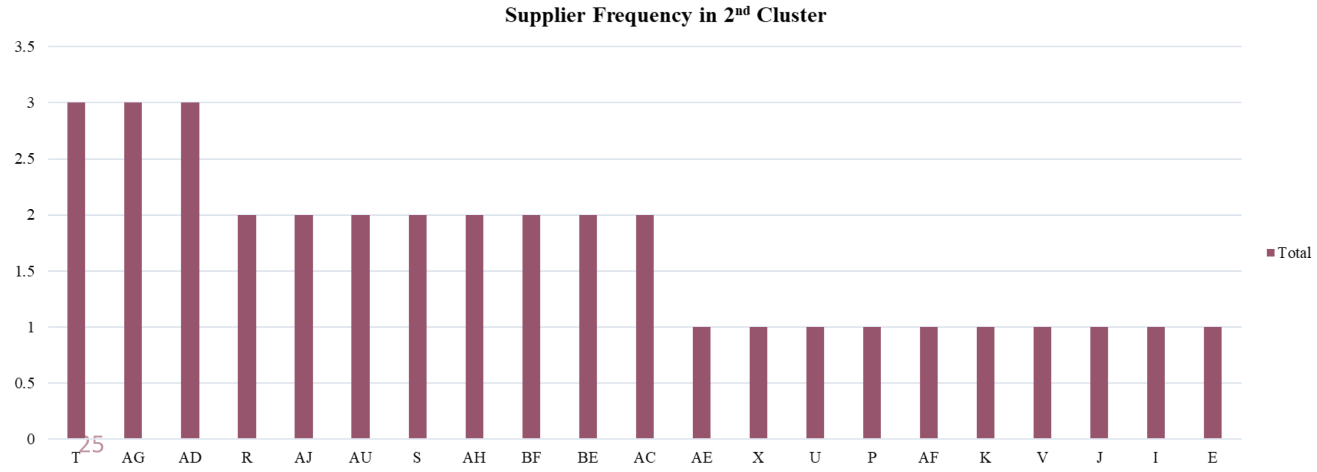
- Has the ***least number of lots inspected*** compared to other clusters
- Has the least number of claims from the quality indicators of NRS A to Demerit when compared to other clusters
- A total of ***593 data from 682*** supplier data entered into this cluster
- 60 out of 62 suppliers are included in this cluster
- Of the 60 suppliers, there are ***37*** suppliers who always enter this cluster every month
- This cluster has ***70 problem sources*** in total. In the pareto chart, the problem sources in parts or components from this cluster that contribute up to 80% of the ***problems are dirty, dented, rusty, bending, NG characteristic, black dot, deform, no solder, stain, peel off, and mix***





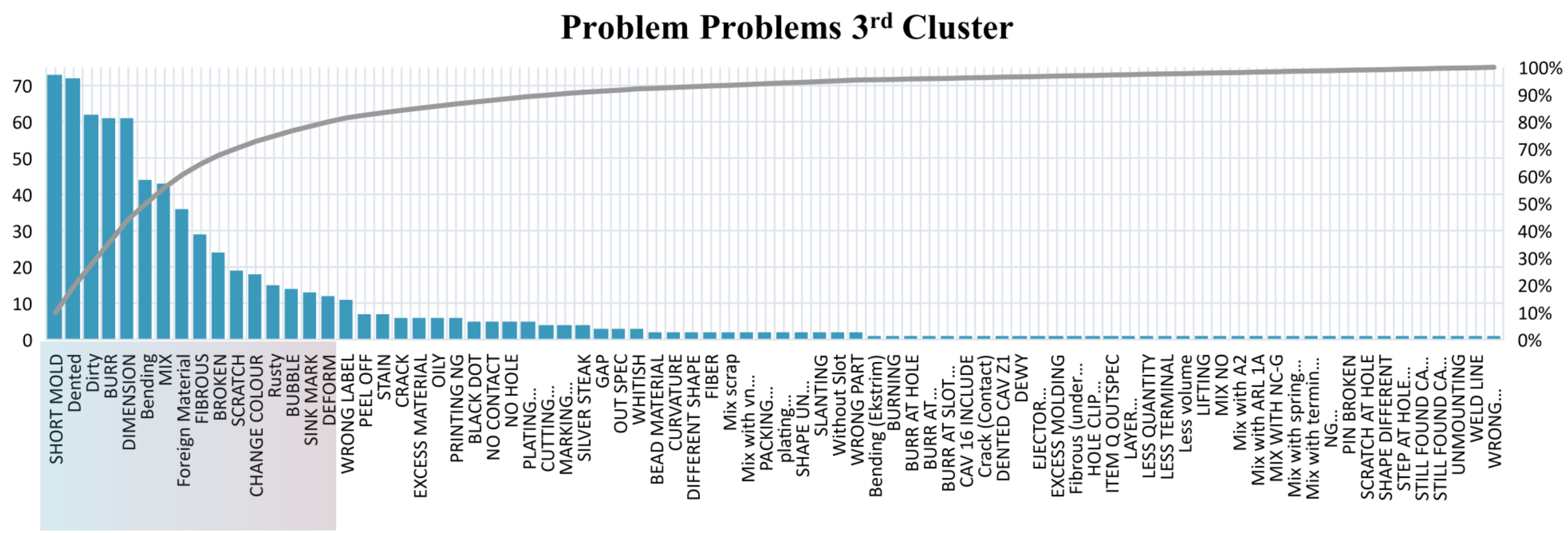


	Kluster 1	Kluster 2	Kluster 3
Index Kluster	0	1	2
Cover %	86.95%	5.13%	7.92%
Aspek Kualitas			
Aspek	Rata-rata	Rata-rata	Rata-rata
Lot Inspected	124.3002	274.8286	1782.5926
NRS A	0.0000	0.0286	0.0000
NRS B	0.0776	0.1429	0.5741
NRS C	0.1518	0.1714	1.3704
QCI A	0.0135	0.0286	0.7407
QCI B	0.1686	0.8286	6.7778
QCI C	0.2901	1.1429	4.3148
SAR	1.3170	2.2857	124.2778
Demerit	0.0001	0.0277	0.0007

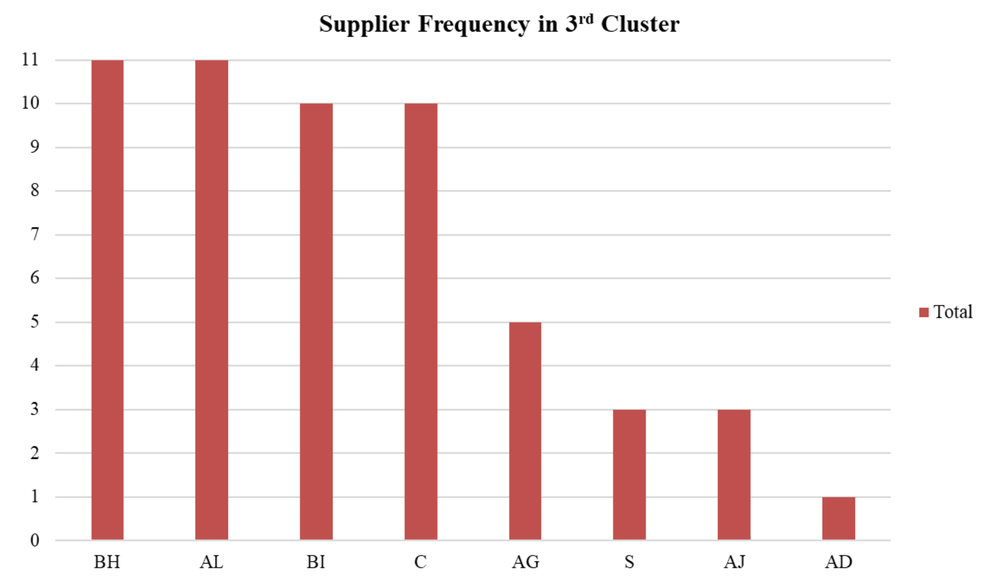


Characteristics of 2nd Cluster

- Has the highest NRS A and Demerit values compared to other clusters
- A total of ***35 data from 682*** supplier data entered into this cluster
- Of the 62 suppliers, 21 are included in this cluster
- Has the least number of problem sources when compared to other clusters, there are ***26 problems*** in total.
- In the pareto chart, the problem sources in parts or components from this cluster that contribute up to 80% of the problems are ***bending, dimension, lifting, broken, change color, short mold, NG printing, black dot, cutting unperfect, dirty, excess material, mix, and out spec***.



	Kluster 1	Kluster 2	Kluster 3
Index Kluster	0	1	2
Cover %	86.95%	5.13%	7.92%
Aspek Kualitas			
Aspek	Rata-rata	Rata-rata	Rata-rata
Lot Inspected	124.3002	274.8286	1782.5926
NRS A	0.0000	0.0286	0.0000
NRS B	0.0776	0.1429	0.5741
NRS C	0.1518	0.1714	1.3704
QCI A	0.0135	0.0286	0.7407
QCI B	0.1686	0.8286	6.7778
QCI C	0.2901	1.1429	4.3148
SAR	1.3170	2.2857	124.2778
Demerit	0.0001	0.0277	0.0007



Characteristics of 3rd Cluster

- Has the ***highest average number of lots inspected***
- Has the highest number of claims from the quality indicators of NRS B, NRS C, QCI A, QCI B, QCI C, and SAR when compared to other clusters.
- There are ***8 out of 62 suppliers*** included in this cluster and ***2 of them are always included in this cluster***
- Has the ***highest number of problem sources*** when compared to other clusters, there are 80 problems in total.
- In the pareto chart, the problem sources in parts or components from this cluster that contribute up to 80% of the problems are ***short mold, dented, dirty, burr, dimension, bending, mix, foreign material, fibrous, broken, scratch, change color, rusty, bubble, sink mark, and deform***.

Cluster Performance Comparison

The Initial Cluster is a cluster based on previous data processing using MATLAB, while the Final Cluster is data processing that we do using Python.

Initial Cluster

MATLAB	Cluster 1	Cluster 2	Cluster 3
Cluster Index	0	1	2
Cover %	5%	8%	87%
Quality Indicators			
Indicator	Average	Average	Average
Lot Inspected	274.829	1782.593	124.3
NRS A	0.0286	0	0
NRS B	0.1429	0.5741	0.0776
NRS C	0.1714	1.3704	0.1518
QCI A	0.0286	0.7407	0.0135
QCI B	0.8286	6.7778	0.1686
QCI C	1.1429	4.3148	0.2901
SAR	2.2857	124.2778	1.317
Demerit	0.0277	0.0007	0.0001

Final Cluster

PHYTON	Cluster 1	Cluster 2	Cluster 3
Cluster Index	0	1	2
Cover %	86.95%	5.13%	7.92%
Quality Indicators			
Indicator	Average	Average	Average
Lot Inspected	124.3002	274.8286	1782.5926
NRS A	0.0000	0.0286	0.0000
NRS B	0.0776	0.1429	0.5741
NRS C	0.1518	0.1714	1.3704
QCI A	0.0135	0.0286	0.7407
QCI B	0.1686	0.8286	6.7778
QCI C	0.2901	1.1429	4.3148
SAR	1.3170	2.2857	124.2778
Demerit	0.0001	0.0277	0.0007

Comparison of Claim Data Processing Results

Comparison of claims data processing based on clustering results using Matlab and Python

- During data processing, inconsistent data were found between claims and supplier quality data per month
- The performance value of each cluster is similar but there are differences in the labeling of the clusters

Clustering Result Using MATLAB

- ***For all cluster, the problem source with the highest frequency is due to dimension.***
- ***Cluster with 87% coverage has the highest problem sources while cluster with 8% coverage is the lowest.***

Clustering Result Using Python

- ***The problem source with the highest frequency for all cluster is different, dirty for 1st cluster, bending for 2nd cluster, and short mold for 3rd cluster.***
- ***Cluster with 8% coverage has the highest problem sources while cluster with 5% coverage is the lowest.***

CONCLUSION

Implementation of Industrial and System Engineering knowledge during practical work are knowledge of ***Statistics***, knowledge of ***Quality Control Engineering, and Data Mining.***

Supplier product of Company X will significantly determine Company X product's quality, but the current condition is that there are ***many suppliers who has yet to achieve the target***

Database about the supplier inspection data can be used to ***determine the type of training and next actions to improve the performance of each suppliers based on the characteristics of each clusters.***

Phyton has the advantage of ***faster data processing, ensuring the centroid are converged, and will need less function*** to conduct K-means Clustering than MATLAB.

CONCLUSION – Cluster Characteristics

The 1st cluster (87% coverage) has ***the least average number of lots inspected; has the least number of claims from the quality indicators of NRS A to Demerit; and has 70 problem sources in total with the most occurred problem is Dirty.***

The 2nd cluster (5% coverage) has the ***highest average for NRS A and Demerit; and has the least number of problem sources, there are 26 problems in total with the with the most occurred problem is Bending***

The 3rd cluster (8% coverage) has the ***highest average number of lots inspected; has the highest number of claims from the 6 out of 8 quality indicators; and also has the highest number of problem sources, there are 80 problems in total with the most occurred problem is Short Mold***


Thank you!