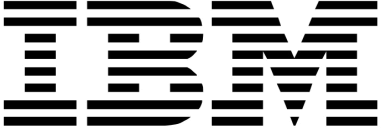
**Final Project Unsupervised Machine Learning:**

**IBM Machine Learning Professional Certificate**

**Course 04: Unsupervised Machine Learning |**

**Dry Bean Dataset**

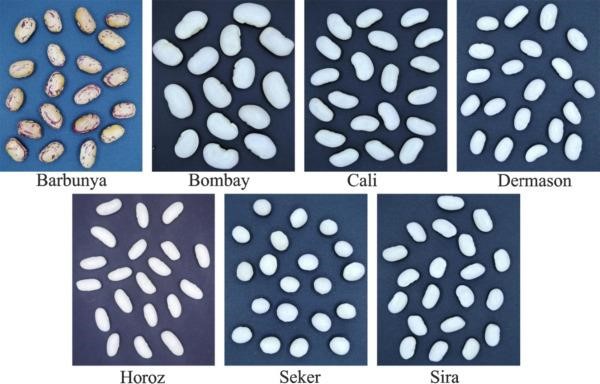
**By Gunuru.phanindra naidu**



**Dry Bean Dataset**

# Contents

* Dataset Description
* Main objectives of the analysis. • EDA, Data Cleaning, Feature Engineering
* Applying Clustering Algorithms.



* Machine learning analysis and findings.
* Models flaws and advanced steps.

Unsupervised Machine Learning

Data Description Section

Unsupervised Machine Learning

**Introduction**

ML

Algorithms

play

an

essential

and

promising

role

in

agricultural

sector,

from

this

point

will

discover

in

this

report

how

we

can

implement

unsupervised

learning

specifically

clustering

algorithms

to

a

dataset

has

Images

of

13

,

611

grains

of

7

different

registered

dry

beans

were

taken

with

a

high

-

resolution

camera

.

A

total

of

16

features

;

12

dimensions

and

4

shape

forms,

were

obtained

from

the

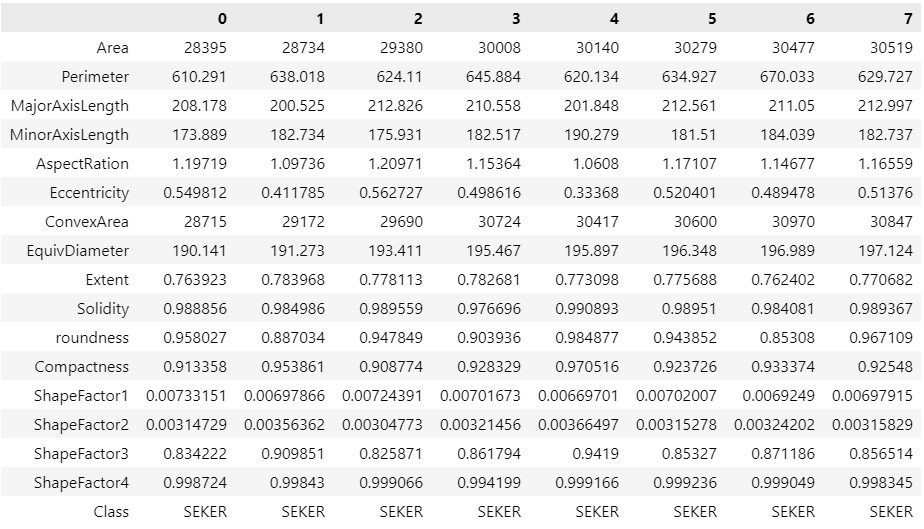
grains

.

Unsupervised Machine Learning

**Dataset Description**

**Features Explanations 01**



We have 16 features

in dry bean dataset in

addition of the

targeted variable

“Class” will explain

each of these features

in the next slides!

Dataset Dimensions:

❑

(

rows, columns) | (13611,

17)

# Dataset Description

**Features Explanations 02**

1. Area (A): The area of a bean zone and the number of pixels within its boundaries.
2. Perimeter (P): Bean circumference is defined as the length of its border.

6

.)

Eccentricity

(

Ec

):

Eccentricity of the ellipse

having the same moments as the region.

7

.)

Convex area (C):

Number of pixels in the

smallest convex polygon that can contain the

area of a bean seed.

3.)

Major axis length (L):

The distance between

the ends of the longest line that can be drawn

from a bean.

4.)

Minor axis length (l):

The longest line that

can be drawn from the bean while standing

perpendicular to the main axis.

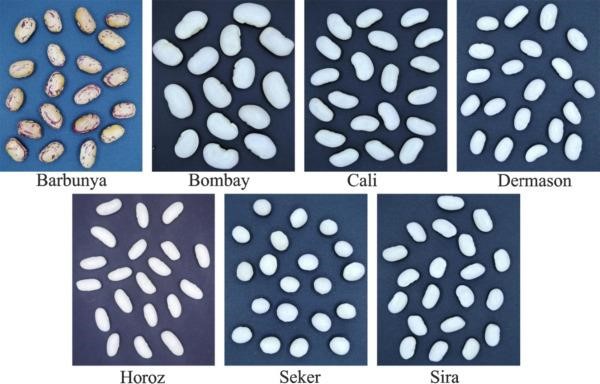
between L and l.

5.) Aspect ratio (K): Defines the relationship

1. Equivalent diameter (Ed): The diameter of a circle having the same area as a bean seed area.
2. Extent (Ex): The ratio of the pixels in the bounding box to the bean area.
3. Solidity (S): Also known as convexity. The ratio of the pixels in the convex shell to those found in beans.

# Dataset Description

**Features Explanations 03**



We have

7

different classes

of beans. So, when

we apply clustering methods, we expect to find

the same

number of Clusters!

using clustering

algorithms.

13

ShapeFactor

.)

1

SF

(

1

)

14

.)

ShapeFactor

2

(

SF

2

)

15

.)

ShapeFactor

3

SF

(

3

)

16

.)

ShapeFactor

4

(

SF

4

)

17

.)

Class:

(

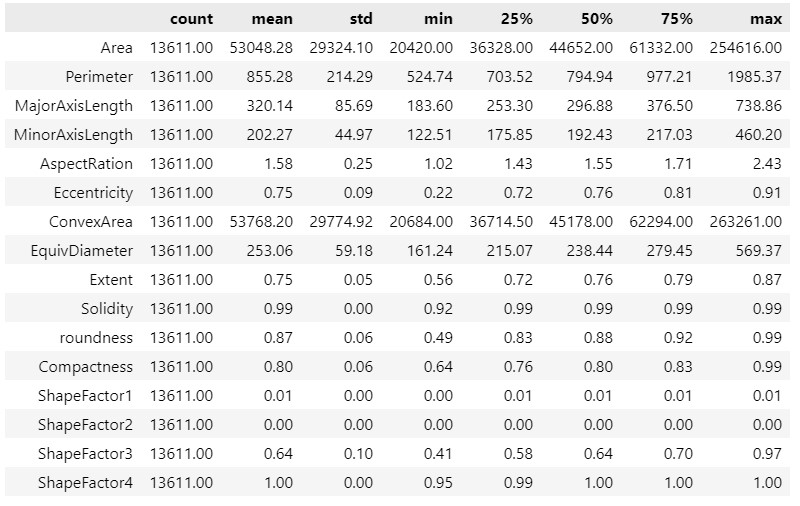
Barbunya, Bombay, Cali,

Dermosan, Horoz, Seker and Sira)

12.) Compactness (CO): Measures the roundness of an object: Ed/L

**Dataset Description**

**Features statistical description**



**Main Objective of the analysis:**

In this section we will explore the dataset in-depth through several EDA techniques such as checking null values, data skewness, and data visualization, furthermore, showing the correlation between the features for the sake of feature engineering implementation and data cleaning.

Unsupervised Machine

Learning

Exploratory Data Analysis (EDA) +

Feature Engineering Section

Great,

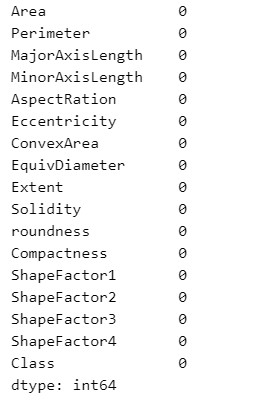
**no missing values**

within

our features !

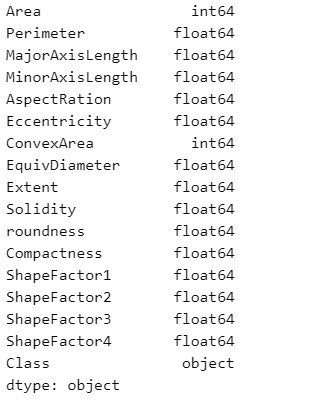
**Exploratory Data Analysis**

**Checking for Null values**



**-**

**Identifying categorical features and numerical features:**



**Exploratory Data Analysis**

Integers + Float

(

Numerical Features)

Categorical Features

# Exploratory Data Analysis

shown,

we

have

unbalanced

bean

dataset,

since

there

are

less

observations

for

some

classes

as

BARBUNYA

and

BOMBAY,

in

this

presentation

we

will

discover

if

this

could

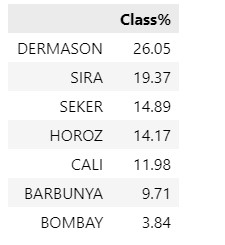
affect

the

clustering

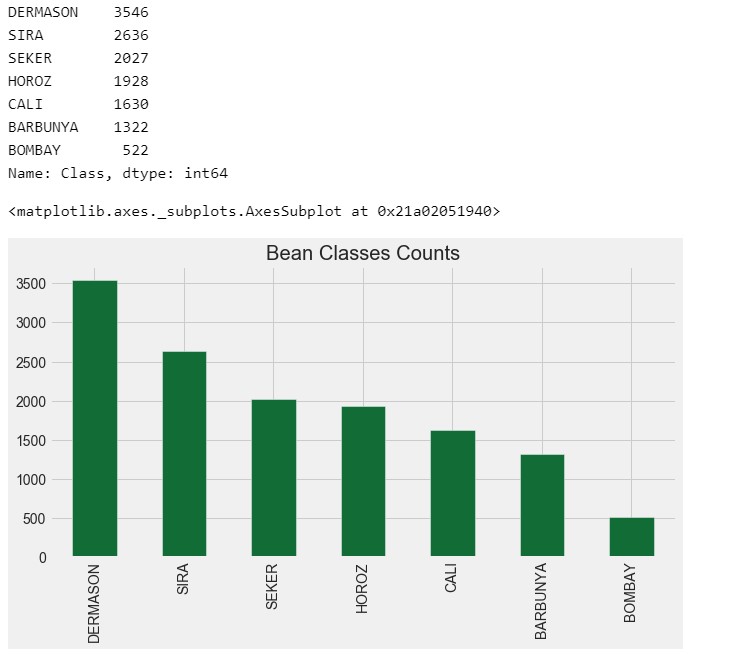
process

.



**Exploring the target variable :**

As



**-**

**Studying the correlations between features using Heat Map!**

The goal of this matrix is to show

the

**correlations**

between features,

and this is useful for feature

engineering techniques in the

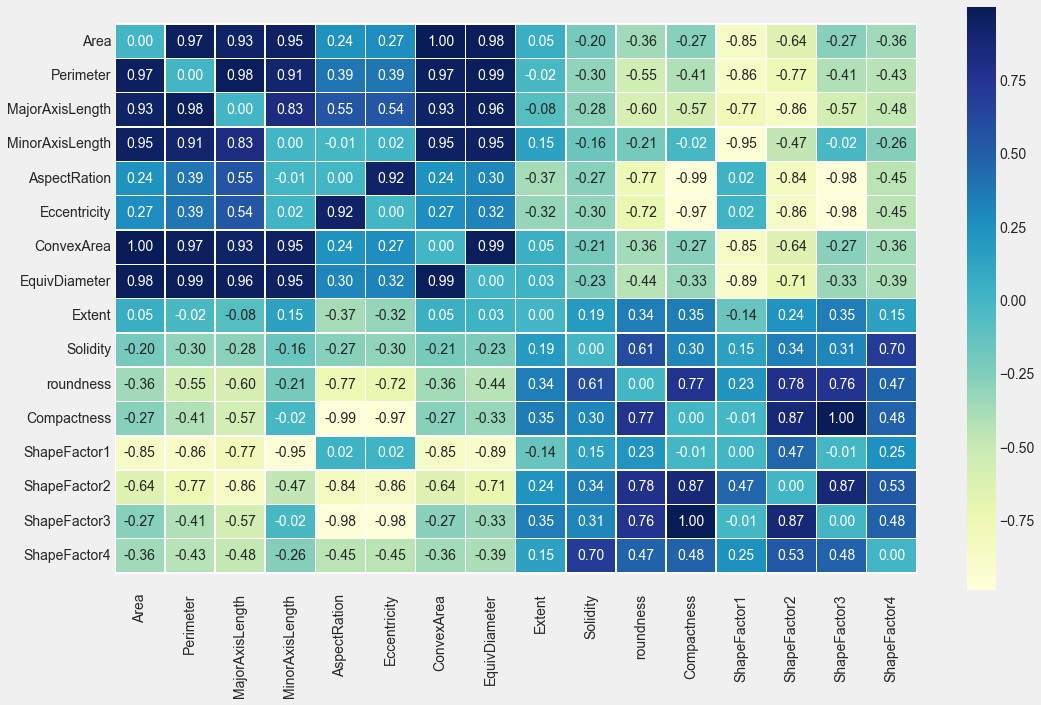
coming slides will show these

**correlations**

more

clearly through

data frames.



**Exploratory Data Analysis**

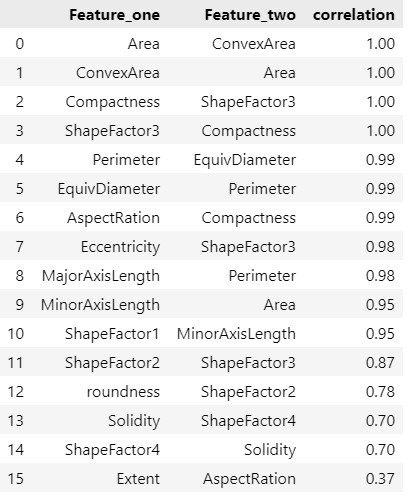
Unsupervised Machine

Learning

**-**

**Studying the highest correlations between the features**

**Exploratory Data Analysis**



We

can

notice

that

there

is

high

correlation

between

the

features

because

most

of

them

are

geometric

features

describe

the

shape

of

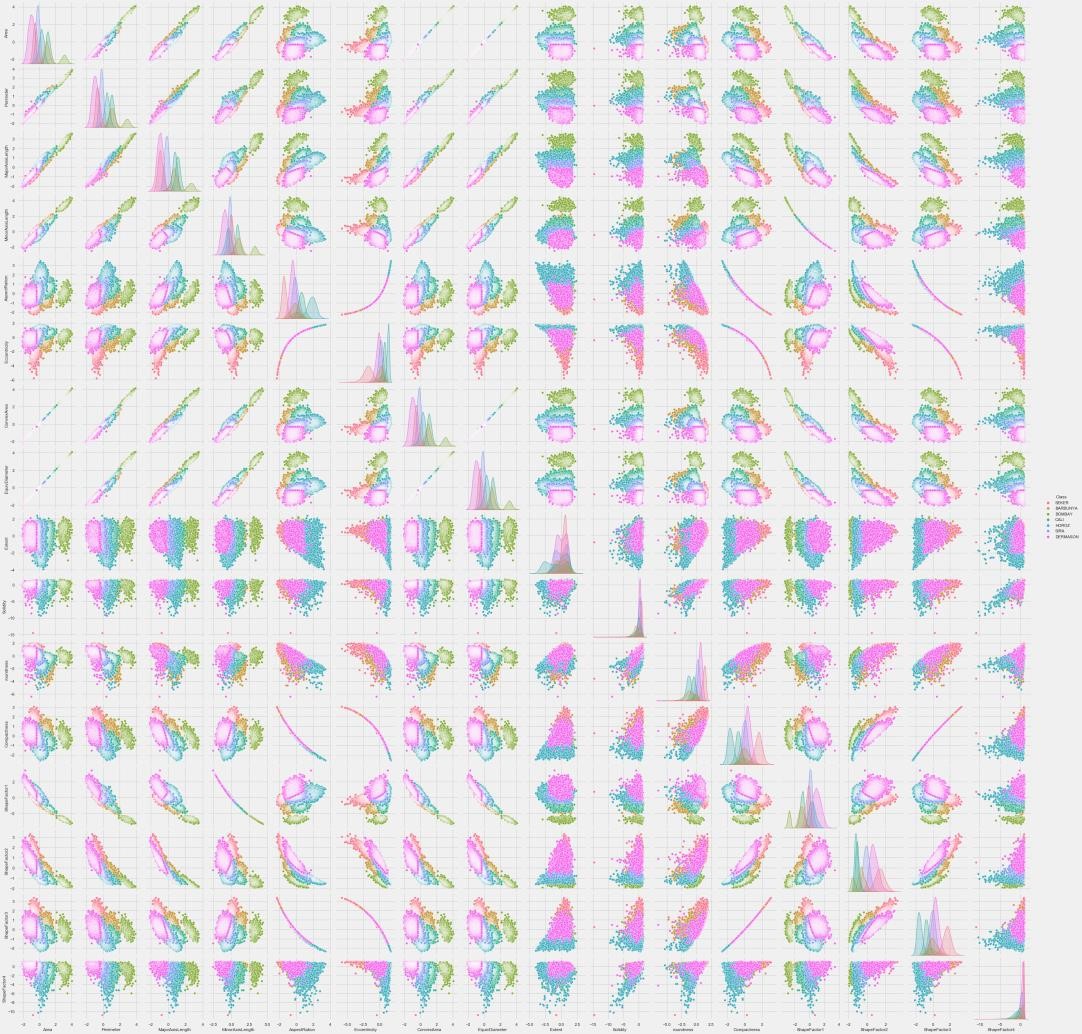
the

bean

.

# Exploratory Data Analysis

**Pair plot**



Unsupervised Machine

Learning

Using pair plot we can obtain the linearity between the features in addition of that we can see how beans classes are distributed to form the clusters.

**Skewness examination in the dataset**

**Exploratory Data Analysis**



Skewness

examination

in

the

features

in

anticipation

of

transformations

.

We

take

in

our

consideration

the

following

:

•

(

nearly

0

)

or

(

-

0

.

75

<

value

<

0

.

75

)

no

skewness

•

Positive

value

(

>

+

0

.

75

)

:

right

skewness

•

negative

(

value

<

-

0

.

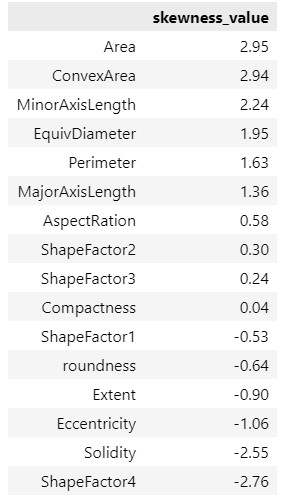
75

)

:

left

skewness



All features

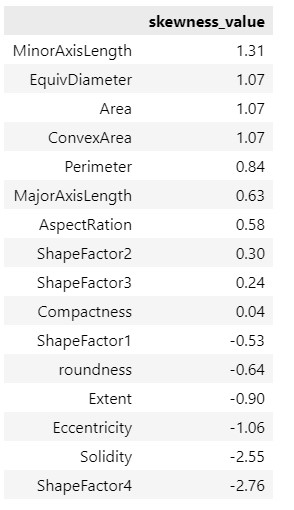
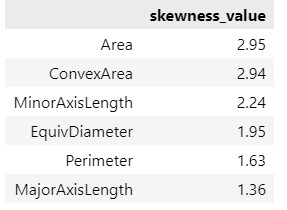
Feature should be transformed

Unsupervised Machine

Learning

**Applying Log transformation to right skewed data.**

**Feature Engineering**



Note

:

when

we

apply

log

transformation

on

the

features,

only

positive

skewness

will

be

handled

since

log

transformation

tries

to

approach

right

skewness

into

left

skewness

to

reach

out

the

symmetric

or

normal

distribution

.

Log

Transformation

**Features Scaling**

Feature scaling is relevant in machine learning models that compute some sort of distance metric, where we are

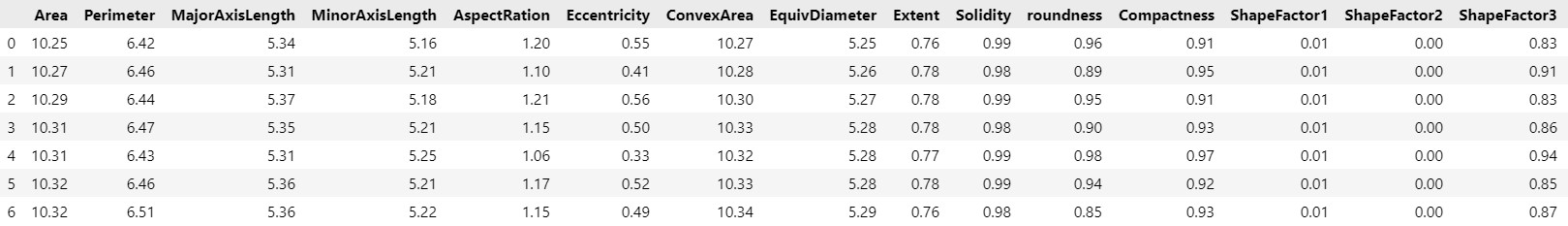
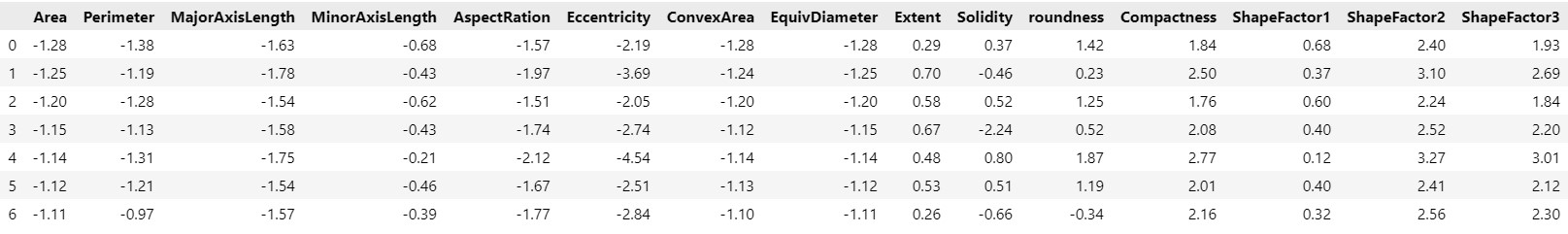
going to use clustering methods like K

-

means which depends on distance metric we must scale our features to

perform clustering in the appropriate way!

**Feature Engineering**



Machine Learning

Analysis & Findings

**Machine Learning Analysis & Findings**

In

the

following

slides

we

will

compare

between

4

different

Clustering

methods

k

-

means,

Agglomerative

Hierarchical

clustering,

DBSCAN,

MeanShift

in

terms

of

finding

the

appropriate

number

of

clusters

and

comparing

the

clustered

observations

with

the

actual

classes

in

the

dry

bean

dataset

.

**means Algorithm 01**

Here

we

have

Implemented

K

-

means

algorithm,

with

a

range

of

different

K

values

(

0

–

20

)

for

the

sake

of

finding

the

appropriate

number

of

clusters

in

dry

bean

dataset,

and

to

measure

the

entropy

in

the

model

we’ve

selected

the

inertia

metric

and

elbow

method

to

find

the

appropriate

value

of

K

no

(

.

clusters)

𝐈

𝐧

𝐞

𝐫

𝐭

𝐢

𝐚

:

is defined as the sum of squared distance

from each point (Xi) to its cluster Ck

෍

𝑖

=

1

𝑛

𝑋

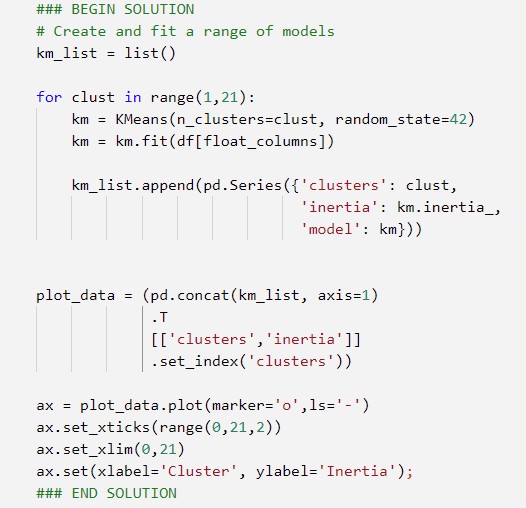
𝑖

−

𝐶

𝑘

2



**means Algorithm 02**

As shown in the graph, in case we follow elbow

method approach we can clearly obtain that the

appropriate number of clusters ranges between

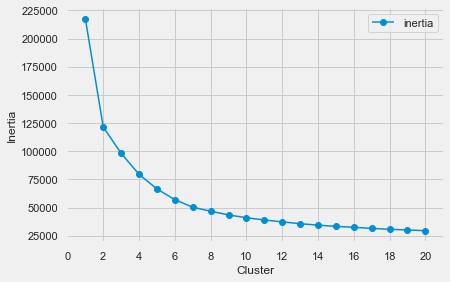
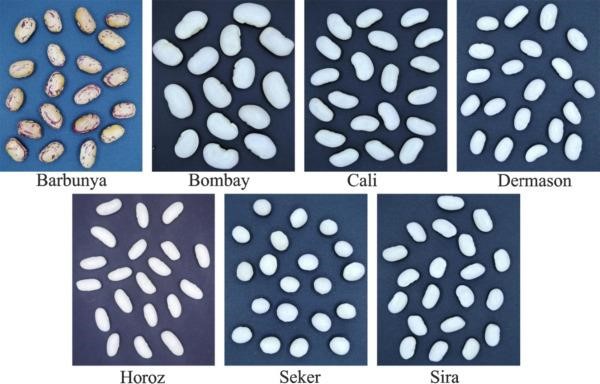
6

–

8

where, this is aligned with our dataset

which contains seven different clusters (Classes)



**means Algorithm 03**

After

applying

K

-

means

algorithm

with

no

.

clusters

=

7

to

the

dry

bean

dataset,

it

will

classify

each

observation

to

the

7

clusters

which

we

assigned

in

the

model

then

we

will

see

how

many

of

these

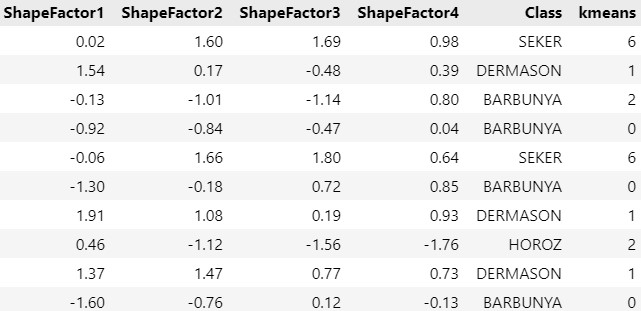
observations

are

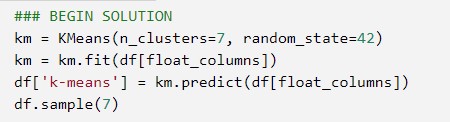
classified

correctly

.



Applying K-means with no. clusters = 7



|  |  |
| --- | --- |
| **Class Label** | **Cluster Number** |
| BARBUNYA | 0 |
| DERMASON | 1 |
| HORZO | 2 |
| Cali | 3 |
| BOMBAY | 4 |
| SIRA | 5 |
| SEKER | 6 |

**Machine Learning Analysis**

**K**

**-**

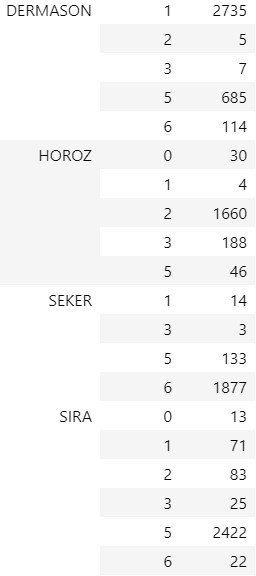
**means Algorithm 04**

Here we group by Class and k

-

means columns to validate our

clustering model with the actual classes labeling:



**Cluster**

**index**

**Class Label**

**Correct : all**

**Correct Clustering**

**percentage**

0

BARBUNYA

1176 : 1322

88.96

%

1

DERMASON

2735 : 3546

77.13

%

2

HORZO

1660 : 1928

86.10

%

3

CALI

268 : 1630

16.44

%

4

BOMBAY

520 : 522

99.17

%

5

SIRA

2422 : 2636

91.88

%

6

SEKER

1877 : 2027

92.60

%

The outcomes were very good and satisfactory for

5

clusters, , but only one of the clusters

(

CALI)

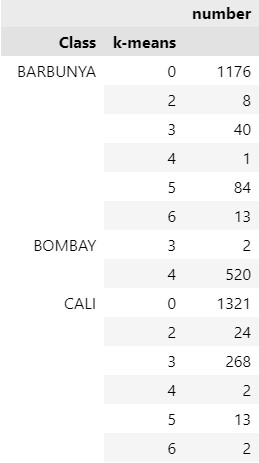
achieved poor results because of characteristics

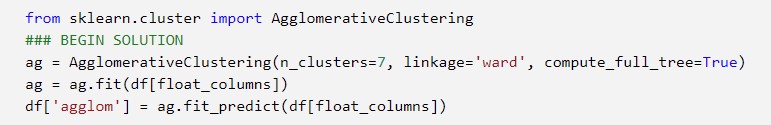
similarity with another cluster

BARBUNYA

)

(





**Machine Learning Analysis**

**Agglomerative Clustering Algorithm 01**

Unsupervised Machine

Learning

Applying agglomerative hierarchical clustering

with no. clusters = 7

**Class Label**

**Cluster Number**

BARBUNYA

0

HORZO

1

SIRA

2

SEKER

3

BOMBAY

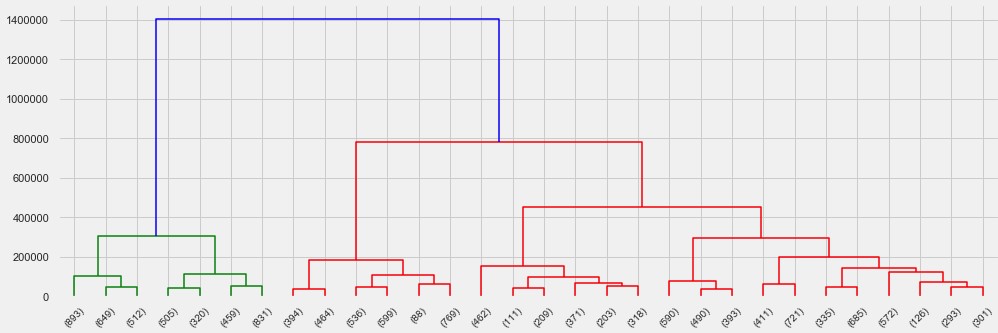
4

DERMASON

5

CALI

6



# Machine Learning Analysis

**Agglomerative Clustering Algorithm 02**

**Cluster**

**index**

**Class Label**

**Correct : all**

**Correct Clustering**

**percentage**

0

BARBUNYA

1273 : 1322

%

96.30

5

DERMASON

2969 : 3546

%

83.73

1

HORZO

1602 : 1928

%

83.09

6

CALI

6

: 1630

0.37

%

4

BOMBAY

522 : 522

100

%

2

SIRA

2252 : 2636

85.43

%

3

SEKER

1879 : 2027

92.70

%

Again

(

CALI)

Cluster

achieved very poor results

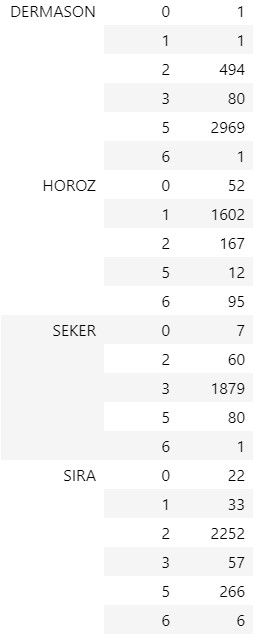
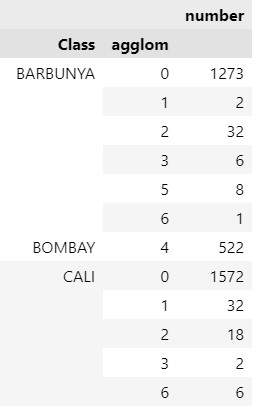
because of characteristics similarity

(

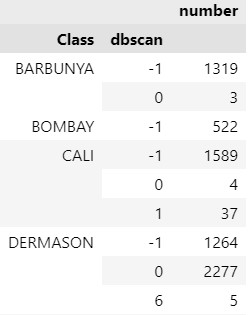
BARBUNYA)

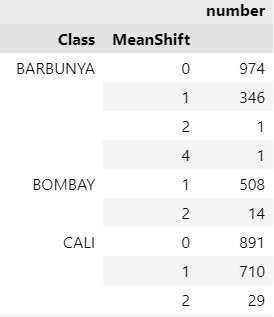
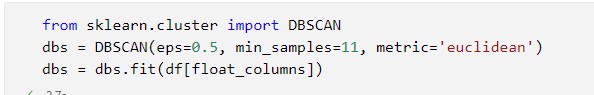
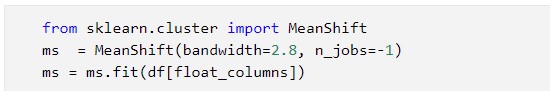
cluster

Here we group by Class and agglom columns to validate our clustering model with the actual classes labeling:



# Machine Learning Analysis

**DBSCAN and MeanShift Algorithms**



After applying these two algorithms many times with different parameters we got poor outcomes, where we validated the clustered observations with the actual classes, we got high error though the algorithm predicted the appropriate number of clusters

# Machine Learning Findings

**algorithms comparison**

Both clustering methods achieved great

Overall accuracy

but

at

the

end

will

choose

the best model one which is

K

-

means

with

overall accuracy = 92.05%.

As we mentioned in the previous slide DBSCAN and MeanShift achieved poor results in terms of observations clustering so will be excluded from the comparison and we will compare only between K-means and Agglomerative Clustering Algorithm

|  |  |  |
| --- | --- | --- |
| **Class Label** | **Correct Clustering**  **Percentage**  **K-means** | **Correct Clustering**  **Percentage**  **Agglomerative** |
| BARBUNYA | 88.96 % | 96.30 % |
| DERMASON | 77.13 % | 83.73 % |
| HORZO | 86.10 % | 83.09 % |
| CALI | 16.44 % | 0.37 % |
| BOMBAY | 99.17 % | 100 % |
| SIRA | 91.88 % | 85.43 % |
| SEKER | 92.60 % | 92.70 % |
| **Overall Accuracy** | **92.05 %** | **90.03 %** |

Models flaws and strengths

and advanced steps

# Models flaws and strengths

**Models Flaws and Strength:**

K-means and agglomerative hierarchical clustering methods both were fast and accurate in terms of finding the appropriate number of clusters in addition of that they clustered most majority of bean observations correctly except one class which was CALI due to its characteristic's similarity with BARBUNYA class but in overall both algorithms achieved above 90 % in terms of clustering accuracy.

In contrast, DBSCAN and MeanShift required a lot of time to find the appropriate number of clusters since they identify the number of clusters dependably on the parameters that we need to change again and again to find the expected number of clusters furthermore, the outcomes of the observations clustering were so poor.

# Advanced steps

**further suggestions:**

* To enhance the detection of CALI cluster in K-means and AHC models we can add another column represents the color of the bean to avoid the similarity issue with BARBUNYA



* To enhance the clustering process in DBSCAN and MeanShift we can use beam images as data instead of geomatic characteristics of bean because both algorithms play very good roles in computer vision applications or we can use grid search and hyperparameters tuning to find the best parameters, but this will require a lot of time especially with MeanShift model

# Thank you

**IBM Machine Learning Professional Certificate**

Unsupervised Machine Learning

By: Mohamad Osman