ÇUKUROVA UNIVERSITY

ENGINEERING AND ARCHITECTURE FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CEN438

GRADUATION THESIS

DATA ANALYSES BY USING OPTIMIZATION ALGORITHMS

BY

2017555071 – İrem YUVALI

ADVISOR

Dr. Havva Esin ÜNAL

JUNE, 2022

ADANA

ABSTRACT

Optimization is the problem of finding a set of inputs to an objective function that results in a maximum or minimum function evaluation. Firefly algorithm is one of the metaheuristic algorithms for optimization problems. In this study, we emphasize Firefly algorithm and compare our two data (Iris and Breast Cancer) with standard K-Means and analyze them according to the optimized version.

TABLE OF CONTENTS

PREFACE

The document contains the data analyses of Iris and Breast Cancer datasets according to the K-Means clustering extended by optimization algorithms named as especially Firefly and with some other algorithms. The mission of the project is to create and compare the analyses by using optimization algorithms for Graduation Thesis given by Havva Esin Ünal.

LIST OF FIGURES

DEFINITIONS

Firefly: optimization algorithm

WEKA: Data mining tool for data analyses.

SimpleKMeans: K-Means clustering feature on WEKA

Scikit-Learn library: Scikit-learn is a free machine learning library for Python..

ACO: Ant Colony Optimization.

FPA: Flower Pollination Algorithm

GWO: Grey Wolf Optimization

MFO: Moth Flame Optimization

PSO: Particle Swarm Optimization

WOA: Whale Optimization Algorithm

1. INTRODUCTION

The mission of the project is to create and compare the analyses by using optimization algorithms. For that aim, we define our datasets that are used in this study are explained. Then, we look at our optimization algorithms and start to analyze comparing K-Means and optimized versions.

2. DATASETS

In this section, the datasets that are used in this study are explained. These datasets are found in Scikit-Learn library as default. **[1]**

2.1 IRIS DATASET

The Iris dataset is one of the most famous and used datasets. It includes three iris species (named *"setosa"*, *"versicolor"* and *"virginica"*) with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

Figure 1: Setosa

Figure 2: Versicolor

Figure 3: Virginica

For importing the dataset from scikit-learn, it is used following commands:

```
from sklearn.datasets import load_iris
iris = load_iris()
iris.data
```

Also, it can be showed some details about the data:

```
iris.feature_names
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
'petal width (cm)']
iris_data.shape
(150, 4)

iris_data.describe()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Figure 4: Iris data describe

Moreover, the Iris data can be shown as a scatter plotting as follows.

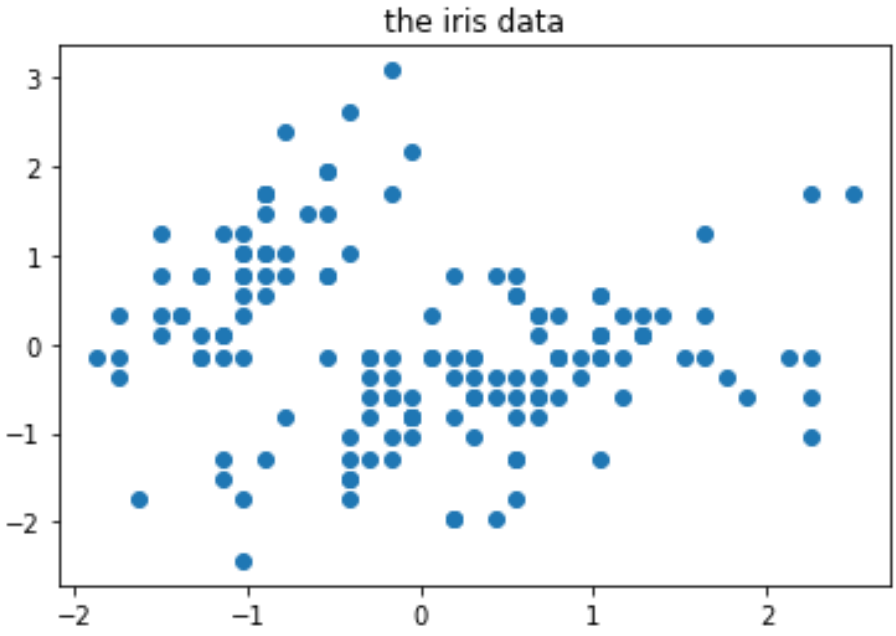

Figure 5: Iris data as a scatter plot

2.2 BREAST CANCER DATASET

Breast Cancer dataset, also known as Breast Cancer Wisconsin, shows breast mass features that computed from a digitized image of a fine needle aspirate (FNA). It contains 569 samples and 30 numeric features. Some of the features included in the dataset are; properties such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal size for each cell nucleus. There are two types of classes named *"benign"* and *"malignant"*.

For importing the dataset from scikit-learn, it is used following commands:

```
• from sklearn.datasets import load_breast_cancer
• cancer = load_breast_cancer()
• cancer.data
```

Also, it can be showed some details about the data:

```
• cancer.feature_names
• array(['mean radius', 'mean texture', 'mean perimeter',
  'mean area', 'mean smoothness', 'mean compactness', 'mean
  concavity', 'mean concave points', 'mean symmetry', 'mean
  fractal dimension', 'radius error', 'texture error',
  'perimeter error', 'area error', 'smoothness error',
  'compactness error', 'concavity error', 'concave points
  error', 'symmetry error', 'fractal dimension error', 'worst
  radius', 'worst texture', 'worst perimeter', 'worst area',
  'worst smoothness', 'worst compactness', 'worst concavity',
  'worst concave points', 'worst symmetry', 'worst fractal
  dimension'], dtype='<U23')
• #
• cancer_data.shape
• (569, 30)
```

Moreover, the Breast Cancer data can be shown as a scatter plotting as follows.

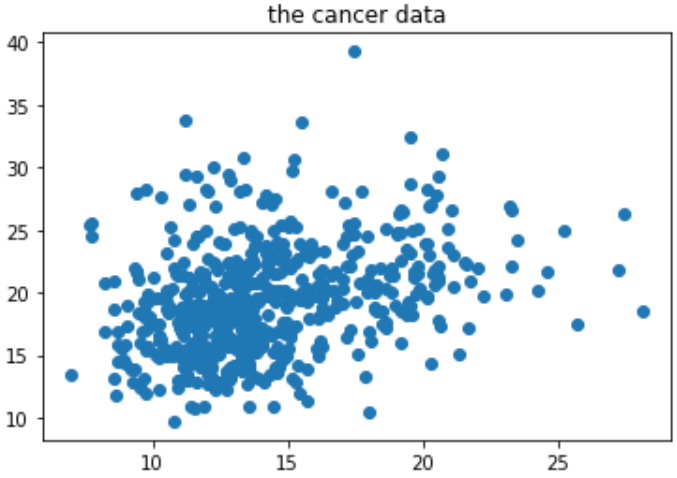

Figure 6: Breast Cancer data as a scatter plot

3. OPTIMIZATION ALGORITHMS

3.1 FIREFLY OPTIMIZATION

Firefly algorithm, being a Nature Inspired algorithm, is inspired by biochemical and social aspects of real fireflies. Real fireflies produce a short and rhythmic flash that helps them in attracting their mating partners and also serves as protective warning mechanism. Firefly Algorithm formulates this flashing behavior with the objective function of the problem to be optimized. (Yang, 2008) **[2]**

The swarm of n fireflies solve the problem iteratively and xi represents a solution for a firefly i at iteration t, whereas f(xi) denotes its fitness. The fitness of each firefly is determined by the landscape of the objective function. Furthermore, this fitness value determines the attractiveness of every i^{th} member in the swarm and represented as a light intensity Ii. Initially, all fireflies are dislocated in S (randomly or employing some deterministic strategy). Each firefly finds its matting partner based on attractiveness of other fireflies in the space and move towards that partner in order to improve its fitness. The next position of firefly i at iteration t + 1 is determined that considers two factors first, the attractiveness of other swarm members with higher light intensity, which is varying across distance and second a fixed random step vector ui where a Euclidean distance between two firefly i and j.

In general, describes the attractiveness when two fireflies are found at the same point of search space S. The value of $\gamma \in [0, 10]$ determines the variation of attractiveness with increasing distance from communicated firefly. It is basically the light absorption coefficient and generally $\gamma \in [0, 10]$ could be suggested (Yang, 2008).

The proposed algorithm FClust recasts the firefly algorithm scheme to improve the performance of clustering algorithm. Each firefly i is represented in k dimensions where each dimension represents the centroid of cluster and moves its position in order to achieve the objective functions stated. The algorithm will allow a firefly to move towards its mating partner only if there is an improvement in its intensity otherwise it will move with randomized steps μi in the space with probability p. The movement of a firefly i is attracted to another (brighter) firefly j is determined with β0 and $\gamma = 1$ as suggested by Yang.

- Pseudocode of Firefly algorithm is given in Figure 7 below.

Figure 7: Pseudocode of Firefly algorithm **[3]**

Algorithm 1. Pseudo-code of the base Firefly algorithm.

```
1:  t = 0; s* = ∅; γ = 1.0;        // initialize: gen.counter, best
    solution, attractiveness
2:  P(0) = InitializeFA();         // initialize a population
3:  while (t < MAX_FES) do
4:      α(t) = AlphaNew();         // determine a new value of α
5:      EvaluateFA(P(t), f(s));    // evaluate s according to f(s)
6:      OrderFA(P(t), f(s));       // sort s according to f(s)
7:      s* = FindTheBestFA(P(t), f(s));   // determine the best
    solution
8:      P(t+1) = MoveFA(P(t));     // vary the attractiveness
    accordingly
9:      t = t + 1;
10: end while
```

- The Firefly clustering algorithm structure is given in Figure 8 below.

Input:
Create randomly an initial population of n fireflies within k
dimensional search space x_{ik}, $i = 1, 2, \ldots\ldots N$ and $k = 1, 2, \ldots K$
Evaluate the fitness of the population $f(x_i)$ which is directly
proportional to light intensity I_i
Algorithm's parameters β_0 and γ, p
Output:
Obtained firefly i *with minimum* $f(x_i)$
while (t < maxiteration)
  for $i = 1$ to n **do**
    for $j = 1$ to n **do**
      if $f(x_j) < f(x_i)$ **then**
        find most attractive j for i using equation (7)
      end if
    end for j
    for $k = 1$ to K **do**
      find $x_{(t+1,k)}$ using equation (6)
    end for k
    Evaluate $x_{(t+1,k)}$ using equations (2) and (4)
    if $f(x_{t+1}) < f(x_i)$ then
      Update x_{it} with x_{t+1}
    else if random $\sim U(0, 1) >= p$
      $x_{t+1} = x_t + u_i$
    end if
  end for i
end while

Figure 8: Firefly clustering algorithm structure **[4]**

3.2 OTHER OPTIMIZATION ALGORITHMS

There is some information about other optimization algorithms that is used for comparison on the datasets.

- FPA is inspired from the pollination procedure of flowers. Several binary variants of FPA were developed to solve feature selection problem.

- WOA is based on the special behavior of the hunting method of humpback whales. The special hunting way of humpback whales is considered as the main interesting point of these whales. This method is called bubble-net feeding method.

- MFO: The special navigation method of moths in the night is the most interesting fact about them. They used the moonlight to fly in the night. They utilized transverse orientation mechanism for navigation. In this mechanism, a moth can travel long distances in a straight path through setting a fixed angle respect to the moon.

- GWO algorithm is one of the meta-heuristic algorithms. The main inspiration of GWO came from the hunting technique, and social leadership of grey wolves belonged to the Canidae family. Alpha is their leader. It is responsible for deciding on: sleeping place, hunting, etc. Beta is the second leader after alpha. It helps the alpha in decision making. The lowest ranking gray wolf is defined as omega. Its responsibility to submit the information to all the others dominant wolves. The rest of gray wolves are called delta. They dominate the omega.

- ACO: The inspiring source of ant colony optimization is the foraging behavior of real ant colonies. Though they live in colonies, they follow their own routine of tasks independent of each other. They perform many complex tasks necessary for their survival. They perform parallel search over several constructive threads based on local problem data and also have a dynamic memory structure which contains information on the quality of previously obtained results.

- PSO is a population-based stochastic optimization algorithm motivated by intelligent collective behavior of some animals such as flocks of birds or schools of fish. These swarms conform a cooperative way to find food, and each member in the swarms keeps changing the search pattern according to the learning experiences of its own and other members.

4. ANALYSES

4.1 STANDARD KMEANS CLUSTERING

Clustering is the task of dividing the population or data points into several groups such that data points in the same groups are more like other data points in the same group than those in other groups. It is the technique to classify objects or cases into relative groups called clusters. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

K-Means clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science, which groups the dataset into different clusters. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. Here K defines the number of pre-defined clusters that need to be created in the process.

4.1.1 KMEANS CLUSTERING WITH IRIS DATASET

By using WEKA tool, there are some results generated with “SimpleKMeans” on Iris dataset.

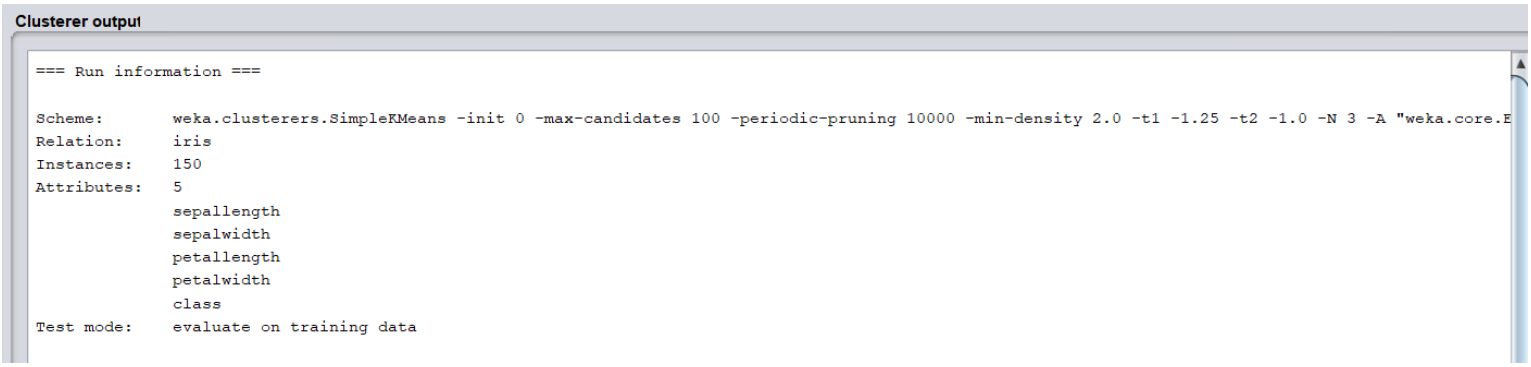

Figure 9: Iris Clustering Output-1

```
=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 7.817456892309574

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute               Full Data             0              1              2
                          (150.0)          (50.0)         (50.0)         (50.0)
================================================================================
sepallength                5.8433           5.936          5.006          6.588
sepalwidth                  3.054            2.77          3.418          2.974
petallength                3.7587            4.26          1.464          5.552
petalwidth                 1.1987           1.326          0.244          2.026
class                 Iris-setosa Iris-versicolor     Iris-setosa  Iris-virginica



Time taken to build model (full training data) : 0 seconds
```

Figure 10: Iris Clustering Output-2

```
=== Model and evaluation on training set ===

Clustered Instances

0       50 ( 33%)
1       50 ( 33%)
2       50 ( 33%)
```

Figure 11: Iris Clustering Output-3

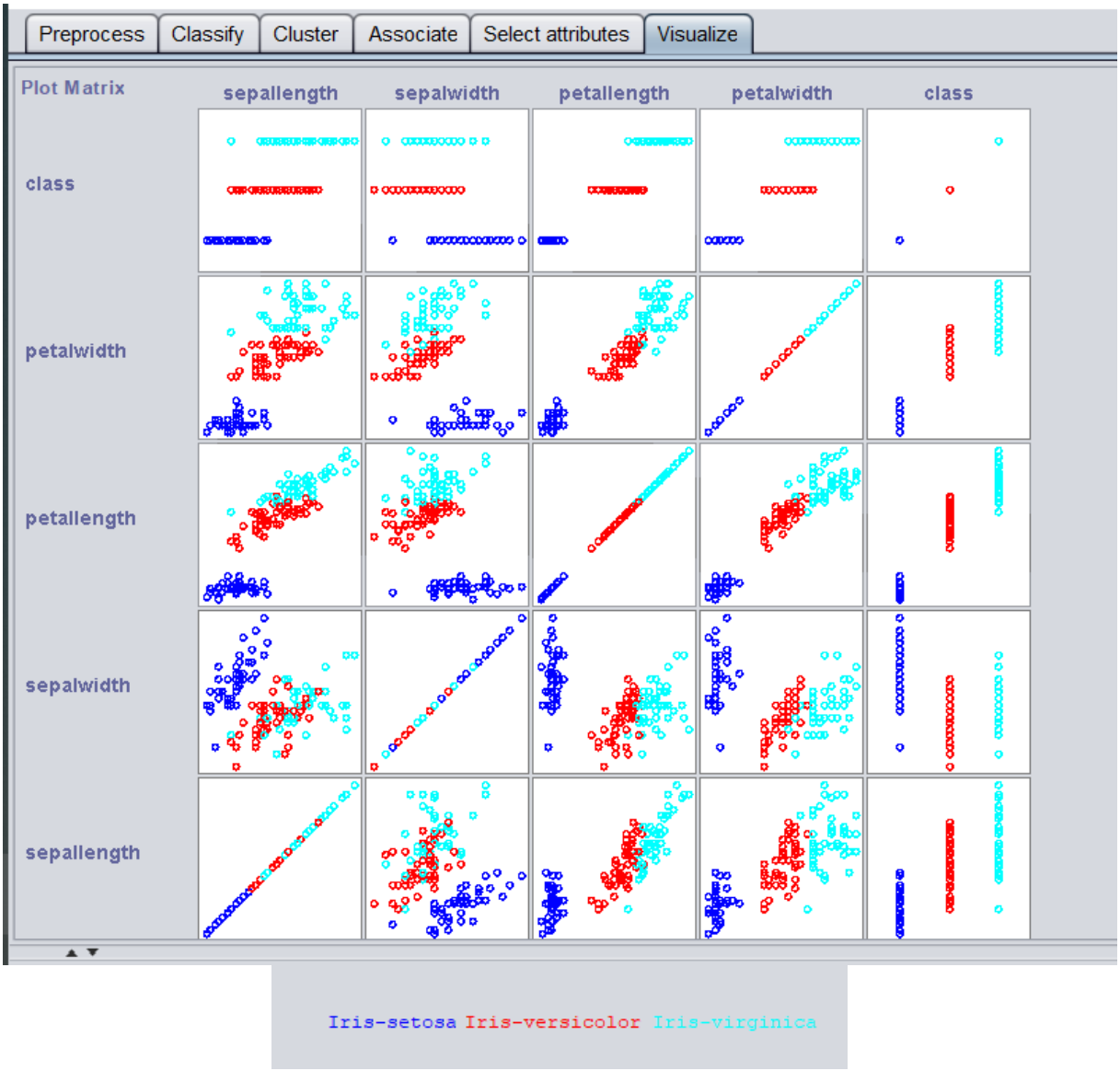

Figure 12: Iris Clustering Output-4

For importing the K-means features and apply on the dataset, it is used following commands:

```
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score

iris_ = pd.DataFrame(iris.data)
km_model = KMeans(n_clusters=3)
km = km_model.fit(iris_)

km.labels_
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2,
2, 1, 2, 1, 2, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2,
1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1], dtype=int32)
```

K-Means Clustered by k=3 version of the data can be shown as a scatter plotting as follows.

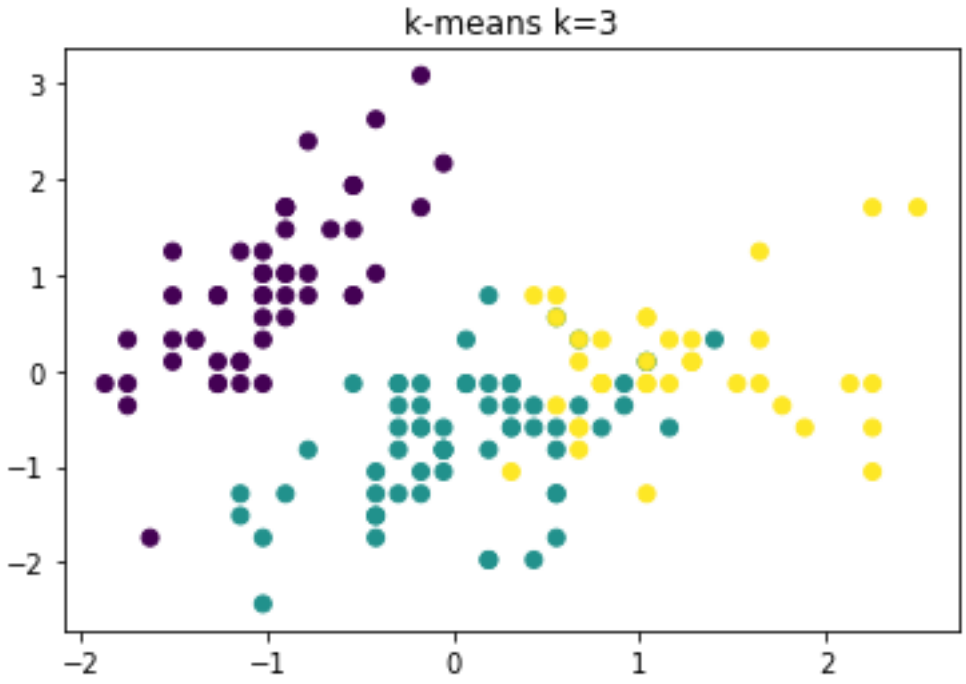

Figure 13: Iris K-Means Clustering by K=3 as a scatter plot

4.1.2 KMEANS CLUSTERING WITH BREAST CANCER DATASET

By using WEKA tool, there are some results generated with “SimpleKMeans” on Breast Cancer dataset.

Clusterer output

Attribute	Full Data (569.0)	Cluster# 0 (358.0)	1 (211.0)
diagnosis	B	B	M
radius_mean	14.1273	12.1454	17.4899
texture_mean	19.2896	17.9154	21.6213
perimeter_mean	91.969	78.0668	115.5567
area_mean	654.8891	462.7017	980.9701
smoothness_mean	0.0964	0.0925	0.1029
compactness_mean	0.1043	0.08	0.1456
concavity_mean	0.0888	0.046	0.1614
concave points_mean	0.0489	0.0257	0.0882
symmetry_mean	0.1812	0.1742	0.1931
fractal_dimension_mean	0.0628	0.0629	0.0627
radius_se	0.4052	0.2851	0.6089
texture_se	1.2169	1.2229	1.2067
perimeter_se	2.8661	2.0063	4.3248
area_se	40.3371	21.2133	72.7841
smoothness_se	0.007	0.0072	0.0068
compactness_se	0.0255	0.0214	0.0324
concavity_se	0.0319	0.026	0.042
concave points_se	0.0118	0.0099	0.0151
symmetry_se	0.0205	0.0206	0.0205
fractal_dimension_se	0.0038	0.0036	0.0041
radius_worst	16.2692	13.3797	21.1717
texture_worst	25.6772	23.5147	29.3463
perimeter_worst	107.2612	87.0006	141.637
area_worst	880.5831	558.8846	1426.4033
smoothness_worst	0.1324	0.1249	0.145
compactness_worst	0.2543	0.1824	0.3762
concavity_worst	0.2722	0.1659	0.4525
concave points_worst	0.1146	0.0744	0.1828
symmetry_worst	0.2901	0.27	0.3241
fractal_dimension_worst	0.0839	0.0794	0.0916

Figure 14: Breast Cancer Clustering Output-1

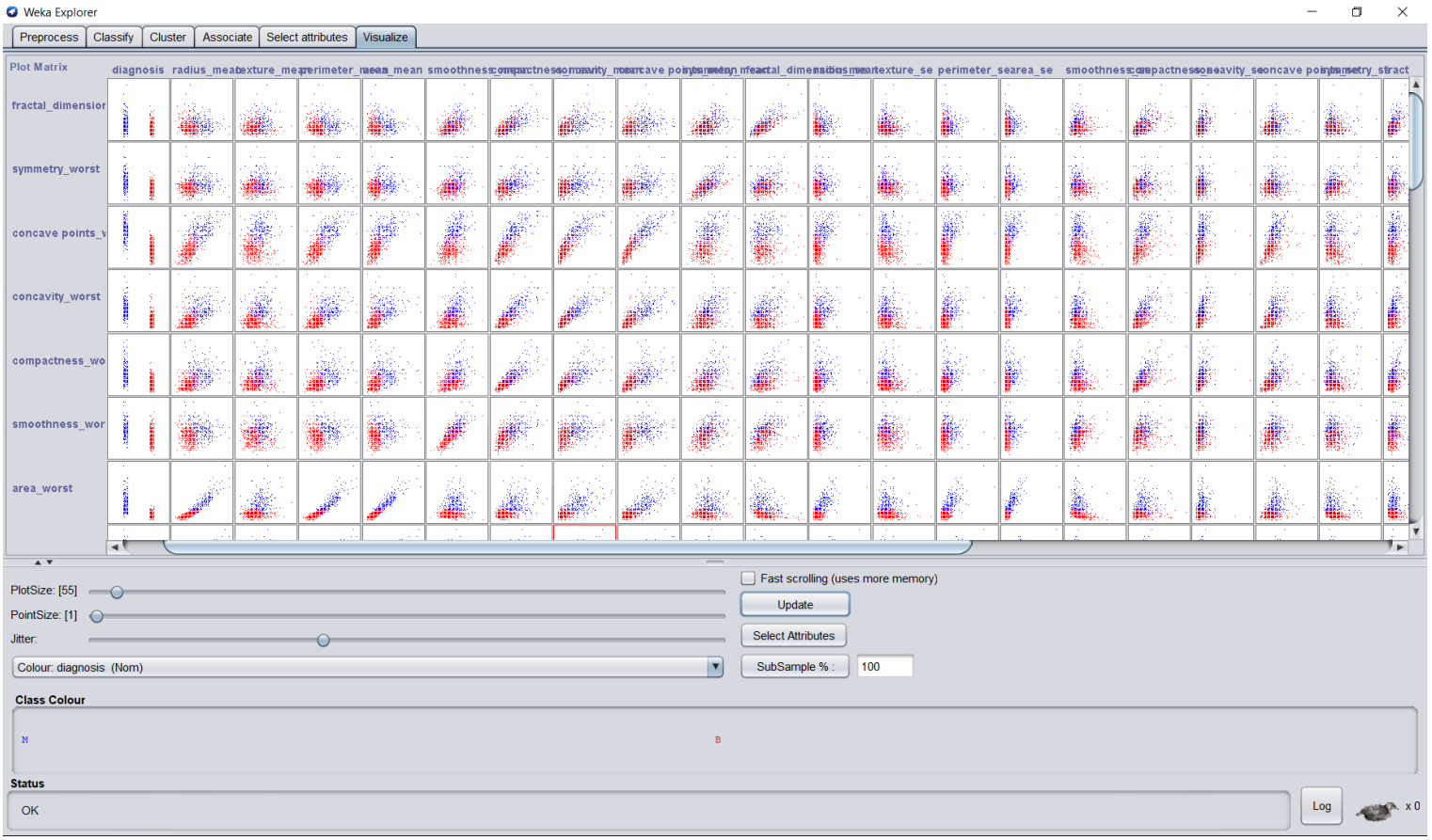

Figure 15: Breast Cancer Clustering Output-2

For importing the K-means features and apply on the dataset, it is used following commands:

```
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score

cancer_ = pd.DataFrame(cancer.data)
km_model = KMeans(n_clusters=2)
km = km_model.fit(cancer_)

km.labels_
```

```
array([0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0,
1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0,
1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0,
1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1,
1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1,
1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0,
1], dtype=int32)
```

K-Means Clustered by k=2 version of the data can be shown as a scatter plotting as follows.

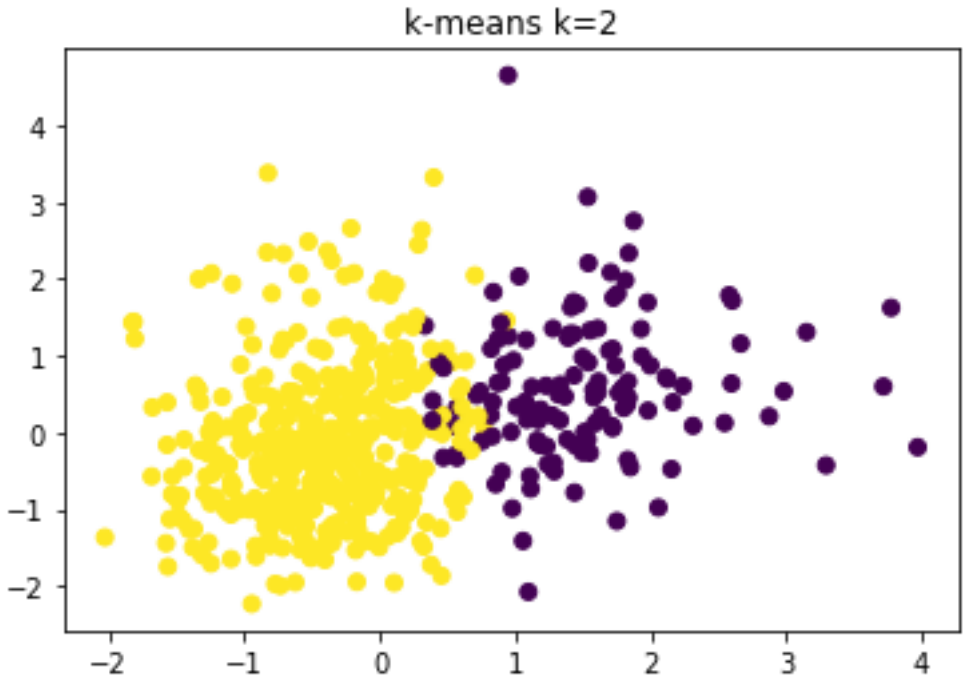

Figure 16: Breast Cancer K-Means Clustering by K=2 as a scatter plot

4.2 OPTIMIZED VERSION BY USING FIREFLY ALGORITHM

There is a code block that shows optimized version Iris dataset below.

```
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler

x = iris_data.drop('Species', axis=1)
y = iris_data.Species

scaler = StandardScaler().fit(x)
x = scaler.transform(x)

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x,y,test
_size=0.5, random_state=0)

IterationNumber, BestQuality, accuracy = FFA(x_train, x_test
, y_train, y_test, -50, 50, 4, 50, 10)
```

The output of the previous code is given below:

```
•
•  [0.87, 0.87, 0.89, 0.89, 0.92, 0.92, 0.93, 0.95, 0.98, 1.0]
•  At iteration is: 10 the best fitness is 11.0
•  accuracy: 1.0
•
```

There is a code block that shows optimized version Breast Cancer dataset below.

```
•  from sklearn import preprocessing
•  from sklearn.preprocessing import StandardScaler
•
•  x = cancer_data.drop('Species', axis=1)
•  y = cancer_data.Species
•
•  scaler = StandardScaler().fit(x)
•  x = scaler.transform(x)
•
•  from sklearn.model_selection import train_test_split
•
•  x_train, x_test, y_train, y_test = train_test_split(x,y, tes
   t_size=0.5, random_state=0)
•
•  IterationNumber, BestQuality, accuracy = FFA(x_train, x_test
   , y_train, y_test, -50, 50, 30, 50, 10)
```

The output of the previous code is given below:

```
•
•  [0.78, 0.8, 0.85, 0.91, 0.93, 0.94, 0.94, 0.95, 1.0, 1.0]
•  At iteration is: 10 the best fitness is 5.0
•  accuracy: 1.0
•
```

*The detailed codes are given in Appendix.

4.3 OPTIMIZED VERSIONS WITH OTHER ALGORITHMS

In this section, optimized versions of the datasets are compared with some other optimization algorithms that mentioned before. [in section 3.2]

Dataset	Parameters	Algorithms			
		K-means [49]	PSO [53]	ACO [52]	CSO [55]
Iris	Best	97.12	96.48	96.89	96.94
	Average	112.44	98.56	98.28	97.86
	Worst	122.46	99.67	99.34	98.58
	Std	15.326	0.467	0.426	0.392
	F-measure	0.781	0.78	0.778	0.781
Cancer	Best	2989.46	2978.68	2983.49	2985.16
	Average	3248.25	3116.64	3178.09	3124.15
	Worst	3566.94	3358.43	3292.41	3443.56
	Std	256.58	107.14	93.45	128.46
	F-measure	0.832	0.826	0.829	0.831

Figure 17: Comparison of Iris and Breast Cancer datasets [5]

Table 2: Highest obtained results of All swarms-based feature selection (FS) algorithms, where (NRA) is number of reduced attributes - (a) for WPBC and (b) for WBCD

Algorithm	**(a) NRA**	**(a) Accuracy (%)**	**(b) NRA**	**(b) Accuracy (%)**
WOA	8	84.34	10	98.77
FPA	12	80.81	11	96.66
MFO	15	81.82	14	95.32
GWO	13	81.31	16	96.65

Table 3: Stability of Bio-Inspired Features Selection Algorithms; (a) for WDBC and (b) for WPBC

	G	**(a) Mean**	**(a) Std.**	**(b) Mean**	**(b) Std.**
GWO	10	95.00687	0.848431672	78.55130333	1.25603624
	20	95.80955333	0.089101335	78.59649333	1.064270856
	30	95.74875133	0.264392974	79.22805667	0.119808738
	40	95.39284667	0.270428736	79.00001333	0.642459696
	50	95.57144667	0.236895453	79.29826667	0.248100199
MFO	10	91.00884667	0.60835743	74.056	0.368
	20	94.10652	0.02696	74.261	0.368
	30	93.50333333	0.1714	73.80133	0.00602
	40	92.9007	0.1869	74.4645	0.122529
	50	90.70333333	0.081989159	72.32267	0.044192
FPA	10	95.45236333	0.487356887	78.89124	1.015747167
	20	95.46366333	0.59817889	79.04710667	1.476681498
	30	96.20238667	0.353911543	80.87719333	0.129708218
	40	95.29866333	0.511592204	78.24561667	1.533457541
	50	96.21344	0.224282301	81.15789333	1.4148586
WOA	10	95.66669	0.813633011	80.78421667	1.844077133
	20	96.09897333	0.804263524	82.17545667	1.461934463
	30	96.23289667	0.524910606	82.30176	0.309381631
	40	96.24519667	0.326183197	80.47365667	1.484587255
	50	95.94824667	0.564285148	82.15791	1.361316191

Figure 18: Comparison of Breast Cancer dataset [6]

Results	Used Algorithm	K-Means	FCM	FCMPSO	FCMALO	FCMGWO	FCMSCA	FCMWOA	FCWOA1-c	FCWOA2-c	FCWOA3-c	FCWOA4-c	FCWOA5-c	FCWOA6-c	FCWOA7-c	FCWOA8-c	FCWOA9-c	FCWOA10-c
Iris Dataset	Benchmark Function (Mean)	91,6319	60,5057	60,5057	60,5057	61,4973	105,1161	63,0240	**37,2824**	**37,7335**	**37,7014**	**37,8305**	**37,3291**	**37,0239**	**37,4434**	**36,8744**	**37,3076**	**36,8381**
	RI (Max)	0,8797	0,8797	0,8797	0,8797	0,8797	**0,9412**	0,9195	0,9195	0,9267	0,9055	0,9195	0,9124	0,9124	0,9124	0,9124	0,9341	0,9195
	RI (Mean)	0,8481	0,8797	0,8797	0,8797	0,8785	0,8359	0,8788	**0,8923**	**0,8894**	**0,8860**	**0,8897**	**0,8894**	**0,8948**	**0,8893**	**0,8942**	**0,8936**	**0,8942**
	ARI (Max)	0,7302	0,7294	0,7294	0,7294	0,7294	**0,8672**	0,8180	0,8176	0,8343	0,7860	0,8176	0,8015	0,8022	0,8015	0,8019	0,8512	0,8176
	ARI (Mean)	0,6708	0,7294	0,7294	0,7294	0,7268	0,6476	0,7283	**0,7571**	**0,7523**	**0,7437**	**0,7529**	**0,7511**	**0,7624**	**0,7507**	**0,7609**	**0,7599**	**0,7611**
Breast Cancer Dataset	Benchmark Function (Mean)	19323,17	14916,68	14916,68	14916,68	14916,74	17796,01	14923,07	**5395,92**	**5620,80**	**5429,89**	**5421,45**	**5397,81**	**5344,28**	**5491,01**	**5490,57**	**5456,20**	**5469,72**
	RI (Max)	0,924	0,9159	0,9159	0,9159	0,9159	0,9403	0,9213	0,9132	0,9213	**0,9458**	0,9321	0,9321	0,9240	0,9294	0,9159	0,9294	0,9321
	RI (Mean)	**0,924**	0,9159	0,9159	0,9159	0,9159	0,9098	0,9166	0,8520	0,8411	0,8534	0,8177	0,8362	0,8690	0,8741	0,8426	0,8301	0,8729
	ARI (Max)	0,8465	0,83	0,8300	0,8300	0,8300	0,8797	0,8409	0,8247	0,8414	**0,8909**	0,8631	0,8631	0,8465	0,8577	0,8302	0,8575	0,8631
	ARI (Mean)	**0,8465**	0,83	0,8300	0,8300	0,8300	0,8175	0,8315	0,6970	0,6718	0,6984	0,6184	0,6604	0,7312	0,7414	0,6733	0,6515	0,7396

Figure 19: Comparison of Iris and Breast Cancer datasets with WOA algorithms [7]

5. COMPARISON

First of all, we got results with standard K-Means and here our accuracy values.

```
acc_km = accuracy_score(iris.target, km.labels_)
print(acc_km)

0.8933333333333333
```

```
acc_km = accuracy_score(cancer.target, km.labels_)
print(acc_km)

0.8541300527240774
```

- According to the results, we can say that K-Means method is more accurate for Iris dataset than Breast Cancer.

And then, we applied the Firefly algorithm to our datasets.

```
#Iris

[0.87, 0.87, 0.89, 0.89, 0.92, 0.92, 0.93, 0.95, 0.98, 1.0]
At iteration is: 10 the best fitness is 11.0
accuracy: 1.0

```

```
#Breast Cancer

[0.78, 0.8, 0.85, 0.91, 0.93, 0.94, 0.94, 0.95, 1.0, 1.0]
At iteration is: 10 the best fitness is 5.0
accuracy: 1.0

```

- According to the optimized versions by using Firefly algorithm, we can obtain the best quality states of the datasets.

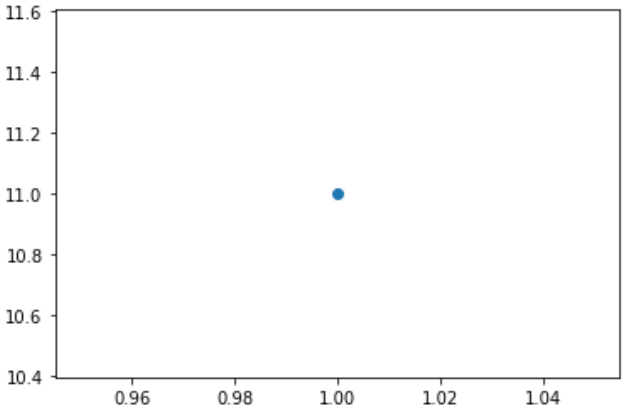

Figure 20: Best Quality for Iris

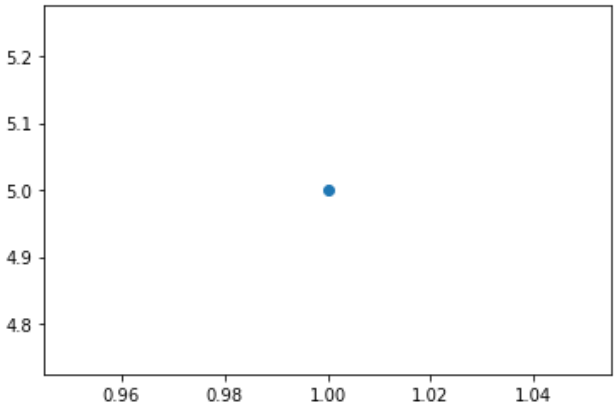

Figure 21: Best Quality for Breast Cancer

6. CONCLUSION

The results of the datasets have been analyzed and models are compared with each other. In this study Iris and Breast Cancer datasets were used in for observing the clustering results with by using optimization algorithms.

Although we get accurate results with standard K-Means, we improve our accuracies better. By means of Firefly algorithm, we can fit our accuracies at 1 after a number of iterations. Training iteration by iteration is an advantage for large dataset as it is seen in accuracy calculations in Breast Cancer dataset. It reached the 1 before than Iris.

Also, we can say K-Means and Firefly Algorithms are more fit for Iris dataset. We obtained more accurate results with it.

Consequently, we can improve our results with using optimization algorithms.

7. REFERENCES

[1]: https://scikit-learn.org/stable/datasets/toy_dataset.html

[2]: Yang X. S., Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[3]: http://www.iztok-jr-fister.eu/static/publications/23.pdf

[4]: https://www.researchgate.net/publication/264822886_Performance_analysis_of_firefly_algorithm_for_data_clustering

[5]: https://www.researchgate.net/publication/328366724_A_chaotic_teaching_learning_based_optimization_algorithm_for_clustering_problems

[6]: https://joems.journals.ekb.eg/article_23316_44fd5be7950dd8323a04daea867d2cc0.pdf

[7]: https://sigma.yildiz.edu.tr/storage/upload/pdfs/1635862524-en.pdf

ACKNOWLEDGEMENT

I would like to thank my supervisor Dr. Ünal. The work presented in this paper was supported by Dr. Havva Esin Ünal, who is esteemed acknowledged. I am thankful to her for their assistance at every stage of the project.

CV/RESUME

İrem Yuvalı was born in Istanbul in 1998. She completed her primary and secondary education at Ziya Gökalp Primary School. Then in 2016, she graduated from Bahçelievler Dede Korkut Anatolian High School with 81,55 GPA. She is currently a senior student at the department of Computer Engineering at Çukurova University, where she started in 2017.

Contact: 2017555071@ogr.cu.edu.tr

APPENDIX

Source Codes:

- https://colab.research.google.com/drive/1RGwbA5uz9PucA3fgqSEFe0tGNzENXVfx?usp=sharing
- https://colab.research.google.com/drive/1sC9xFCF5f2k6IyRM2CZEu9OFSICtPxii?usp=sharing