Information Technology Course
Module Software Engineering
by Damir Dobric / Andreas Pech

# *Implement Anomaly Detection with LearningApi (K-Mean Algorithm)*

Pretom Kumar Saha
saha@stud.fra-uas.de | Matrikal
No:1276545

***Abstract*— In the process of discovering patterns in large data sets, the identification of uncommon data is called anomaly detection. Data clustering is a method to make the groups of data depend on their behavior. In term of Data Clustering K-Mean Algorithm is the most popular. K-mean is basically used for clustering numeric data. Here, I implement k-mean algorithm through LearningApi to detect the anomaly from a data sate. From this Data cluster, Anomaly Detection is a process to find the unusual data which is different from other clustering data. K-mean is more reliable to create cluster and find Anomaly.**

***Keywords*— LearningApi, Anomaly Detection, Data Clustering, K-Mean Algorithm**

## I. INTRODUCTION

An approach in which can draw references from datasets consisting of input records without specified classification is called unsupervised learning method. Generally, to create a set of data in a meaningful structure, explanatory underlying processes, generative features and groupings, this unsupervised learning method is used.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

To identify the unusual data from a group of data that have different functionality is Anomaly detection technique. If it is possible to create cluster of data that have same functionality then it will be easy way to identify the Anomaly. K-mean is the most popular algorithm for creating cluster of data. K-mean is a simple algorithm for data clustering. K-mean algorithm create K number of data clustering with a centroid in every cluster. In every iteration, the algorithm creates a new set of data which have very shortest distance from the centroid [2].

## II. PROJECT OVERVIEW

Machine learning is a process which is used by computer with some algorithm and statistical model to perform some specific task by itself. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. To detect the anomaly from data set is one of them.

In anomaly detection, the task is to discover which parts of a collection are most odd with regard to the rest of the collection. For occasion, in case we had a collection of numeric information with height and width, we would need to identity anomalous, since it has different height and width with regard to the rest of the records within the collection. In this example we have no prior knowledge or training data.

There has lot of procedure to detect anomaly from data sets. Those method are also depending on the data type. For numeric data sets k-mean gives more accurate result. The focus point of my project is that integrate anomaly detection with LearningApi. The basic architecture is following:

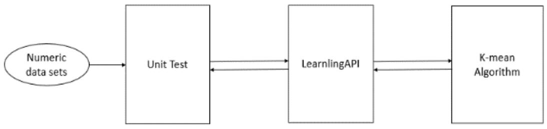


***Figure 1.*** *The basic architecture*

A numeric data sets will import through unit test. As a result, the data will use to train the LearningAPI which is using K-mean algorithm to make the data Cluster and detect the anomalies.

It provides a unique processing API for Machine Learning solutions. Here I used K-mean algorithm to make cluster of data and detect anomaly. Software testing with Unit test is a process, where individual components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. I used different unit test to check the accuracy of my project. [3]

## III. . METHODOLOGY

In the most common technique for clustering numeric data is called the k-means algorithm. In principle, at least, the k-means algorithm is quite simple. The central concept in the k-means algorithm is the centroid. In data clustering, the centroid of a set of data tuples is the one tuple that's most representative of the group. The idea is best explained by example.

Here, I consider double array type data has two dimensions length and width. If I consider length as x axis and width as y axis. We get data points Fig. 2.

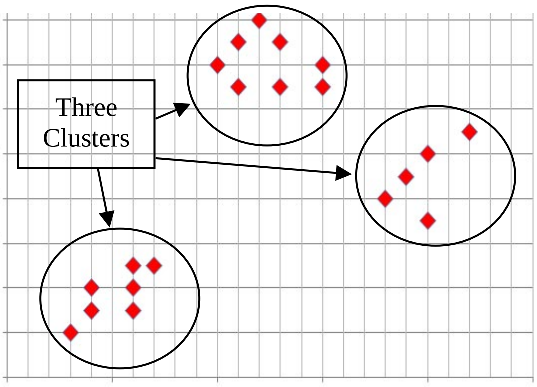


***Figure 2.*** *Data to Cluster [4]*

To use K-mean algorithm to find the final data cluster. The pseudo-code:

```
initialize clustering
loop until done
  compute mean of each cluster
  update clustering based on new means
end loop
```

In every iteration of k-mean well will get new cluster of data set. In final iteration we will get the actual cluster.

If, some new data with height and width value Fig. 3 is added which has same characteristics as the cluster, then the clusters will be same. But if there has some data that has different length and width then the clustering will be changed.

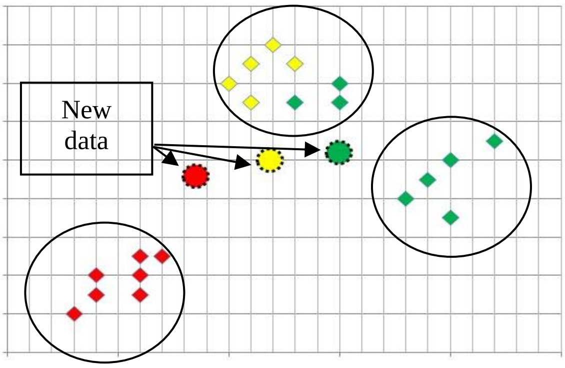


***Figure 3.*** *New Data added to Cluster [4]*

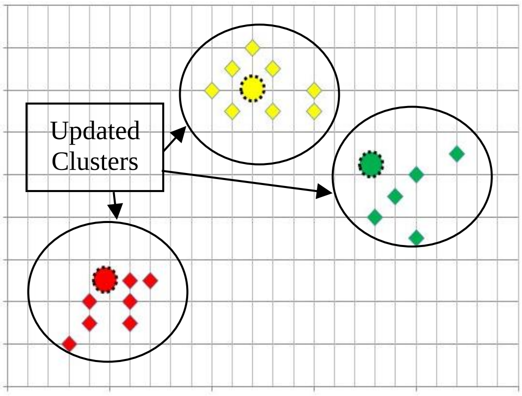


***Figure 4.*** *New Cluster with new Data [4]*

If the cluster is changed then there is a probability to have an anomaly. The data which change the cluster is the anomalous data. So, this data is the anomaly data in respect of previous dataset [4].

To find the anomaly data from the cluster, to calculate the distance d from the centroid to every element on that cluster. Then calculate the average change of distance in the cluster. If the average of distance change lass then 20% then there has no anomaly. If the average of distance change greater than 20% then there has an anomaly. The last data that makes the change of the cluster is the anomaly data.

$$d = |(c_o - c)|$$

$$d_s = \sum d$$

$$\alpha = d/n \quad (\alpha \geq 20) \text{ anomaly}$$

Here,

$d$ = Distance between centroid and other points

$c_o$= centroid of the cluster

$c$= other points on the cluster

$d_s$= total distance

$\alpha$= average of distance

## IV. IMPLEMENTATION

In this experiment below all numeric data contain height and width value like

| Height | Width |
|---|---|
| 5.1 | 3.5 |
| 4.9 | 3 |
| 4.7 | 3.2 |

First 80% of total data will be used as train data and the other 20% data will be used in prediction method. In test data contain anomalous. I insert anomalous like height: 11 and width: 1.9 segment on test data. The method returns a list of all segments ranked by how anomalous they are with respect to the whole data sets. If the program has performed well, then the truly anomalous segment should find out.

## A. *Structure*

The key object of Implementation is using K-mean Algorithm to create k number of cluster and detect the Anomaly of input data set inherits IAlgorithom interface from LearningApi. Moreover, I have used unit test for testing my methods.

After retrieving dataset from CSV file, it goes throw the training and prediction methods. And, at the unit test some of the cases are used to check the functionality of every methods.

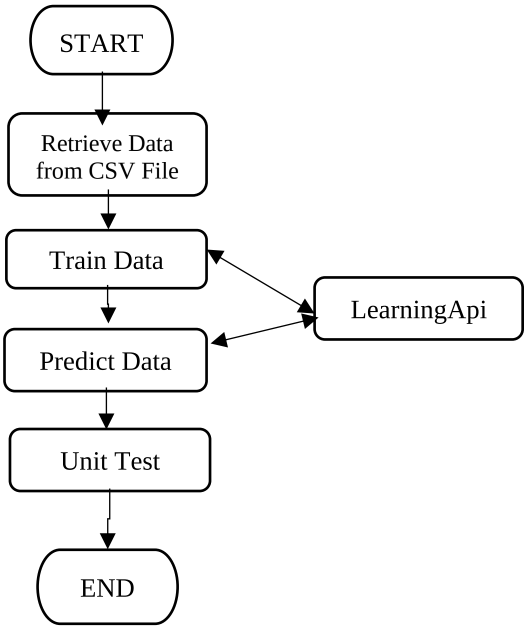


***Figure 5.*** Schematic Representation of the overall project.

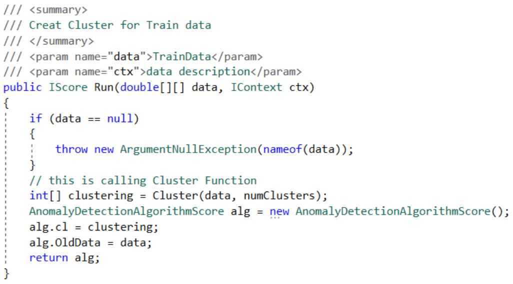
## B. *LearningApi*

The LearningApi interface is a set of three methods.

- Run() Method
- Train() Method
- Predict() Method

To implement the K-mean Algorithm, I have used Run() and Train() method to train input data. Predict() method is used to find the anomaly. The process is explaining here as following:

### a) *Train() and Run() Method*



```
/// <summary>
/// Creat Cluster for Train data
/// </summary>
/// <param name="data">TrainData</param>
/// <param name="ctx">data description</param>
public IScore Run(double[][] data, IContext ctx)
{
    if (data == null)
    {
        throw new ArgumentNullException(nameof(data));
    }
    // this is calling Cluster Function
    int[] clustering = Cluster(data, numClusters);
    AnomalyDetectionAlgorithmScore alg = new AnomalyDetectionAlgorithmScore();
    alg.cl = clustering;
    alg.OldData = data;
    return alg;
}
```

***Figure 6.*** *Run() method*

In the Run() method, K-mean Algorithm is implemented by a set of class.

b) Cluster()

c) Normalized()

d) InitClustering()

e) Allocate()

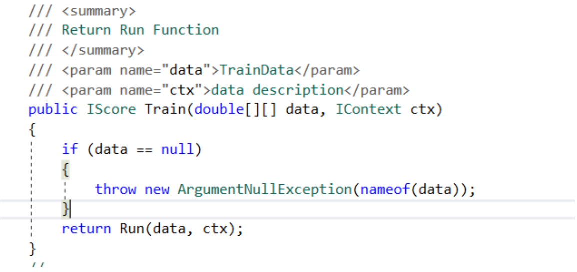
f) UpdateMeans()

g) UpdateClustering()

h) Distance()

i) MinIndex()

The Run() method takes two Parameters data and data description. The data is two dimensional array which is retrieved from CSV file. The CSV file contains Train data and Test data. Then the train data is passed throw the Run() method. The Run() method returns IScore type value which contain list of cluster with data set.

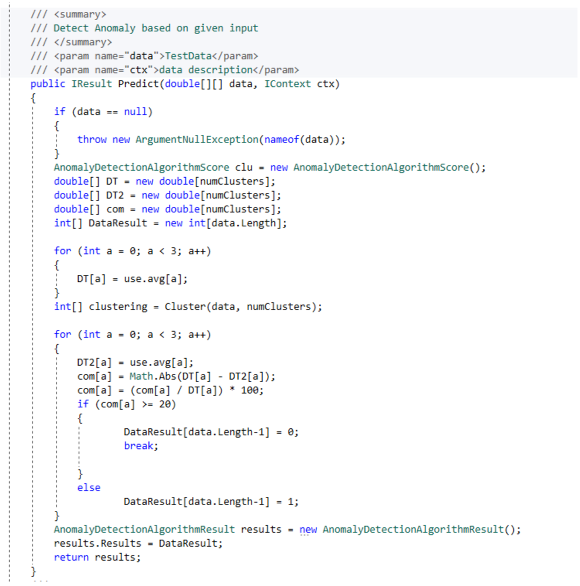


```
/// <summary>
/// Return Run Function
/// </summary>
/// <param name="data">TrainData</param>
/// <param name="ctx">data description</param>
public IScore Train(double[][] data, IContext ctx)
{
    if (data == null)
    {
        throw new ArgumentNullException(nameof(data));
    }
    return Run(data, ctx);
}
```

***Figure 7.*** *Train() method*

The train method is used to train data and call the Run() method with those data. When the Run() method returns the IScore Type value with clustering, the Train() method also return the same IScore type value.

### *b) Predict() method*



```
/// <summary>
/// Detect Anomaly based on given input
/// </summary>
/// <param name="data">TestData</param>
/// <param name="ctx">data description</param>
public IResult Predict(double[][] data, IContext ctx)
{
    if (data == null)
    {
        throw new ArgumentNullException(nameof(data));
    }
    AnomalyDetectionAlgorithmScore clu = new AnomalyDetectionAlgorithmScore();
    double[] DT = new double[numClusters];
    double[] DT2 = new double[numClusters];
    double[] com = new double[numClusters];
    int[] DataResult = new int[data.Length];

    for (int a = 0; a < 3; a++)
    {
        DT[a] = use.avg[a];
    }
    int[] clustering = Cluster(data, numClusters);

    for (int a = 0; a < 3; a++)
    {
        DT2[a] = use.avg[a];
        com[a] = Math.Abs(DT[a] - DT2[a]);
        com[a] = (com[a] / DT[a]) * 100;
        if (com[a] >= 20)
        {
                DataResult[data.Length-1] = 0;
                break;

        }
        else
                DataResult[data.Length-1] = 1;
    }
    AnomalyDetectionAlgorithmResult results = new AnomalyDetectionAlgorithmResult();
    results.Results = DataResult;
    return results;
}
```

***Figure 8.*** *Predict() method*

Predict() method contains a new data set. By using K-mean algorithm predict method creates a new clustering for the data. The cluster is compared with the previous cluster from IScore. If the clustering value is changed more than 80% then it has Anomaly. The Predict() method return IResult type value. The process is:

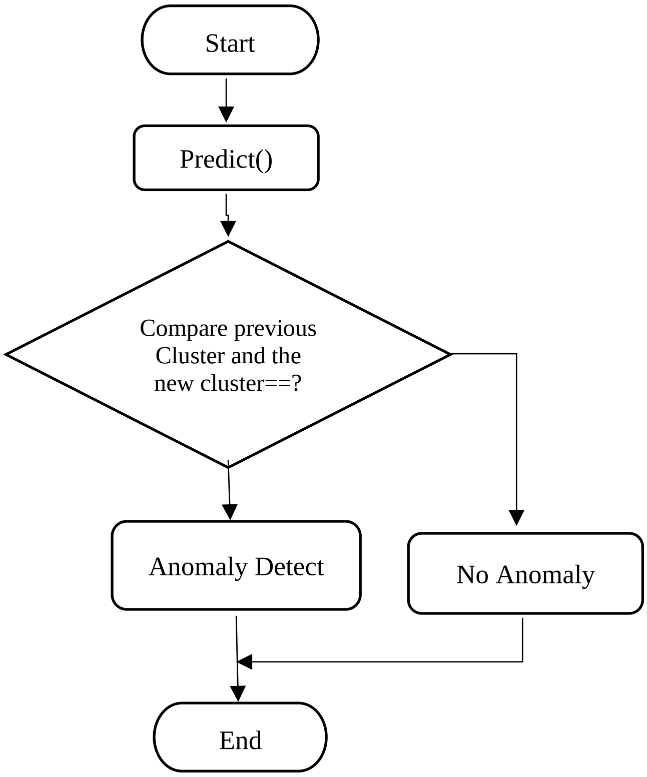


***Figure 9:*** *Schematic Representation of Predict().*

## V. Result and Testing

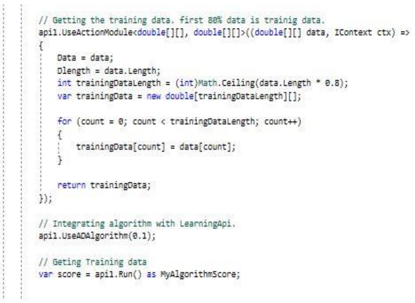
Data clustering is closely related to and sometimes confused with data classification. Clustering is an unsupervised technique that groups data items together without any foreknowledge of what those groups might be. Clustering is typically an exploratory process. The given data transfer into three cluster depend on there weighted values. Then this cluster value is used to detect anomaly from test data. I tested the cluster and anomaly with some know values like this:

| | height | width | Anomaly |
|---|---|---|---|
| 1 | 11 | 1.9 | 0 |
| 2 | 5.1 | 15 | 1 |
| 3 | 5.7 | 10 | 1 |

In this test value first column has anomaly. Because the train cluster have the value similar column 2 and 3. So these values don’t change the cluster. But the value of column has different value that change the cluster. So this data is the anomaly data.

The test process is done by some step.

- Step1: Creating an object of LerningApi
- Step2: Initializing the learningApi
- Step3: Retrieve data from CSV file.
- Step4: Separate the train data and Test data.



```
// Getting the training data. first 80% data is trainig data.
api1.UseActionModule<double[][], double[][]>((double[][] data, IContext ctx) =>
{
    Data = data;
    Dlength = data.Length;
    int trainingDataLength = (int)Math.Ceiling(data.Length * 0.8);
    var trainingData = new double[trainingDataLength][];

    for (count = 0; count < trainingDataLength; count++)
    {
        trainingData[count] = data[count];
    }

    return trainingData;
});

// Integrating algorithm with LearningApi.
api1.UseADAlgorithm(0.1);

// Geting Training data
var score = api1.Run() as MyAlgorithmScore;
```

***Figure 10:*** *Data Separation*

- Step5: Train the data set using Run()
- Step6: Use Test data to get predict value.

  var result = api1.Algorithm.Predict(testData, api1.Context) as MyAlgorithmResult;
- Step7: Test case that tests if the calculated predictions are correct.( Using last 3 data set with Anomaly)
- Step8: Test the Number of data length is equal.
- Step9: Test case that checks if ArgumentNullException is thrown in case of null argument for test data.
- Step10: Test case that tests if number Cluster value is calculated for each training data.

## VI. Conculasion

An approach in which can draw references from datasets consisting of input records without specified classification is called unsupervised learning method. Generally, to create a set of data in a meaningful structure, explanatory underlying processes, generative features and groupings, this unsupervised learning method is used.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

To identify the unusual data from a group of data that have different functionality is Anomaly detection technique. If it is possible to create cluster of data that have same functionality then it will be easy way to identify the Anomaly. K-mean is the most popular algorithm for creating cluster of data. K-mean is a simple algorithm for data clustering. K-mean algorithm create K number of data clustering with a centroid in every cluster. In every iteration, the algorithm creates a new set of data which have very shortest distance from the centroid [2]

## References

[1] https://github.com/UniversityOfAppliedSciencesFrankfurt/LearningApi

[2] James McCaffrey "Data Clustering - Detecting Abnormal Data Using k-Means Clustering"

[3] https://msdn.microsoft.com/en-us/magazine/jj891054

[4] https://visualstudiomagazine.com/articles/2013/12/01/k-means-data-clustering-using-c