Application of K-Means Clustering Algorithm for Prediction of Students' Academic Performance

CSCI 4144: Data Mining Project

Yucheng Liu, B00590305
Faculty of Computer Science, Dalhousie University
Yiying Zhang, B00560058
Faculty of Computer Science, Dalhousie University

April 10, 2015

Contents

Abstract

K-means classifier is a widely-known classification/machine learning algorithm. In this paper, a case where k-means algorithm was used for students' classification was studied. The traditional k-means algorithm was implemented in Python and was used to classify Iris dataset. The results of Radial Basis Function (RBF) algorithm and Multi-Layer Perceptron (MLP) algorithm on the same data set were compared to that of k-means algorithm. The result shows that RBF and MLP performed better on the Iris dataset than k-means. The overall accuracy for k-means was between 56%-58% most of the time, but k-means could sometimes reach an accuracy of almost 98%.

1 Introduction

In the education system, monitor students' academic performance is an important step for educators to adjust their teaching styles or changing their teaching plans to enhance the comprehension performance of their students. They need to make sure that all students understand the course material well and most of them feel comfortable with the amount and difficulty of homework, exams. GPA is the most common and powerful measurement to discover students' academic performance. [1] Many universities use GPA to estimate students' academic performance. In most university, the GPA value is in the range from 0 to 4.3. Higher GPA represents better understanding in the courses. In [1], authors chose to use k-means clustering algorithm to help the educator divide all students' GPA into k clusters. The basic idea of k-means is to divide the whole data set into several groups according to the distance between them. So the first step, users need to specify the number of clusters, denoted as 'k'. Then the program would create k centers. After that program would calculate Euclidean distance between the points and the centers. So that we would generate k clusters base on it.

1.1 Related Work

Mining student data, this project had been done by many studies. Most of them chose to mine student data using decision trees. In [2] and [3], both of them mining students' data base on decision tree. The only difference between them is that they are focusing on different ages. In the first paper, they chose to analyze the students' performance on the Mathematic and the Portuguese language. These two courses are the basis of their further studies. And in this paper they applied four DM models to test three selected dataset. In [3], the purpose of it is to help improving the quality of the higher education system so that they chose to run the project base on the university students' performance. The data mining tool they use is decision tree. Based on the result from decision tree, they try to predict the student's final grades. Although the results did not achieve a high accuracy, this project still shows that through a more efficient classification tool and a larger dataset would provide a more accurate result.

Another papers apply decision tree into a more interesting aspects. In [4], authors combine the decision tree with CHAID algorithm to analyze students' opinions to predict what satisfied student. In this paper the result is not absolute, and it cannot provide a numerical accuracy since the opinions of students are subjective. This paper provides some aspects that satisfy students mostly.

1.2 Application Scenario

It is not impossible for educators to go through all students' GPA one by one would cost too much time, therefore, it is necessary for educators to implement a useful tool to help them review students' GPA more efficiently. The k-means clustering algorithm divides all students into k clusters, then based on the results, educators can discover the key characteristic of different clusters of students which would help them improve their education quality.

1.3 Project Objective (Project Goal)

In this paper, a k-means clustering algorithm is implemented to monitor the progression of students' academic performance in higher institution. After clustering the entire students' record, the data about one specific cluster can tell educators more information about this group of students. This motivates educators' plans, which will improve the entire education quality.

2 Data Preparation

The first step of data preparation was to download the dataset folder from http://archive.ics.uci.edu/ml/. After download iris data set, the last column of this data set which represents types of Iris had been converted into 0, 1, and 2.

In order to reduce noise in the experiment, a shuffle method is added to rearrange the order of dataset. After rearranging the order of dataset, it can help program to avoid special circumstances.

3 Programing Architecture

As mentioned before, the first step is doing the data preparation. After doing all the converting jobs, a shuffle method added into the program. Since in order to apply this method to other dataset, not just constraint on one specific dataset. Shuffle method guarantees that the elements in dataset are not in a particular order. So that it can reduce noise from datasets. The following steps are highly according to the steps mentioned in the paper. The program contains two files:

Kmeans.py: Implements the k-means classifier

Iris_kmeans.py: The Demo program that contains the data preprocess

3.1 Algorithm

The first step of the k-means algorithm is to accept users' input, which tells program the number of clusters to group the dataset into, denotes as K. Then initialize the K clusters centroids, take random K instances from dataset. For each datapoint, the squared Euclidean distance of the point and each centroid is computed using the formula $d^2 = (x - x_c)^2 + (y - y_c)^2$, then a nearest centroid based on the squared Euclidean distance is assigned to the point. [1] After doing the above to all datapoints, for each centroid, a new position is assigned using the learning function: $\mu_i = \frac{1}{N_i}\Sigma_{j=1}^{N_i} x_j$. Repeat all the above steps until all the cluster centroids stop moving or it reaches the max number of iteration. The reason why we set a max number of iteration is that the clusters may be overfitted, or there is a possibility that the clusters never stop moving. Adding a limit on the number of iterations will be performed can avoid this problem.

Step 1: Accept the number of clusters to group data into and the dataset to cluster as input values

Step 2: Initialize the first K clusters
- Take first k instances or
- Take Random sampling of k elements

Step 3: Calculate the arithmetic means of each cluster formed in the dataset.

Step 4: K-means assigns each record in the dataset to only one of the initial clusters
- Each record is assigned to the nearest cluster using a measure of distance (e.g Euclidean distance).

Step 5: K-means re-assigns each record in the dataset to the most similar cluster and re-calculates the arithmetic mean of all the clusters in the dataset.

Figure 3.1: Generalized pseudocode of Traditional k-means [1]

3.2 Code Structure

The program is implemented in an object-oriented way. A k-means class is built in the file `kmeans.py`, the class's constructor accepts two parameters, an integer *k*, the number of clusters needed, and a preprocessed dataset matrix (multi-dimensional array). A function called `kmeanstrain()` is implemented to train the program so that it finds the suitable clusters for the dataset, and another function `kmeanstest()` is also implemented to classify the input test dataset.

Thanks to the nature of Python, there is no class structure in the demo program `iris_kmeans.py`. The first part contains the initial setups for reading a processing dataset. The k-means process is then set to run 1000 times using a for-loop.

3.3 Run Demo

To run this program, the user can open a console and use UNIX "`cd`" command to access into the directory containing the program files. After typing in the command "`python iris_kmeans.py`", the program starts.

Figure 3.3 gives a sample demo of this program.

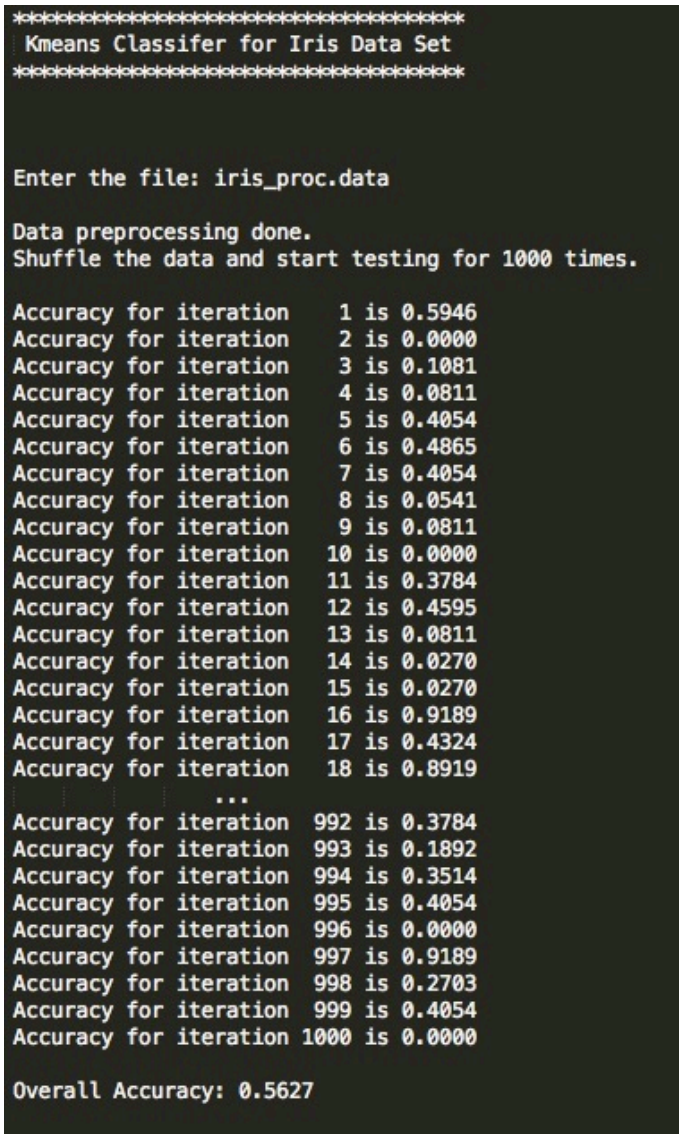

Figure 3.3

4 Evaluation

In this section, we give both the result of the paper and the result of our implementation on the iris dataset. And then we discuss the accuracy of k-means algorithm and other two algorithms.

4.1 Paper Result

This paper applied the model on the dataset of a university in Nigeria to analyze the academic result of one semester.

In table 4.1.1, the academic performance is divided into 6 parts. If the student's score is more than or equal to 70, the corresponding performance is "Excellent". Respectively, 60-69, 50-59, 45-49, and 40-45 correspond to the performance of "Very Good", "Good", "Very Fair", and "Fair". Finally, if the student had a score below 45, the performance is "Poor".

70 and above	Excellent
60-69	Very Good
50-59	Good
45-49	Very Fair
40-45	Fair
Below 45	Poor

Table 4.1.1: The score range and the corresponding performance [1]

For different number of clusters, the result generated by k-means algorithm is shown in different tables.

The paper gives three cluster values of 3, 4, and 5.

Cluster #	Cluster Size	Overall Performance
1	25	62.22
2	15	45.73
3	29	53.03

Table 4.1.2: The result table of k = 3 [1]

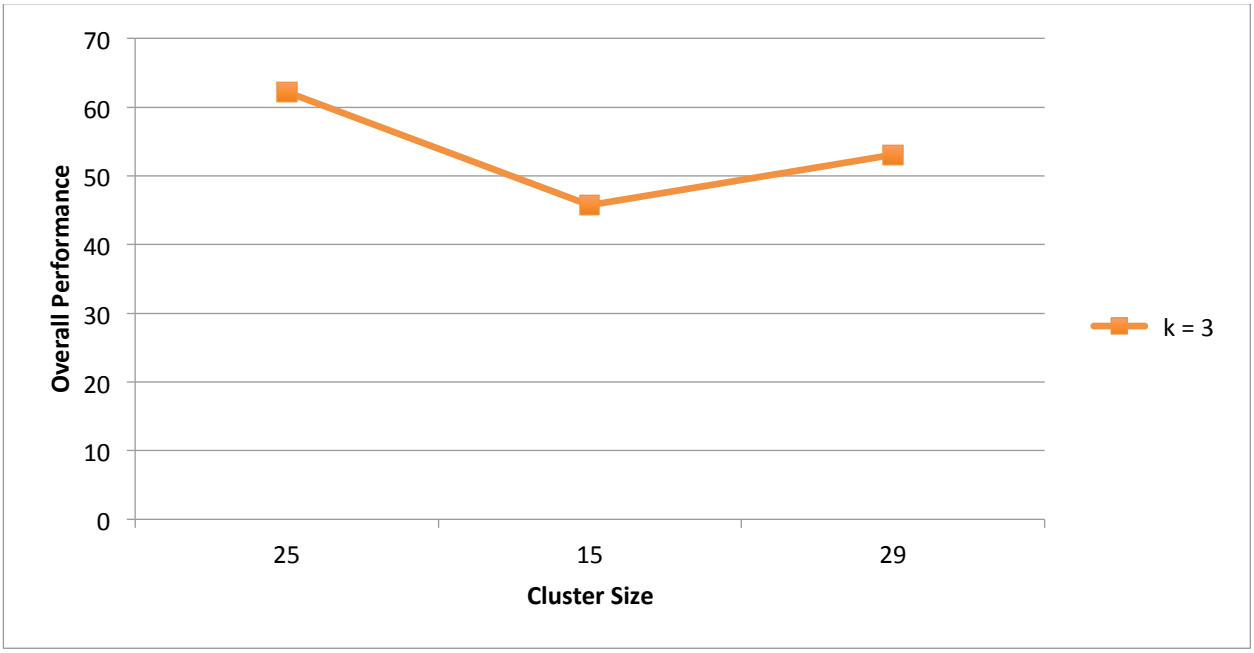

Figure 4.1.3: Line chart of overall performance versus cluster size when k = 3 [1]

Table 4.1.2 shows the result of k = 3. It indicates that for cluster numbers 1, 2, and 3, the cluster sizes (number of students) are 25, 15, and 29, and the overall performances are 62.22, 45.73, and 53.03.

Figure 4.1.3 is the line chart of Table 4.1.2, which illustrates the relationship of overall performance versus cluster size. It provides an intuitive description of the clustering results. According to Table 4.1.1, 25 students had a "Very Good" performance, while 15 students had a "Very Fair" performance. And for the remaining 29 students, they had performance in the region of "Good".

The other 2 clustering can be analyzed in the similar way.

Cluster #	Cluster Size	Overall Performance
1	24	50.08
2	16	65.00
3	30	58.89
4	9	43.65

Table 4.1.4: The result table of k = 4 [1]

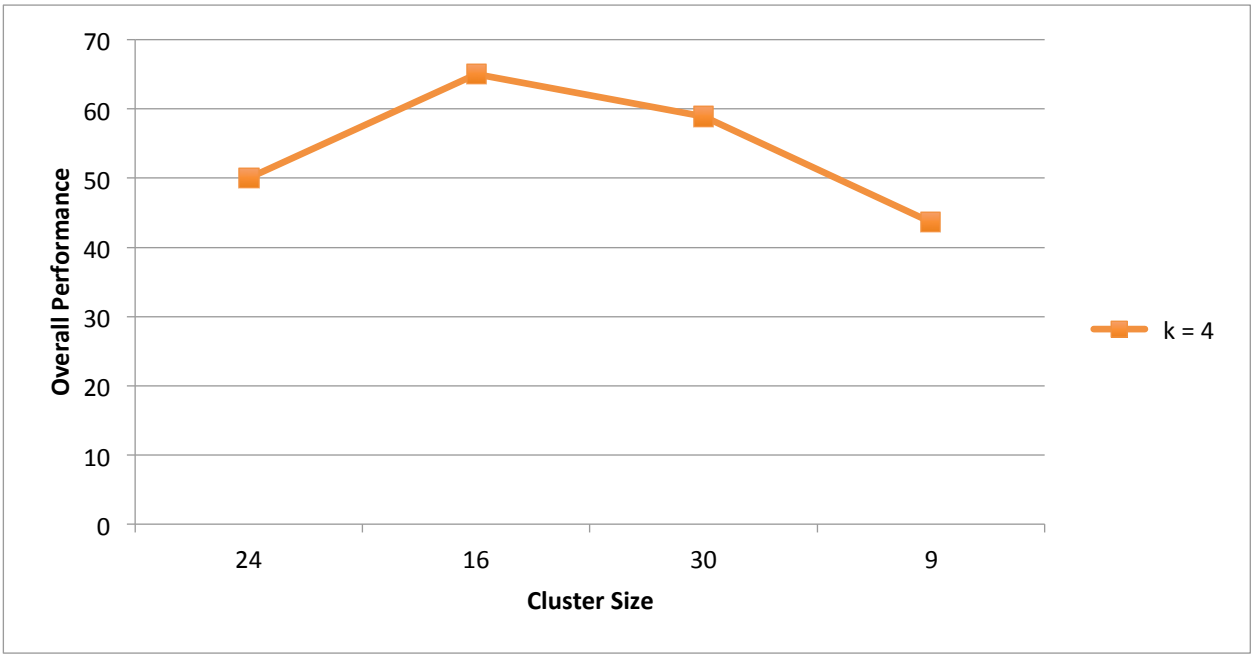

Figure 4.1.5: Line chart of overall performance versus cluster size when k = 4 [1]

Table 4.1.4 and Figure 4.1.5 are the result table and line chart for k = 4. The analysis showed that, the 1st cluster with 24 students had a "Good" performance, the 2nd cluster with 16 students had a "Very Good" performance, he 3rd cluster with 30 students also had a "Good" performance, and the last cluster with 9 students fell in the region of "Fair" performance.

For k = 5, clustering results are shown in Table 4.1.6 and Figure 4.1.7. The trends in this analysis indicated that, both the 1st cluster with 19 students and the 5th cluster with 20 students had "Good" performance results, while another 2 cluster, the 2nd one and the 4th one, both crossed over to "Very Good" performance region, and the remaining 9 students in the 3rd cluster fell in the region of "Fair" performance.

Cluster #	Cluster Size	Overall Performance
1	19	49.85
2	17	60.97
3	9	43.65
4	14	64.93
5	20	55.79

Table 4.1.6: The result table of k = 5 [1]

Figure 4.1.7: Line chart of overall performance versus cluster size when k = 5 [1]

4.2 Performance Result

We used the iris dataset downloaded from the UC Irvine Machine Learning Repository to test our k-means algorithm.

After running the code `iris_kmeans.py`, we obtain the result by clustering the iris dataset for 1000 times. The result shows us the accuracy of the 1000 tests. Figure 4.2.1 is the screenshot of one part of the `kmeans_result.txt` file.

In these 1000 tests, the lowest accuracy is 0.0, while the highest accuracy is 0.972972972972973.

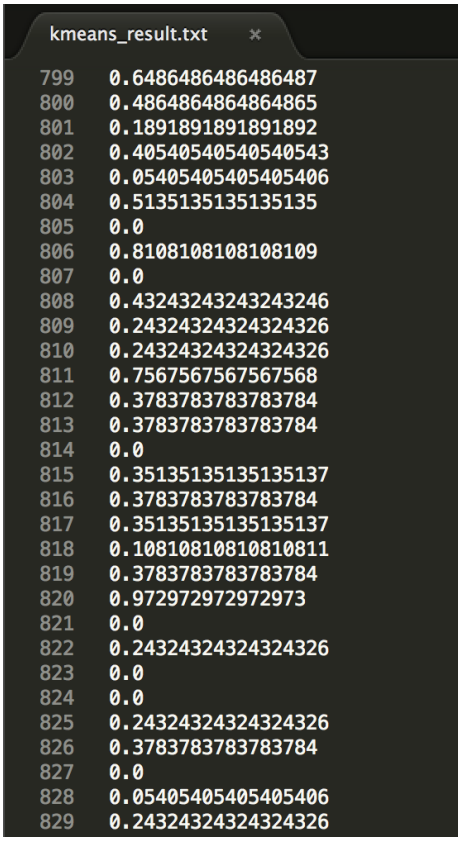

Figure 4.2.1

4.3 Discussions

The k-means algorithm is one kind of unsupervised learning, which belongs to the problem of trying to find hidden structure in unlabeled data. To further analyze our k-means result, we implement another two algorithms, RBF and MLP, to deal with the same iris dataset. RBF denotes for Radial Basis Function, and MLP denotes for Multi-Layer Perceptron. They both belong to supervised learning, which is a machine learning method to deduce from labeled training data.

We use RBF and MLP to do the same thing as that of k-means – handling the dataset for 1000 times each. Figure 4.3.1 is the screenshot of one part of the `rbf_result.txt` file, and Figure 4.3.2 is the screenshot of one part of the `mlp_result.txt` file.

Then we calculate the mean value of each result. The mean value of k-means accuracy is 57.53%. We get rid of the results that are smaller than 30%. The mean value of RBF accuracy is 94.92%, and the mean value of MLP accuracy is 90.40%. Figure 4.3.3 gives the line chart.

Comparing the three results, we can easily find that both RBF and MLP have the accuracy much higher than k-means. Among the 1000 RBF accuracy results, there are only 3 accuracy results below 0.7. And for MLP, there are only 13 accuracy results below 80%. Especially, RBF has 31 accuracy results of 100%, and MLP has 164 of that.

rbf_result.txt

```
259  0.89189189189189189
260  0.94594594594594594
261  0.94594594594594594
262  0.97297297297297303
263  0.91891891891891897
264  0.97297297297297303
265  0.94594594594594594
266  0.91891891891891897
267  0.83783783783783783
268  0.97297297297297303
269  1.0
270  0.94594594594594594
271  0.91891891891891897
272  1.0
273  0.78378378378378377
274  0.97297297297297303
275  0.94594594594594594
276  0.86486486486486491
277  0.81081081081081086
278  0.97297297297297303
279  0.91891891891891897
280  0.86486486486486491
281  0.89189189189189189
282  0.94594594594594594
283  0.89189189189189189
284  1.0
285  0.89189189189189189
286  0.86486486486486491
287  0.86486486486486491
288  0.86486486486486491
289  0.89189189189189189
```

Figure 4.3.1

mlp_result.txt

```
316  97.297297297297305
317  100.0
318  94.594594594594597
319  97.297297297297305
320  94.594594594594597
321  94.594594594594597
322  94.594594594594597
323  100.0
324  94.594594594594597
325  94.594594594594597
326  97.297297297297305
327  97.297297297297305
328  94.594594594594597
329  100.0
330  94.594594594594597
331  91.891891891891902
332  97.297297297297305
333  97.297297297297305
334  91.891891891891902
335  97.297297297297305
336  94.594594594594597
337  89.189189189189193
338  94.594594594594597
339  91.891891891891902
340  91.891891891891902
341  100.0
342  97.297297297297305
343  100.0
344  97.297297297297305
345  94.594594594594597
346  100.0
```

Figure 4.3.2

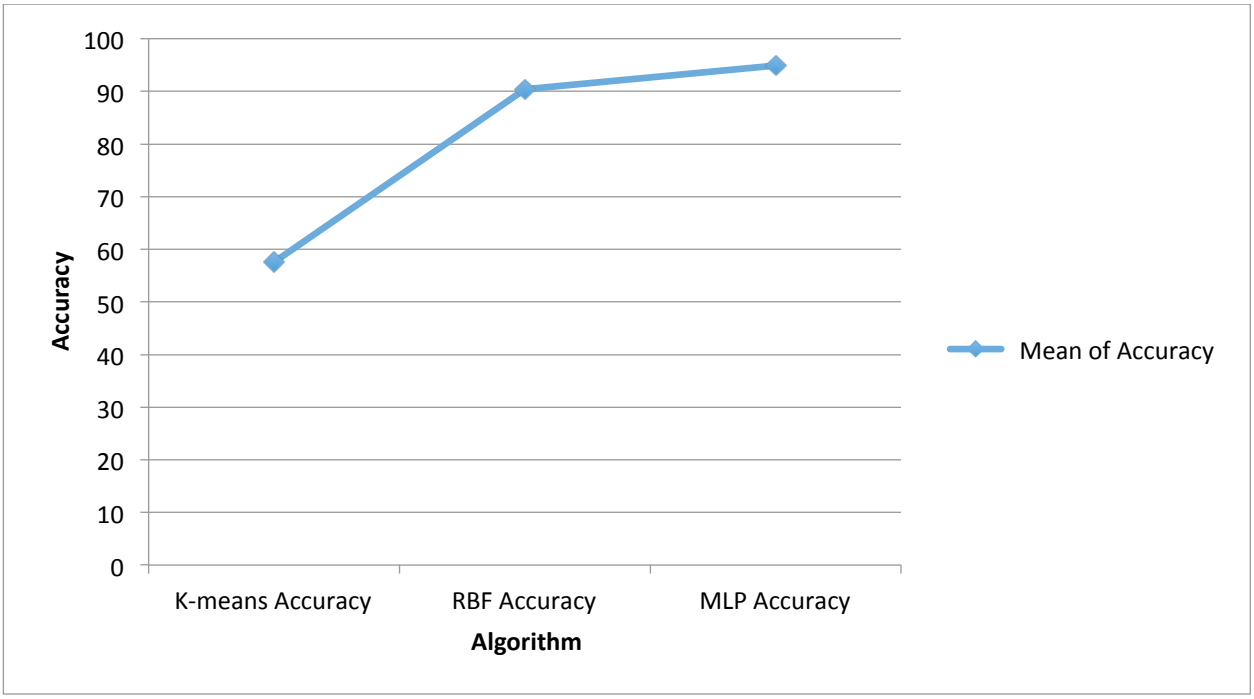

Figure 4.3.3: Line chart of mean value for each accuracy result

By using the clustered column chart, we can directly see the differences between unsupervised learning and supervised learning. Respectively, Figure 4.3.4, Figure 4.3.5, and Figure 4.3.6 are corresponding column charts of k-means, RBF, and MLP.

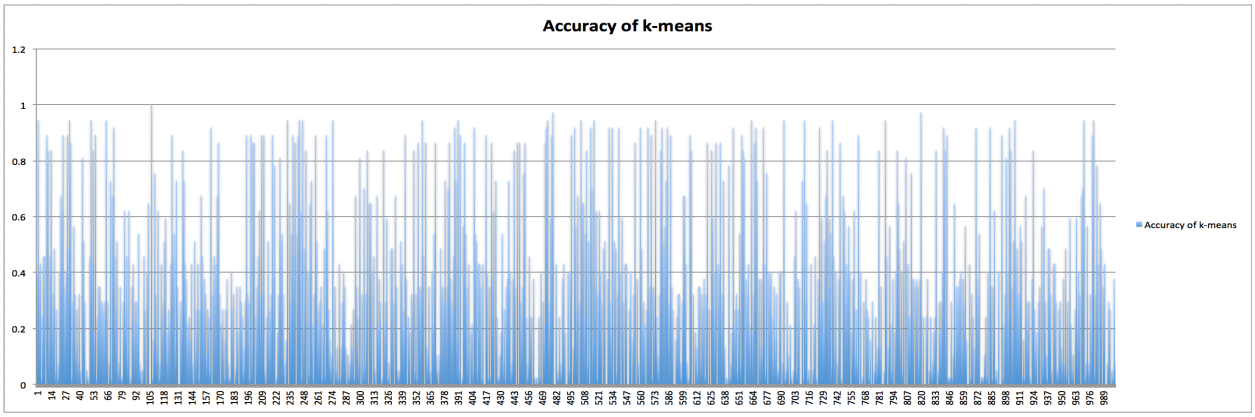

Figure 4.3.4: Column chart of k-means accuracy results

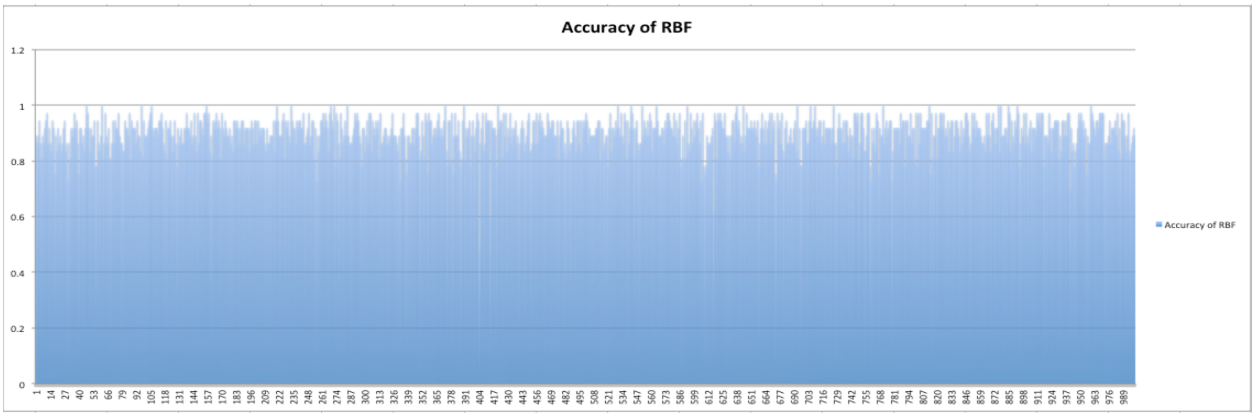

Figure 4.3.5: Column chart of RBF accuracy results

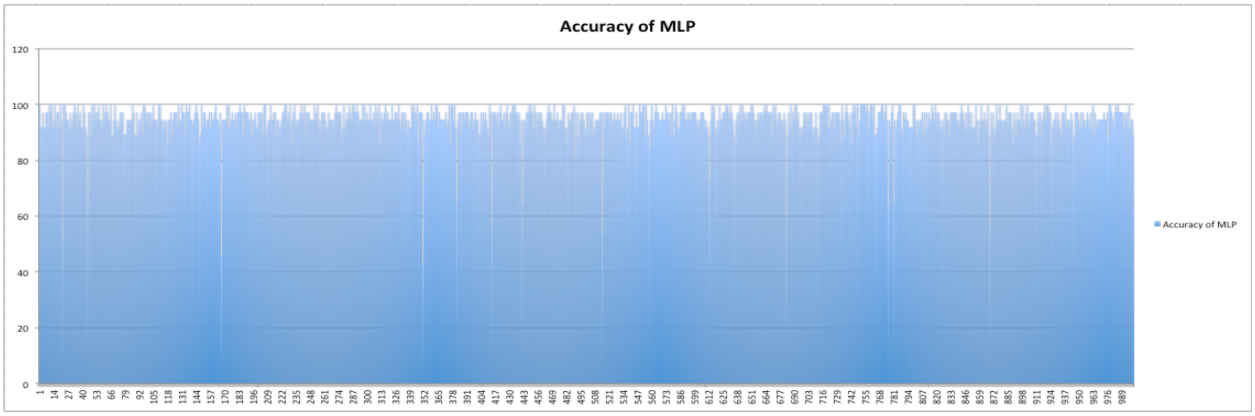

Figure 4.3.6: Column chart of MLP accuracy results

Comparing these three figures, we find that the accuracy of k-means is much more unstable. It varies between 0 and 0.98. Many of these accuracy results are below 0.5. There may be two reasons that result in this situation. One reason is every time using the k-means, the clusters centroids are initialized by taking random instances from the dataset. The other reason is that the use of the `shuffle` method changes the order of elements before clustering. Therefore, the accuracy changes every time.

5 Conclusion

In this paper, k-means is a good method to monitor the progression of students' academic performance to help the academic planners to make decisions.

However, comparing with RBF and MLP, we find that the accuracy of k-means is unstable, and k-means has the lower mean accuracy.

Our implementation has some limitations. On the one hand, the dataset is onefold. We only implement on one iris dataset, which is not enough. It is necessary to find some other datasets and make our implementation more applicable. With different kinds of datasets, our result can be tested and verified. On the other hand, we use only three algorithms to compare the accuracy results. In order to obtain the integrity of this project, more data mining algorithms can be added to the comparison. It has some long-term significance. For instance, it can help us to decide which algorithm is suitable for the specified dataset.

Our implementation can also be improved in the following 3 aspects. First, we can improve the data preprocessing. We used the `shuffle` method to rearrange the order of the elements at the beginning part. This is a random way. We can use some other arrangement to improve stability. Second is about the method of data chosen. We chose the data randomly from the dataset. This is a good way to grantee the randomness. However, it ignores the effectiveness. The solution is to create a rule for effectively choosing the data so that the result could be better. The third part is the measure of distance used for assigning records. Euclidean distance is chosen in our implementation. In further iteration, we can use some other measures, and compares them to find the best way.

References

[1] Oyelade, O. J., O. O. Oladipupo, and I. C. Obagbuwa. "Application of k Means Clustering algorithm for prediction of Students Academic Performance." *arXiv preprint arXiv: 1002.2425* (2010).
[2] Cortez, Paulo, and Alice Maria Gonçalves Silva. "Using data mining to predict secondary school student performance." (2008).
[3] Al-Radaideh, Qasem A., Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar. "Mining student data using decision trees." *International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan*. 2006.
[4] Thomas, Emily H., and Nora Galambos. "What satisfies students? Mining student-opinion data with regression and decision tree analysis." *Research in Higher Education* 45.3 (2004): 251-269.