Improving the *k*-means algorithm

James Phillips

MTSU: Graduate Student

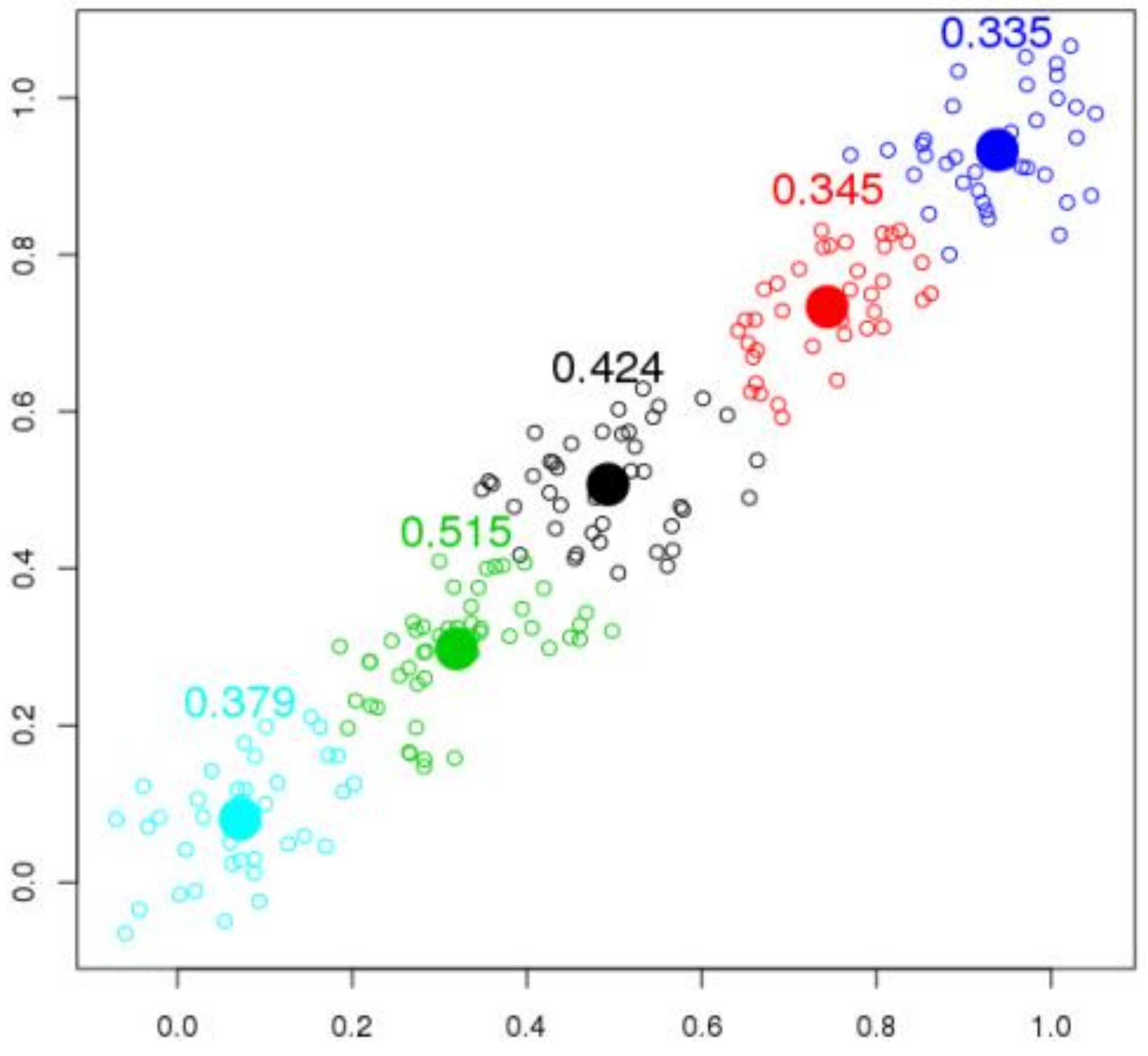
Abstract

The *k*-means algorithm is one of the most widely used clustering algorithms in many different fields; it can be used in many areas from helping diagnose cancer patients to species classification. With that comes its follies though, *k*-means has some challenges with run time and accuracy. While the algorithm can get reasonable accuracy when clustering it is heavily reliant on the number of *k* centers desired, the selection of initial starting points, and having the data in its most usable state. Most usable state in this context means that the data must have labels for testing, a well spread distribution of data within each classification, and contains few outlier data points. These are all problems that I have worked with while attempting to improve the *k*-means algorithm. By combining methods of data censoring and initial cluster selection I have improved the accuracy and efficiency of the algorithm.

Introduction

The *k*-means algorithm works by taking in a set of data points and a total number of centers. Using this information *k* centers are generated randomly. Then, the algorithm checks which center each data point is closest to and moves it into that centers grouping. Once all data points are assigned, the centers are recalculated and the process repeats until no changes are detected. This provides a learned relationship pattern amongst data points. However, he never intended for the algorithm to be used for exact groupings but more so a reasonably good grouping that would help the user understand abundant amounts of N-dimensional data.

With *k*-means clustering we assume that the data can be represented using a smaller number of data points which express the average or mean behavior of the data at each point. With the growing world of Big Data, being able to reduce data sets that contain Gigabytes to Terabytes worth of data is becoming more of a necessity then ever before and this is what *k*-means is great at as can be seen in Figure 1 below.



Problem/Aim

The *k*-means algorithm comes with it's own set of drawbacks though. The original intent for the algorithm was for it to run several times with different starting values and to see any skews in the data that could be lowering the accuracy of the algorithm. Another reason the multiple runs are needed is that the algorithm has a good chance of showing poor convergence of the clusters. This comes from the randomness of the initial cluster selection. Because of this randomness the clusters can run into issues of too few data samples or they can be centered so poorly that they have no data samples at all. This is where the aim of my project begins. I began by developing a new *k*-means algorithm to improve the accuracy and efficiency. Later, I tested my version of *k*-means against the base version of *k*-means for comparison.

Methods

Methods:

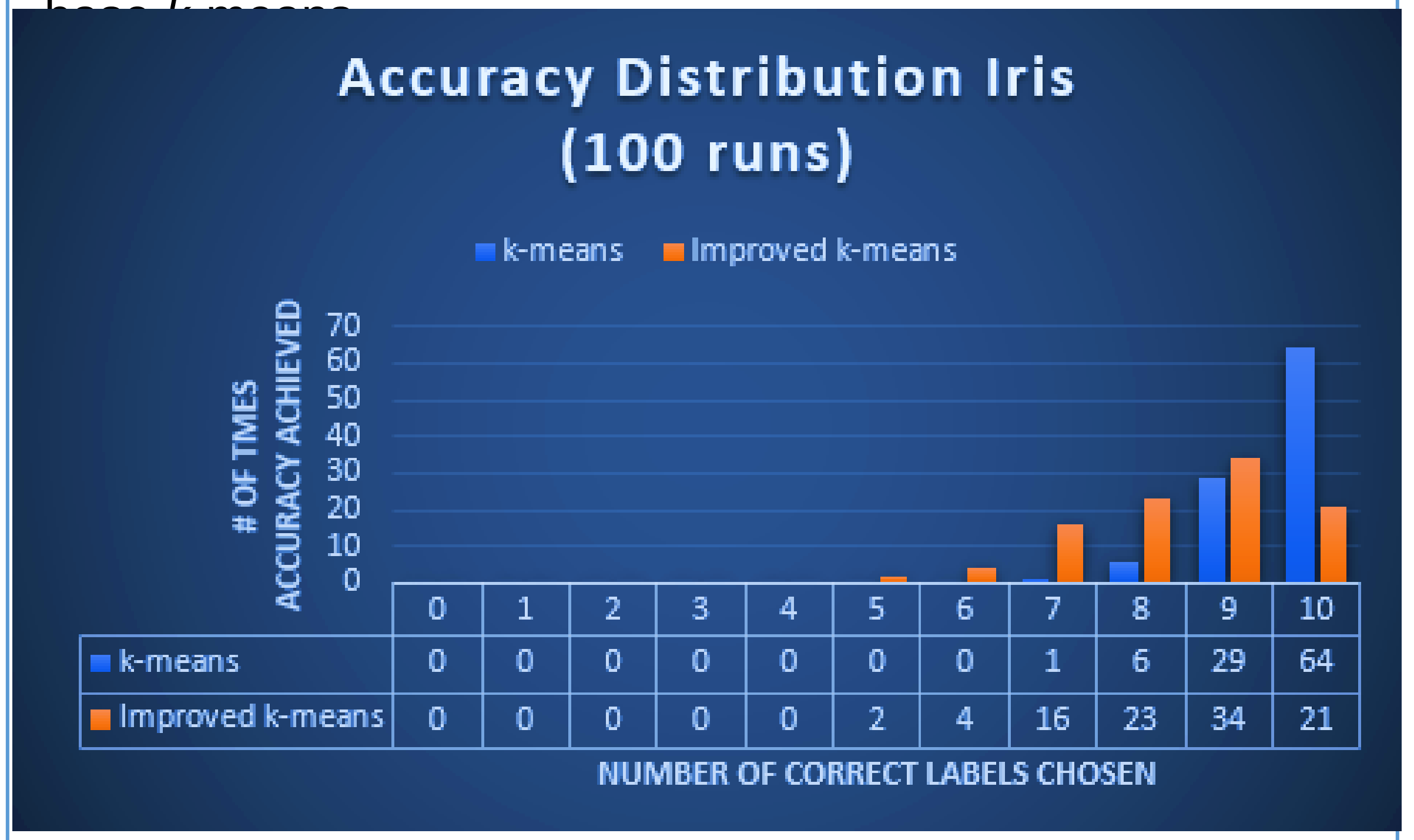
- First, implement the base *k*-means functions
- This function should:
    - Take in the normal data points and labels
    - Select Centroids Randomly throughout the data field.
    - Each data point should select its closest center based on Euclidean distance as shown in the figure below.
    - The centers then adjust to be in the middle of the points that are closest to it.
    - This process is repeated until no change is found

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

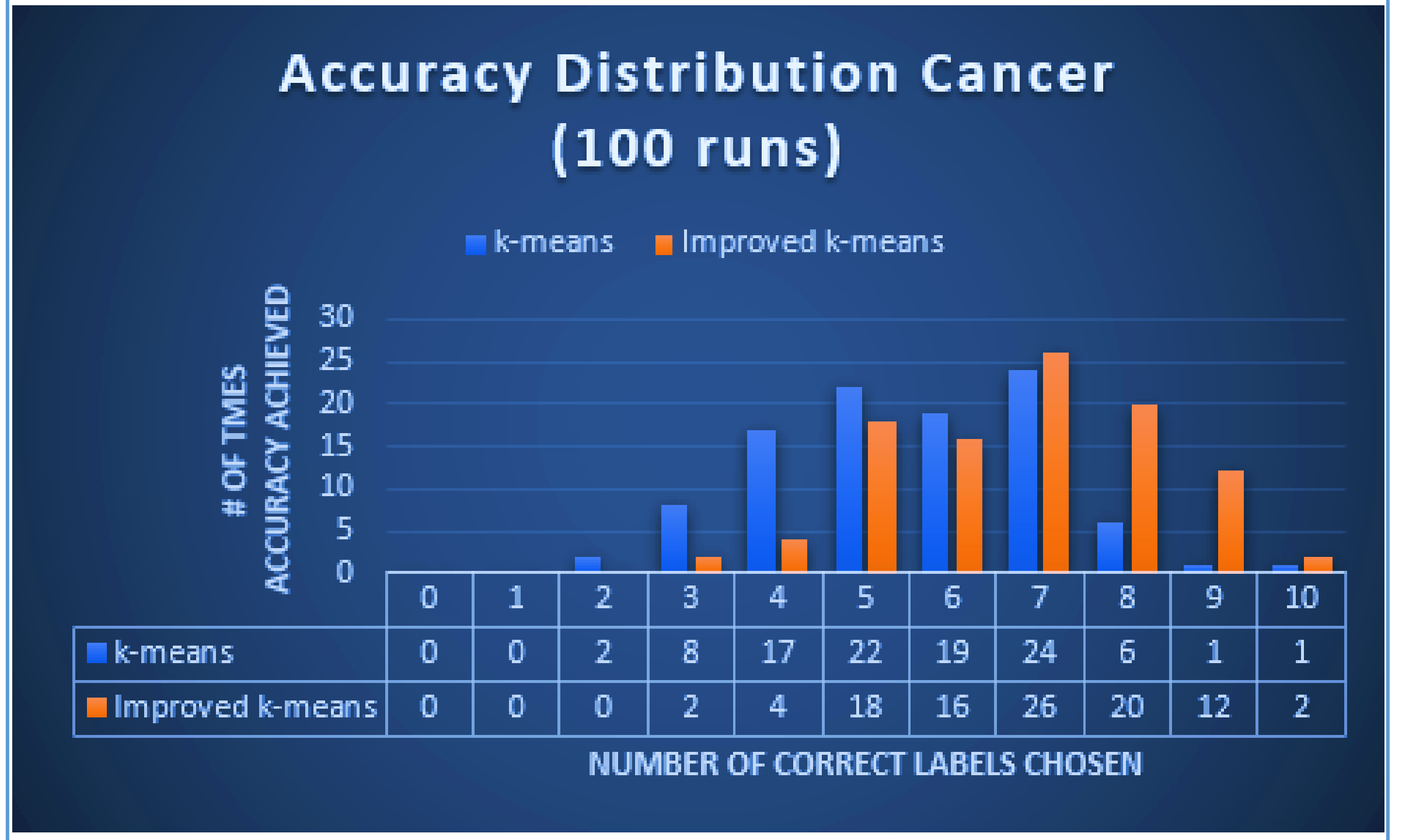
- Second, implement the Improved *k*-means function:
- This function should:
    - Take in a data set that has been zero centered and had its outliers removed for training purposes.
    - The first center should be selected randomly
    - The second center is the farthest point from the first center
    - Every other center is then selected as the farthest point from all previous centers.
        - To do this I calculate the Euclidean distance from each point to each center and sum them together per point. Then the top 10 highest distance measures have their standard deviation taken and the one with the lowest standard deviation is selected.
    - Then only readjust the centers once as this provides the largest gain in accuracy in respect to the time it takes to implement.

Results

To test the functionality of my program I ran each function 100 times and recorded the time and the accuracy of each run against two data sets. The first data set is one that works very smoothly with the base *k*-means implementation while the second is one that showed poor accuracy when using base *k*-means.



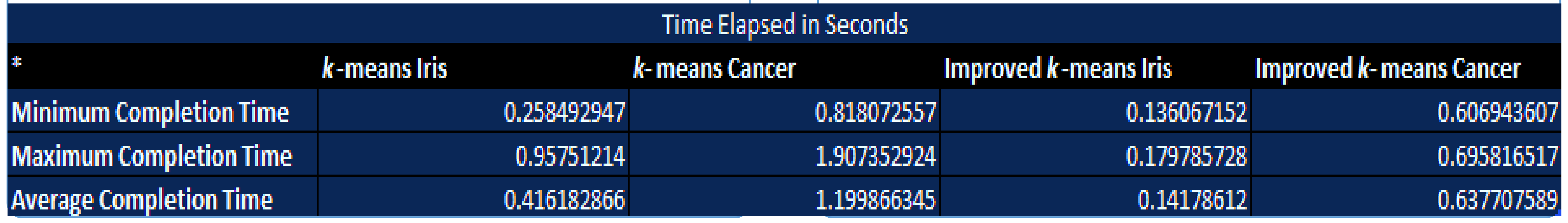
When using the improved *k*-means own the Iris data set I saw a wider accuracy range, but it only had a slight difference in the accuracy average.



However when testing with the Cancer data set, I found that the accuracy tended to be much higher when compared the base *k*-means function.

With both tests the optimal number of centers was selected for each data set. In the case of the Iris data set I used 10 center and with the cancer data set I used 90 centers.

Below is the time elapsed measurements for both implementation as well. For both data sets the Improved *k*-means algorithm shows to run in almost half the time of the base *k*-means algorithm.



Time Elapsed in Seconds

*	*k*-means Iris	*k*- means Cancer	Improved *k*-means Iris	Improved *k*- means Cancer
Minimum Completion Time	0.258492947	0.818072557	0.136067152	0.606943607
Maximum Completion Time	0.95751214	1.907352924	0.179785728	0.695816517
Average Completion Time	0.416182866	1.199866345	0.14178612	0.637707589

Conclusions

From these results I believe I have achieved the goals of this project. The accuracy of the Improved *k*-means function proves to be better against data sets that the base *k*-means function does poorly with as shown by the Accuracy Distribution of the Cancer data set. It only performs slightly worse with the Iris data set and still has the capability of performing at the same accuracy as the base algorithm. The Improved algorithm is also much faster in terms of time elapsed then the base algorithm running in nearly half the time. In terms of time complexity the base algorithm is running at O(n^2) while the Improved algorithm is running at O(n).

Future work with this project could extend into determining the number of centers required during runtime rather then having the user decide before execution. This algorithm was also not test on large data sets or data sets with missing values so those could be other avenues that could be taken with future project.

I believe that the *k*-means algorithm is going to become more useful as we continue to grow in the world of Big Data and as Big Data grows so should the algorithms that we use to process this data.

References

Bradley, P. S., Bennett, K. P., and Demiriz, A. Constrained k-means clustering. *Microsoft Research, Redmond 20*, 0 (2000), 0.

Bradley, P. S., and Fayyad, U. M. *Refining initial points for k-means clustering*. Citeseer.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Fränti, P., and Sieranoja, S.How much can k-means be improved by using better initialization and repeats? *Pattern Recognition* 93 (2019), 95–112.

MacQueen, J., et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, Oakland, CA, USA, pp. 281–297.

Su, T., and Dy, J.A deterministic method for initializing k-means clustering. In *16th IEEE International Conference on Tools with Artificial Intelligence* (2004), IEEE, pp. 784–786.

Yedla, M., Pathakota, S. R., and Srinivasa, T. Enhancing k-means clustering algorithm with improved initial center. *International Journal of computer science and information technologies* 1, 2 (2010), 121–125.