Project Report on

"Efficient clustering algorithm to segregate tests based on execution behavior"

For

PTC (India)

Submitted By

Ganesh Kalidas Londhe

For

Savitribai Phule Pune University

(MSc Computer Science – Semester-IV)
(2018-2019)

Indira College of Commerce & Science, Pune 33

ACKNOWLEDGEMENT

It is my proud privilege to express gratitude to the entire management of Indira College of commerce and science (ICCS)-MSc. Computer Science and teachers of the institute for providing me with the opportunity to avail the excellent facilities and infrastructure of the institute. The knowledge and values inculcated have proved to be of immense help at the very start of my career.

I am grateful to **Dr. Janardan Pawar** (Principal In-Charge and HOD, ICCS MSc. Computer Science), **Amol Godbole, Section manager**, and **Mrs. Manisha Patil, Ms. Sarita Byagar** (Internal Guide, ICCS MSc. Computer Science) for their astute guidance, constant encouragement and sincere support for this project work.
I also thank my project mentors who showed their concerns for my work, encouraged me to keep my best foot forward and gave valuable suggestions which not only helped me in my project work but will be useful in future too.

I would like to thank **PTC(India)** for providing me with an opportunity to pursue my industrial training, as it is an important part of the MSc. Computer Science course and it is the one that exposes you to the industry standards and makes you adapt yourself to the latest trends and technologies. At the same time, it gives an experience of working on a live project. I feel proud and privileged in expressing my deep sense of gratitude to all those who have helped me in presenting this assignment. I would be failing in my endeavor if I do not place my acknowledgment.

Sincere thanks to all my seniors and colleagues at company for their support and assistance throughout the project.

Index:

1. Introduction:

1.1 Company Profile:

Intern at software development team named “Creo Licensing and Installation”. Where we write, build and enhance the security for the Licensing and Installation part of the product.

1.2.1 Existing system and Need for system:

Current system is only be able to show the last time the test has been run and how much time test took to complete.

1.2.2 Need for new System:

Newly developed system has following features:

- shows how many times the test has been run on the product
- what is the minimum time it took to run the test
- what is the maximum time it took to run the test
- mean timing
- standard deviation
- median

1.3 Scope of the work:

Scope of the work is limited to the respective company and also limited to the current product.

1.4 Operating Environment

Hardware – 2GB RAM, Dual core processor (2GH)

Software(Operating System) – Windows, Linux(Platform independent)

2.1 Proposed System:

This project will separate tests based on their timings. This project is about sorting the tests. Including new features like automation, accuracy more data and time saving using Data Mining and Machine Learning based techniques.

2.1.1 Feasibility study:

Technically it's possible to complete the project using existing technologies. As all the technical resources are available within the organization. No estimated cost as the project is carried out along-side regular work.

Since no money is evolved the project is fully profitable. No aspect of the project conflicts with legal requirements like zoning laws, data protection acts or social media laws. This project fits in scheduling feasibility as time required to complete the project is much low.

2.2 Objective of the system:

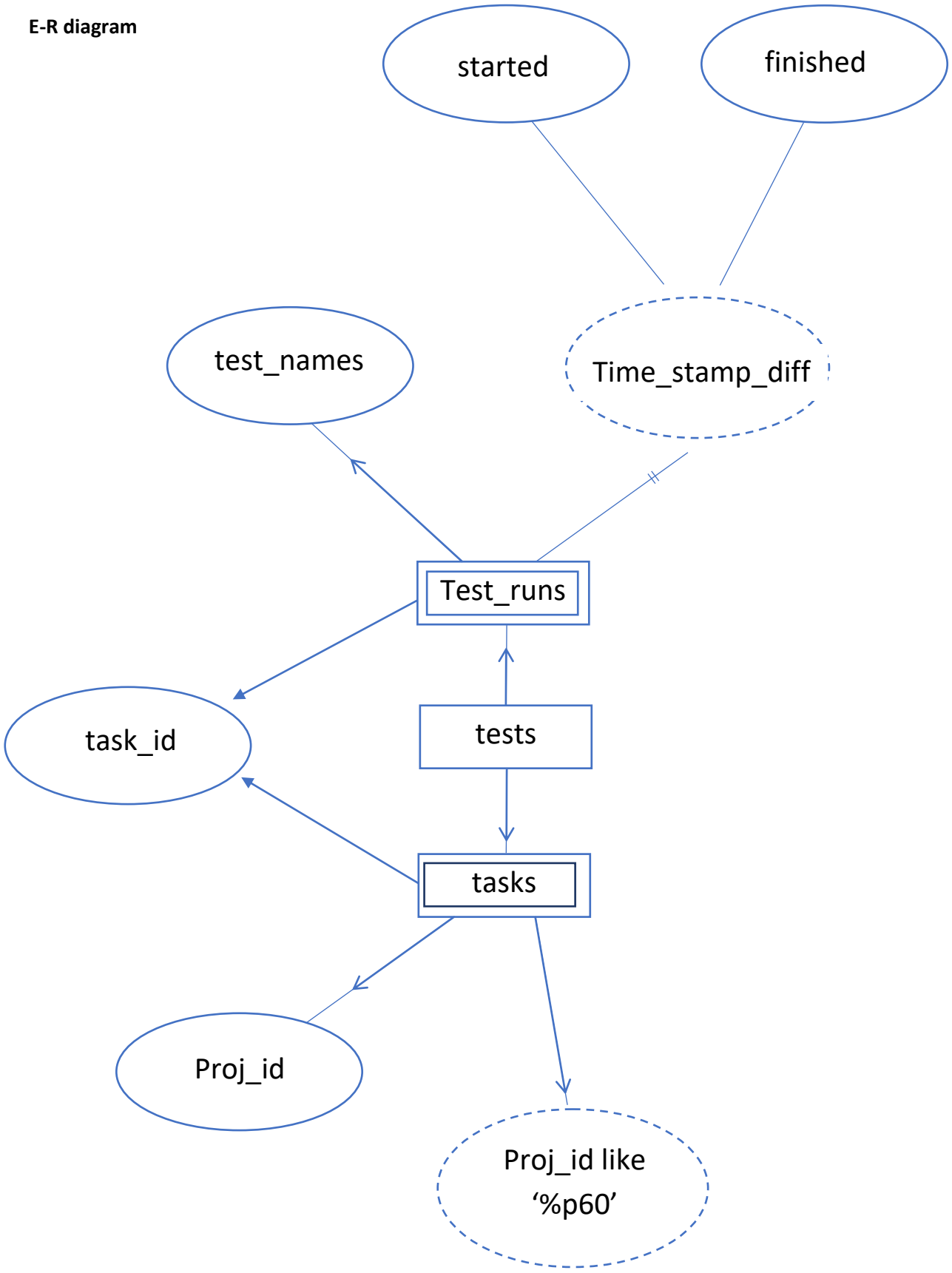
Separate the tests based on time taken.

2.3 User Requirements:

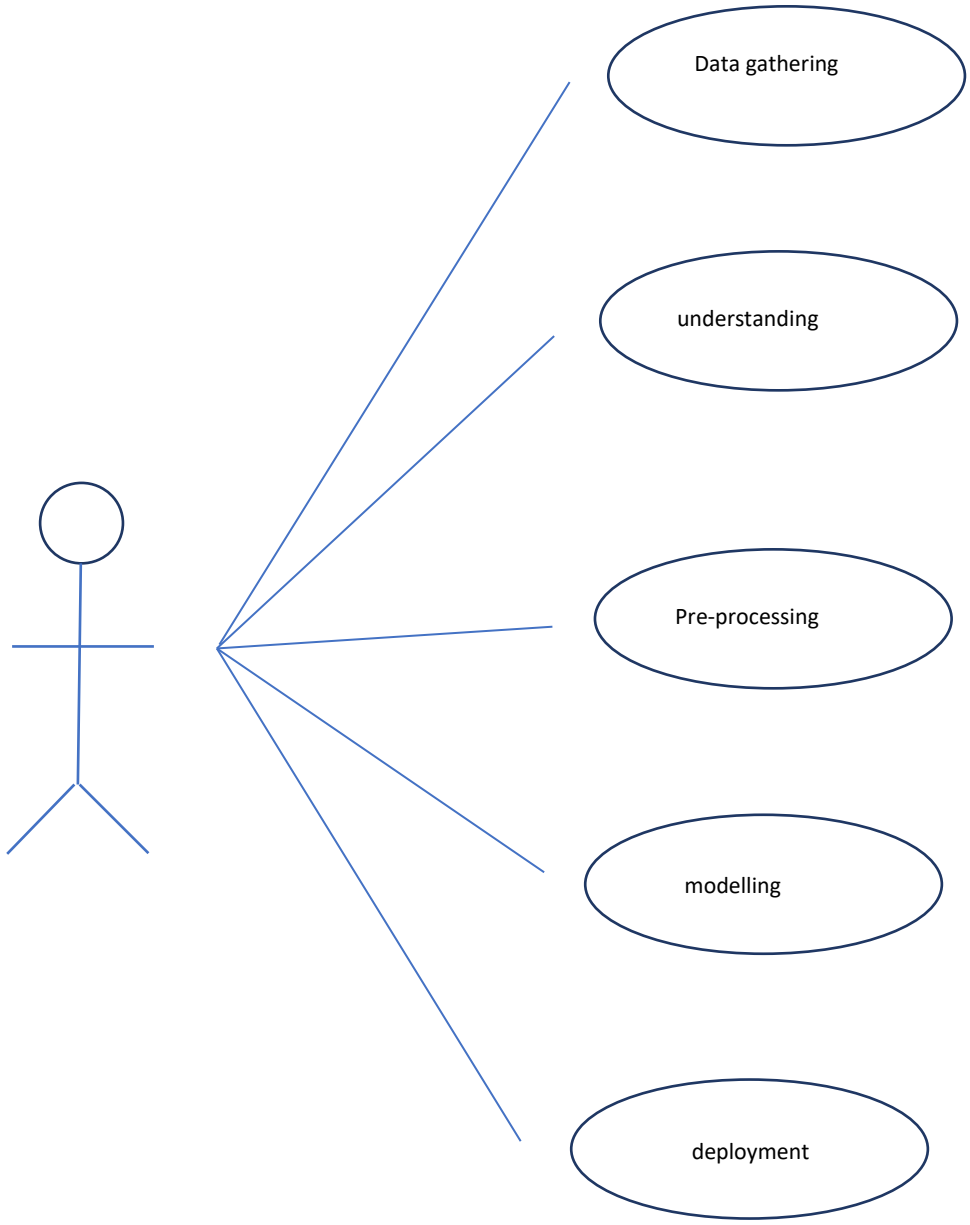
Develop and efficient clustering algorithm to segregate tests based on their execution behavior

3. Analysis and Designs
3.2 E-R Diagram:

E-R diagram

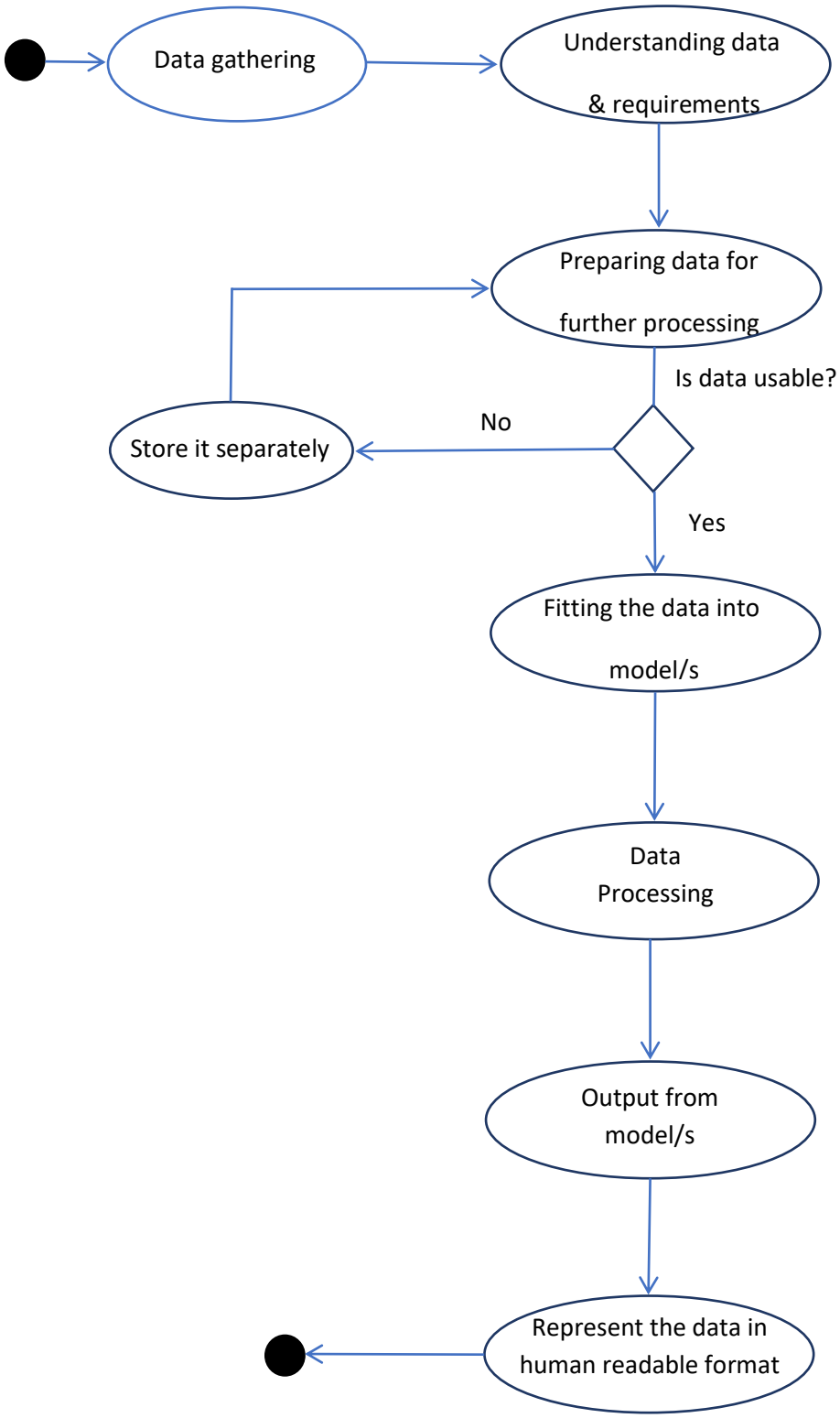


3.2 Use Case Diagram:



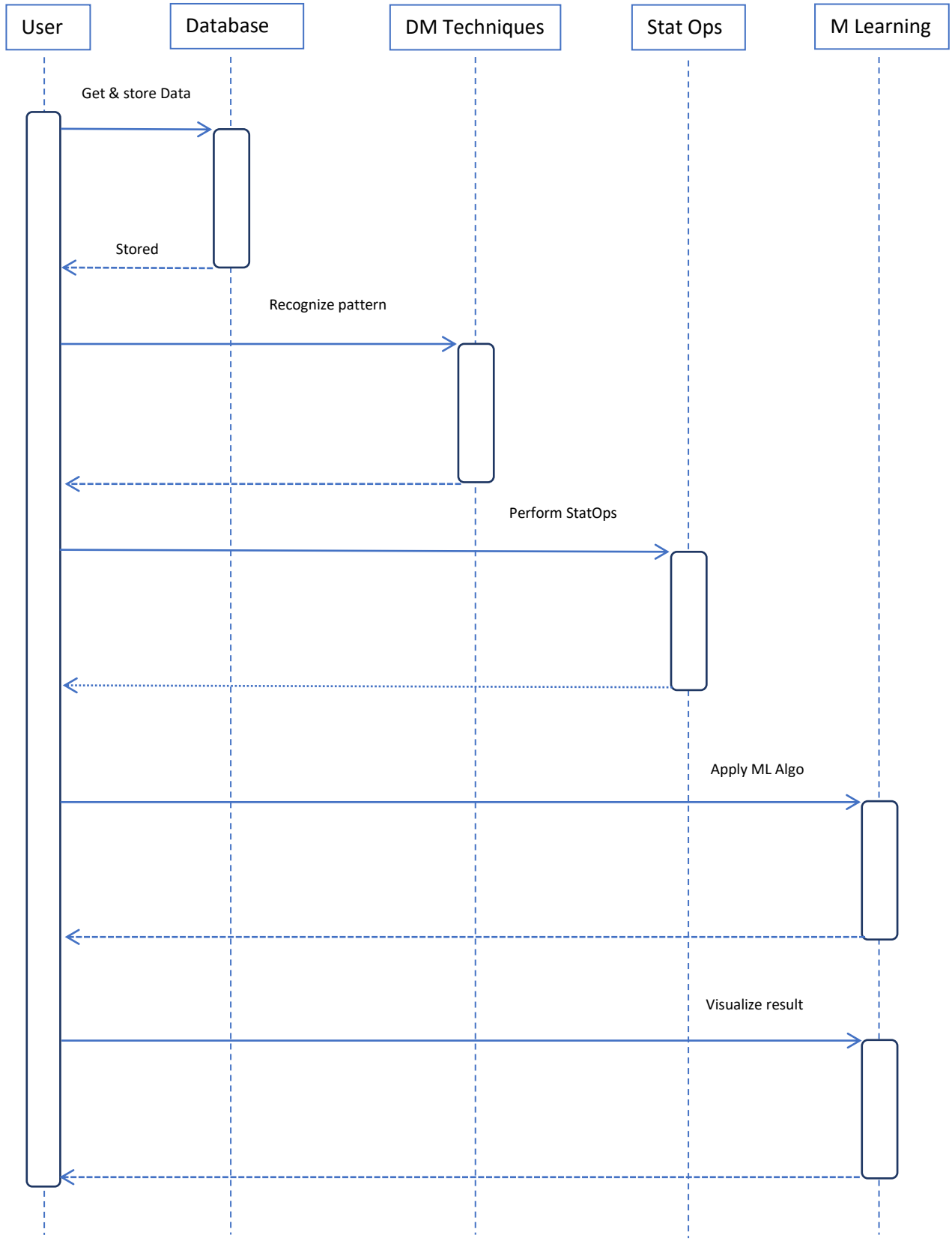
Use Case Diagram

3.3 Activity Diagram:



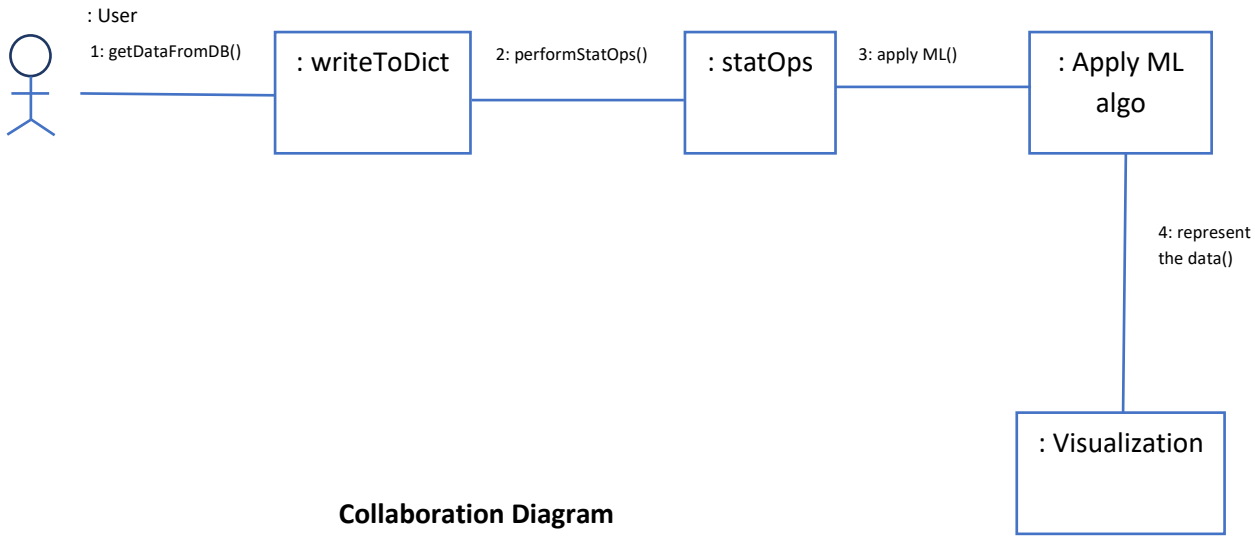
Activity Diagram

3.4 Sequence Diagram:



Sequence Diagram

3.5 Collaboration Diagram:

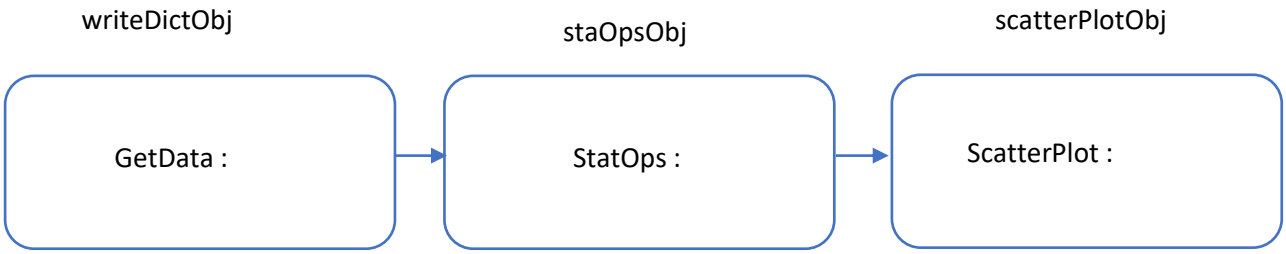


Collaboration Diagram

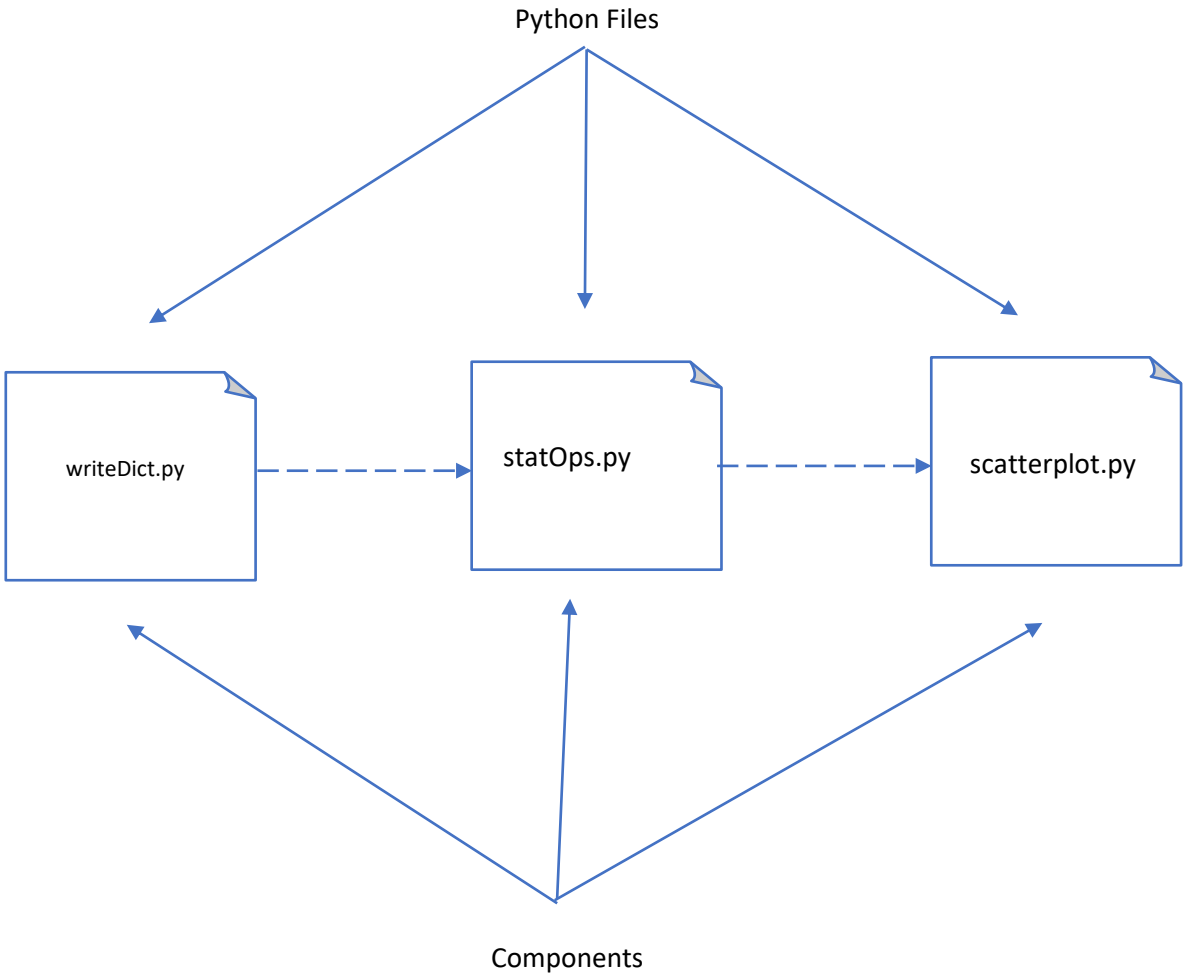
3.6 Class Diagram:

Class Diagram

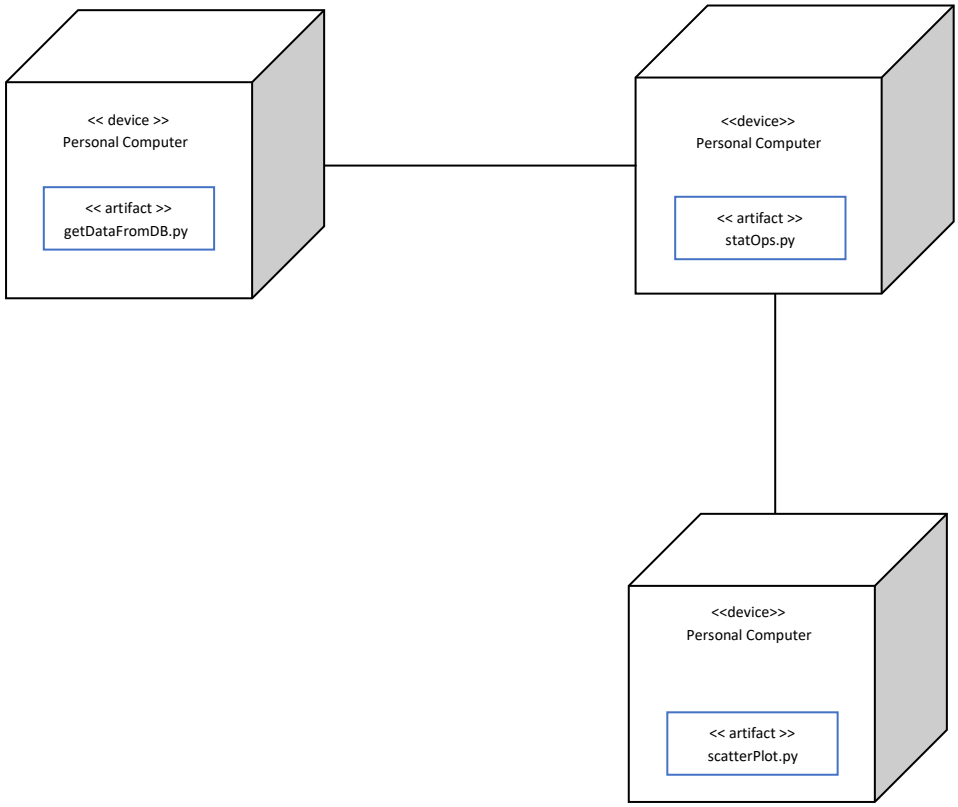
3.7 Object Diagram



3.8 Component Diagram:



3.9 Deployment Diagram:



Deployment Diagram

4. User Manual

This is a scripted program. User need to hit the ENTER KEY to run the program.

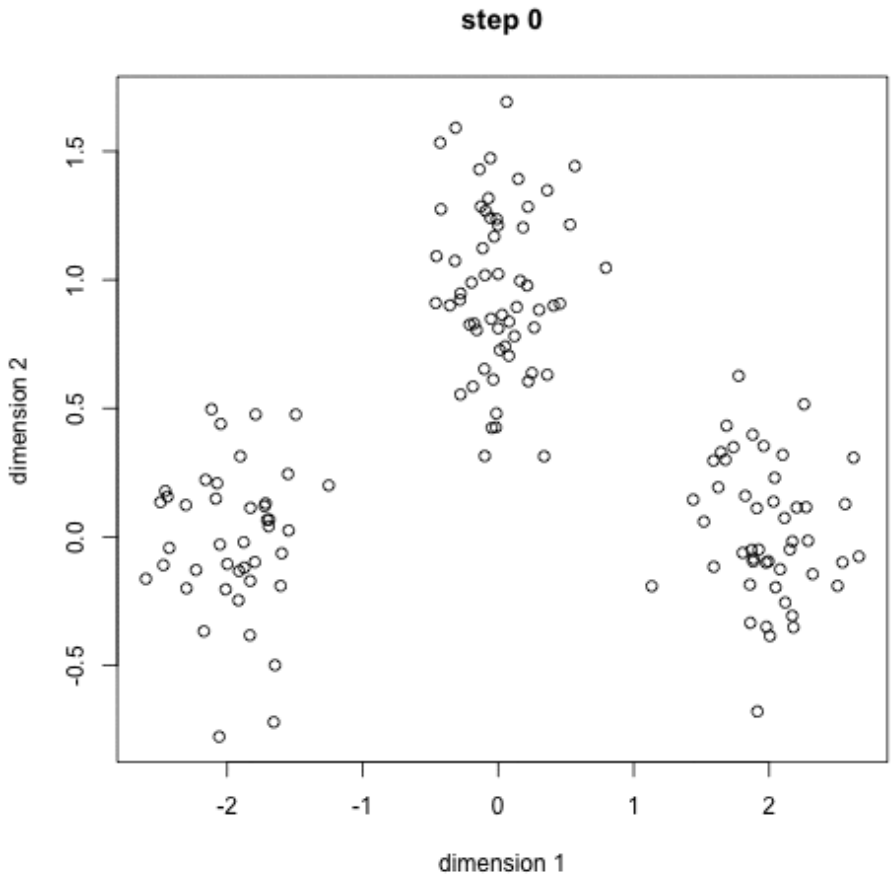
Brief about K-Means:

Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

In Data Science, we can use clustering analysis to gain some valuable insights from our data by seeing what groups the data points fall into when we apply a clustering algorithm.

K-Means Clustering

K-Means is probably the most well know clustering algorithm. It's taught in a lot of introductory data science and machine learning classes. It's easy to understand and implement in code! Check out the graphic below for an illustration.



K-Means Clustering

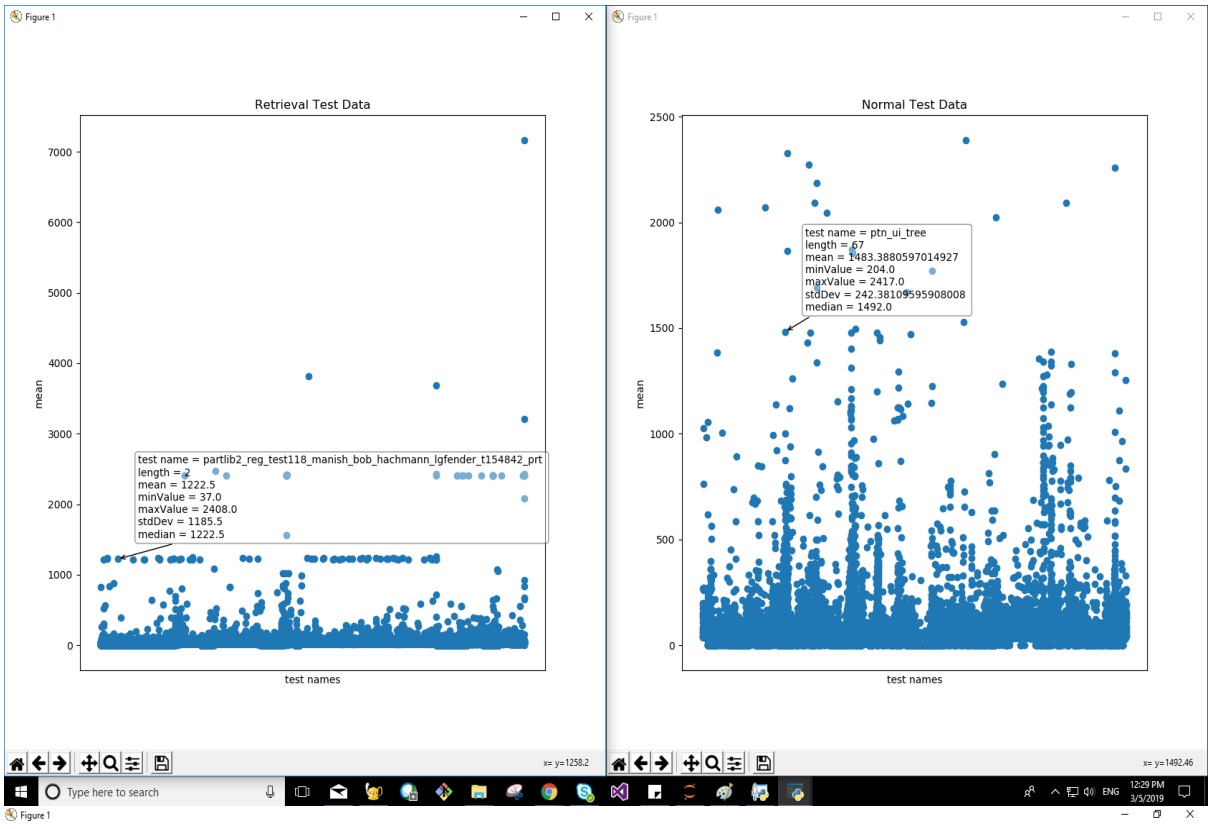
1. To begin, we first select a number of classes/groups to use and randomly initialize their respective center points. To figure out the number of classes to use, it's good to take a quick look at the data and try to identify any distinct groupings. The center points are vectors of the same length as each data point vector and are the "X's" in the graphic above.
2. Each data point is classified by computing the distance between that point and each group center, and then classifying the point to be in the group whose center is closest to it.
3. Based on these classified points, we recompute the group center by taking the mean of all the vectors in the group.
4. Repeat these steps for a set number of iterations or until the group centers don't change much between iterations. You can also opt to randomly initialize the group centers a few times, and then select the run that looks like it provided the best results.

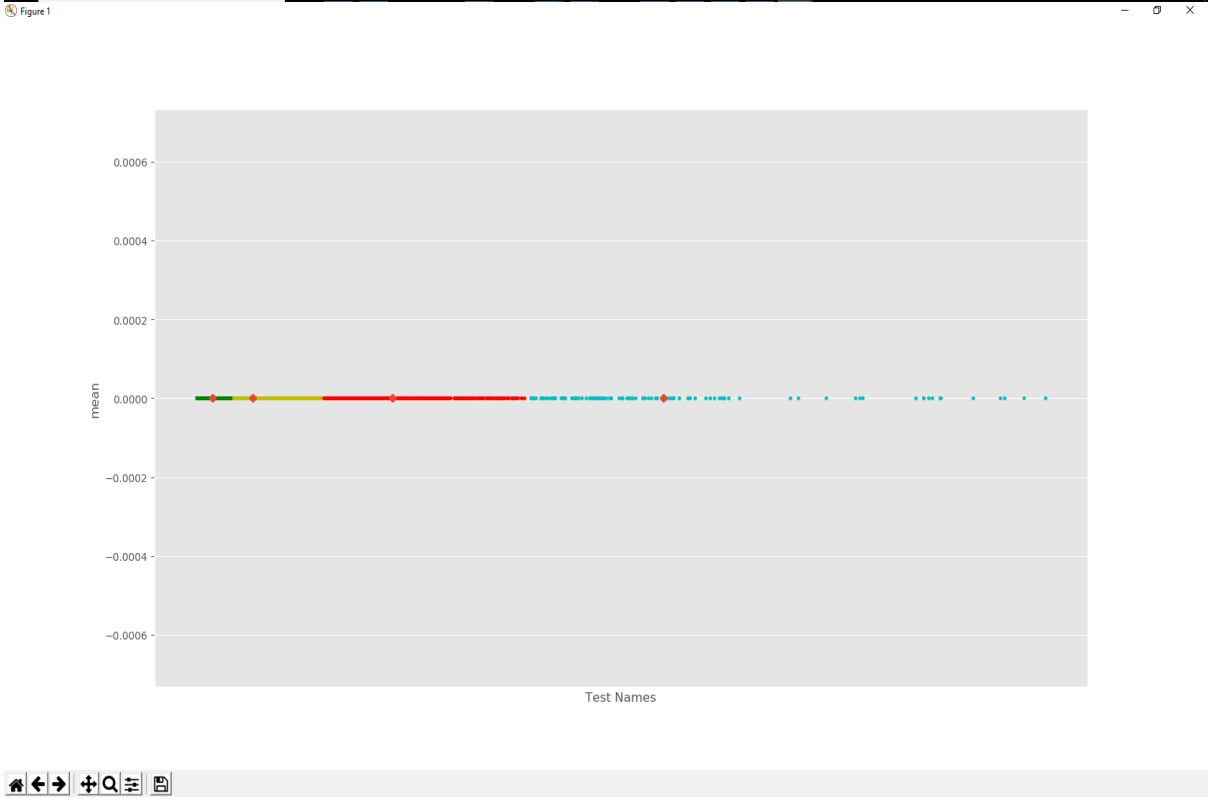
K-Means has the advantage that it's pretty fast, as all we're really doing is computing the distances between points and group centers; very few computations! It thus has a linear complexity $O(n)$.

On the other hand, K-Means has a couple of disadvantages. Firstly, you have to select how many groups/classes there are. This isn't always trivial and ideally with a clustering algorithm we'd want it to figure those out for us because the point of it is to gain some insight from the data. K-means also starts with a random choice of cluster centers and therefore it may yield different clustering results on different runs of the algorithm. Thus, the results may not be repeatable and lack consistency. Other cluster methods are more consistent.

K-Medians is another clustering algorithm related to K-Means, except instead of recomputing the group center points using the mean we use the median vector of the group. This method is less sensitive to outliers (because of using the Median) but is much slower for larger datasets as sorting is required on each iteration when computing the Median vector.

5 Output report:





6. Future Enhancements :

daily, monthly, quarterly and yearly reports are not generated.

Project can go real and give a live status of the tests.

7. Conclusion:

The project has completed the basic needs and requirements of the client and is ready to use.

8.References:

https://www.tutorialspoint.com/dwh/dwh_data_warehousing.htm
https://www.geeksforgeeks.org/supervised-unsupervised-learning/
https://en.wikipedia.org/wiki/Pandas_(software)
https://en.wikipedia.org/wiki/NumPy
https://en.wikipedia.org/wiki/Matplotlib
https://en.wikipedia.org/wiki/SQLAlchemy
https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html