**Finding anomalies from Data.**

**Pre-requisites**

Required Setup : Jupyter, Python and Postgres DB.

**1. How it is Done.**

Importing the dataset (CSV File) into a database(postgres) Using Python

The script will read the given csv file and load into target postgres table.

* **Source : CSV format.**
* **Target: Postgres Table**

Below 2 libraries are used to communicate with Input and target.

* **import csv**
* **import psycopg2**

**Postgres table definition.**

Mapping is done with respect to the given csv file.

*drop table if exists base.device\_series;*

*CREATE TABLE base.device\_series*

*(*

*dtime timestamp,*

*device\_identifier text,*

*bytes\_total integer,*

*bytes\_per\_second double,*

*successes integer,*

*failures integer,*

*fetch\_time integer*

*);*

Using the csv reader method each row is inserted into target postgres table.

with Delimiter being comma and quote character as double quotes.

**why did you choose this approach and how do we run it?**

This is one of the approaches, there are several approaches to this problem, depending on the infra setup and tools available for that particular organization.

This is done using Jupyter,Python & Postgres which are open source.

**Anomalies found in the data, The efficiency of the approach to finding the anomalies.**

**Anomaly Result set:**

The report will list out all the anomalies occurred, Start Time , End time for the given dataset.

Start and end time will reflect how long the downtime lasted for. Thus getting the exact interval for that particular Anomaly.

CTE Query is used for anomalies, Instead of CTE, CTAS can be used as well, which will write the result to table in the disk.

**How to run this application :**

1.To run this script python 2.7 is to be setup on the system.

2.Postgres DB is required, however this application can be scaled to other DB's with required libraries installed

* for example ***to use mysql as backend***
* ***!pip install mysqlclient***
* The same code will work with AWS Redshift as the Redshift uses postgres core.

**Given more time, what would you do to improve your solution?**

Will develop a distributed application for scalability and performance, assuming this will not be limited to a smaller dataset. Will explore more methods, algorithms how to improve and scale. Will visualize data using visualization tools or built in python libraries.

**Distributed Batch Application: (Subject to the required setup)**

**Pre-requisite: Hadoop Stack**

This is a 2 step process by creating 2 hive tables.

1.Hive external table

2. Hive Internal table with ORC format.

Copy the given csv file in Hive external HDFS Location then using a simple (select and insert into) Target ORC table. Which will be way faster and scale as the data grows.

Anomaly result set can be extracted with little or no modification to the query.

**Near Real time** **Streaming Application: (Subject to the required setup)**

**Pre-requisite: Hadoop Stack, Kafka,Spark**

Assuming these events are sent through kafka. under a particular topic.

Will do any pre-processing, Using Spark Dtreams Map them to spark dataframes and load these events to Target Near real time for low latency.