



## **ECE 9014: *Lumos 5G***

Group #09 :

Wenzhe Quan(251167062)

Hui Lin(251253777)

Siyuan Ma(251167028)

Yibo Wang(251141466)

Department of Electrical & Computer Engineering  
The University of Western Ontario

---

## **Table of Content**

- 1 Group Deliverables
  - 1.1 Project Description
  - 1.2 Descriptive data analytics problem
  - 1.3 Main Data Analytics Elements/Concepts
  - 1.4 Data Model for Analytics
  - 1.5 Data Warehouse (DW) Architecture
  - 1.6 Data Model Definition
  - 1.7 Data Warehouse Population (ETL)
  - 1.8 Build and Materialize Cubes
  - 1.9 Data Manipulation: Internal Schema
- 2 Individual Deliverables
  - 2.1 Extension 1: Name of the Extension
    - 2.1.1 Main Data Analytics Elements/Concepts
    - 2.1.2 Data Model Definition
    - 2.1.3 Data Warehouse Population (ETL)
    - 2.1.4 Build and Materialize Cubes
    - 2.1.5 Data Manipulation: Internal Schema
    - 2.1.6 Predictive Data Analytics problem
    - 2.1.7 Predictive Data Model
    - 2.1.8 Predictive Data Analytics Solution Pipeline
      - 2.1.8.1 Samples of the Predictive Data Analytics Solutions
      - 2.1.8.2 Predictive Data Analytics Solution: Accuracy & Performance
  - 2.2 Extension 2: Name of the Extension
    - 2.2.1 Main Data Analytics Elements/Concepts
    - 2.2.2 Data Model Definition
    - 2.2.3 Data Warehouse Population (ETL)
    - 2.2.4 Build and Materialize Cubes

- 2.2.5 Data Manipulation: Internal Schema
- 2.2.6 Predictive Data Analytics problem
- 2.2.7 Predictive Data Model
- 2.2.8 Predictive Data Analytics Solution Pipeline
  - 2.2.8.1 Samples of the Predictive Data Analytics Solutions
  - 2.2.8.2 Predictive Data Analytics Solution: Accuracy & Performance
- 2.3 Extension 3: Name of the Extension
  - 2.3.1 Main Data Analytics Elements/Concepts
  - 2.3.2 Data Model Definition
  - 2.3.3 Data Warehouse Population (ETL)
  - 2.3.4 Build and Materialize Cubes
  - 2.3.5 Data Manipulation: Internal Schema
  - 2.3.6 Predictive Data Analytics problem
  - 2.3.7 Predictive Data Model
  - 2.3.8 Predictive Data Analytics Solution Pipeline
    - 2.3.8.1 Samples of the Predictive Data Analytics Solutions
    - 2.3.8.2 Predictive Data Analytics Solution: Accuracy & Performance
- 2.4 Extension 4: Name of the Extension
  - 2.4.1 Main Data Analytics Elements/Concepts
  - 2.4.2 Data Model Definition
  - 2.4.3 Data Warehouse Population (ETL)
  - 2.4.4 Build and Materialize Cubes
  - 2.4.5 Data Manipulation: Internal Schema
  - 2.4.6 Predictive Data Analytics problem
  - 2.4.7 Predictive Data Model
  - 2.4.8 Predictive Data Analytics Solution Pipeline
    - 2.4.8.1 Samples of the Predictive Data Analytics Solutions
    - 2.4.8.2 Predictive Data Analytics Solution: Accuracy & Performance

---

## **1 Group Deliverables**

### **1.1 Project Description**

---

The Lumos 5G-v1.0 dataset is a dataset which contains extensive experiments and statistical analysis data to determine the factors that affect the 5G performance and quality. What's more, with the help of the dataset, the throughput can be predicted. This dataset contains: run\_num, seq\_num, latitude, longitude, direction, signal quantity, signal receive power, mobility mode, status and throughput. We create 9 entities to describe the factors that affect the 5G signal.

## 1.2 Descriptive data analytics problem

---

*Provide a brief description of the descriptive data analytics problem based on the Application domain of your chosen dataset. It should be in the form of an analytical structure includes quantifiable measurements along analytical (decision) variables of interest within your application domain, to produce high rich "information", which in turn can be a base to identify insights about the context of the application domain. It should include at least three analytical/decision variables*

throughput quality will be the descriptive data analytics problem for this project. The 5G dataset will be used to analyze this problem.

Throughput quality

- location
- signal strength

As shown above, this problem will be discussed based on 3 elements : location(where to detect the throughput),signal strength (combine the throughput) and the movement( walk or drive)

## 1.3 Main Data Analytics Elements/Concepts

---

*Provide a brief description of the main data elements/concepts related to the quantifiable measurements and the analytical (decision) variables required to support your descriptive data analytics problem*

Desirable Measurement would be:

- location(measure the effect in different location)
- strength (how the power of the signal affect through)

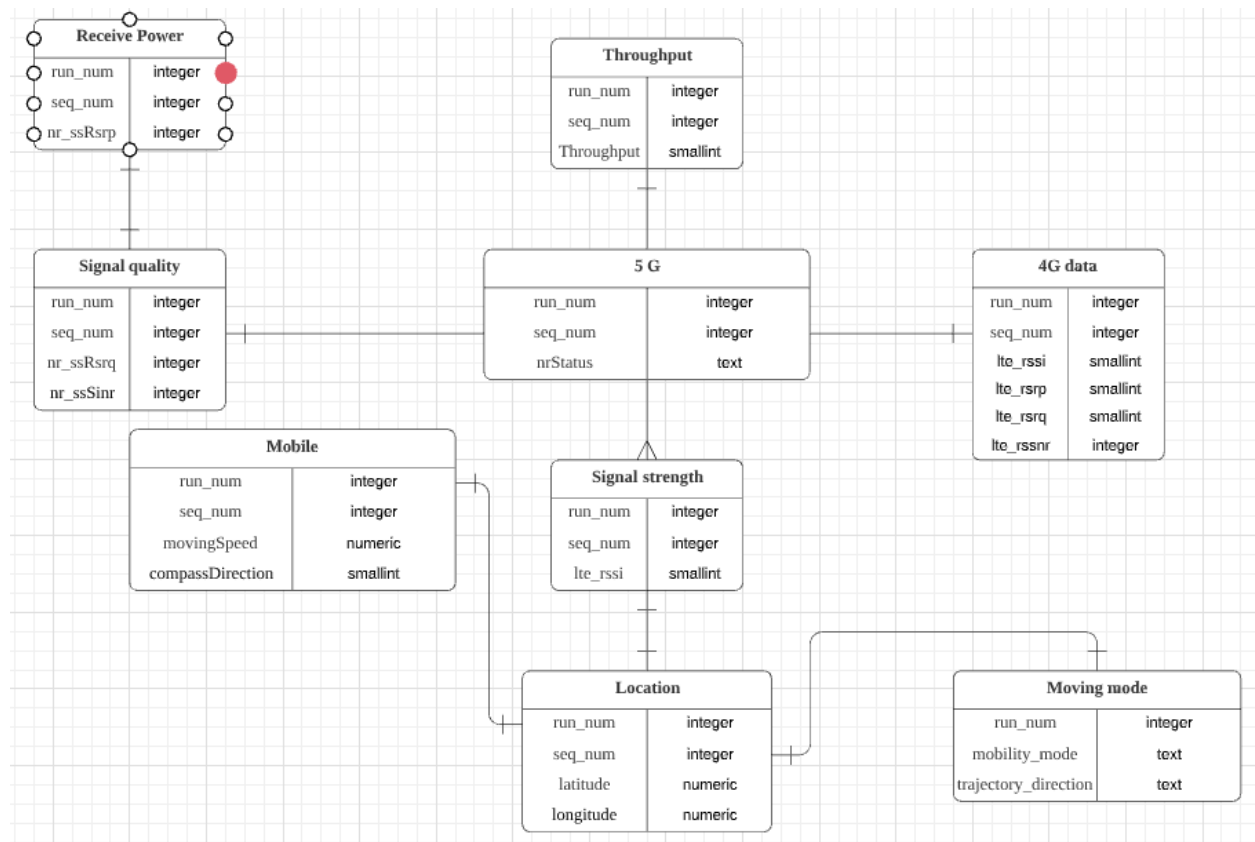
useful dimensions

- throughput
- ite-rsrp
- ite-rsrq

- longitude
- latitude

## 1.4 Data Model for Analytics

*Provide a brief description of an adequate data model capturing the data elements above for your descriptive data analytics. (for the disarms use Lucidchart or any CASE tool of choice)*



The central entity of this model is 5G, which is associated with several important features of different records of 5G. “run\_num” and “seq\_num” help us to identify which record it is, and “nrStatus” helps us to see if this record is completed in 5G signal.

“4G data” is the control group for comparison with 5G measurement data. It can help us visually see the gap between 4G and 5G in terms of signal quality, signal-to-noise ratio, signal strength, and received power. This is not directly concerned with our research this time, just as a comparison.

“Throughput” records the specific value of this feature. Throughput refers to the number of information bits correctly transmitted by a system in a unit of time, and the unit is bit/s. Throughput

refers to the actual transmission rate, so in general, the greater the throughput, the higher the transmission rate, and the faster the network speed.

“Signal quality” is mainly affected by “nr\_ssRsrq” and “nr\_ssSinr”. It directly shows us how good or bad the signal is. This entity is about whether the network signal is stable and whether the network is often disconnected.

“Receive power” is linked to “Signal quality” because it is influenced by signal. It reflects the received power of 5G signals. Generally, the better the signal quality, the higher the signal received power.

In “Signal strength” the most important feature is “lte\_rssi”. It gets Received Signal Strength Indication (RSSI) in dBm of the primary serving LTE cell.

One entity that will greatly influence signal strength is location. That is why we have an entity named “Location”. It uses latitude and longitude to record the specific location of the signal source.

“Mobile” and “Moving mode” are two entities directly linked to “Location”. “Mobile” records equipment’s movement. “compassDirection” is the horizontal direction of travel of this device, and is not related to the device orientation and “movingSpeed” indicates how fast it moves. “Moving mode” is some relevant but less important information about location and movement, like mobility\_mode (walking or driving) and “trajectory\_direction” (Clockwise or counterclockwise).

## **1.5 Data Warehouse (DW) Architecture**

---

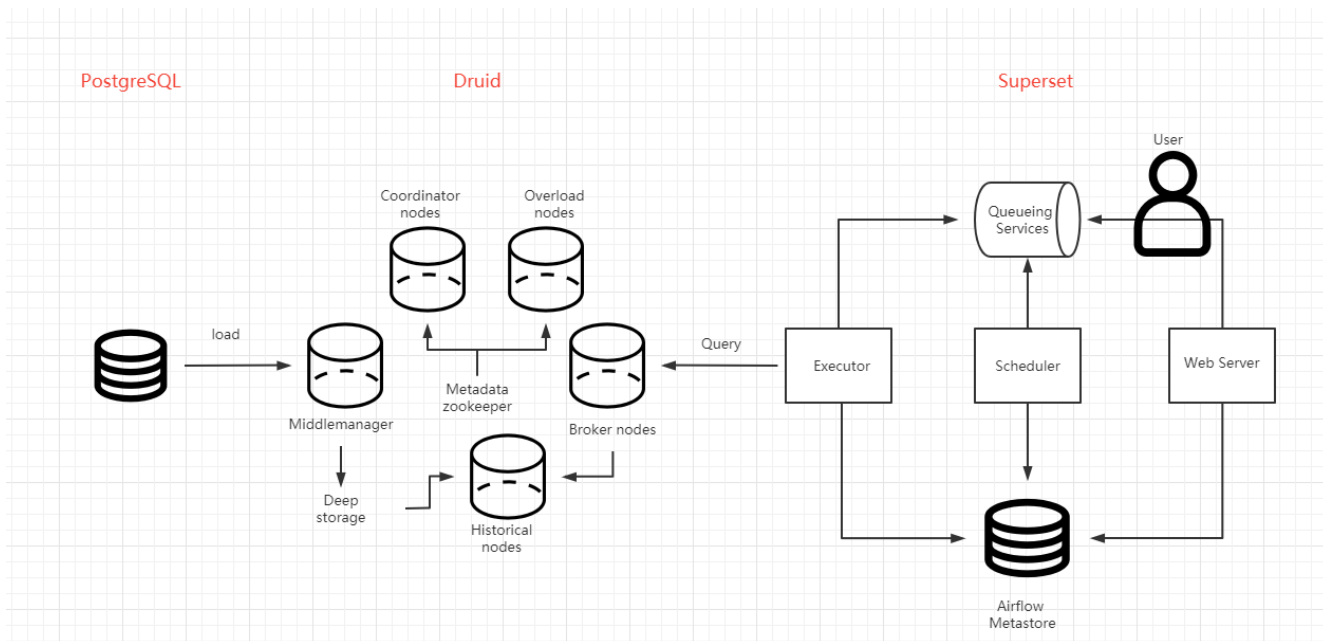


figure 1.5.1 Architecture

PostgreSQL is an object-relational database management system with complete features. It supports most SQL standards and provides many other modern features, such as complex query, foreign key, trigger, view, transaction integrity, multi version concurrency control, etc, to use PostgreSQL, you can download a graphical interface, such as pgadmin. PostgreSQL can be operated with SQL syntax.

Druid is a fast column distributed data storage system supporting real-time analysis. It has a significant performance improvement over the traditional OLAP System in processing Pb level data, millisecond level query and real-time data processing, it has a multi process, distributed architecture designed to be cloud friendly and easy to operate. Each Druid process type can be configured and expanded independently to provide maximum flexibility for your cluster. This design also provides enhanced fault tolerance: an interruption of one component does not immediately affect other components in order to use Druid, first we need to define the configuration file, then get the linked object and connect to the database.

Airflow is a programmable, scheduling and monitoring workflow platform. Based on directed acyclic graph (DAG), airflow can define a group of dependent tasks and execute them in turn according to dependencies. Airflow provides rich command-line tools for system control, and its web management interface can also facilitate the control and scheduling of tasks, and real-time monitor the operation status of tasks, which facilitates the operation, maintenance and management of the system. Generally speaking, we use celery worker to execute specific jobs. Workers can be deployed on multiple machines, and receive queues can be set separately. When there is a job task in the received queue, the worker will receive the job task and start execution. Airflow will automatically deploy a service logs service on each machine deploying workers at the same time, so that we can easily browse the job logs scattered on different machines on the web interface.

Superset is airbnb's open source big data visualization platform. It supports rich data source connections, a variety of visualization methods, and can realize fine-grained permission control for users. The main features of the tool are self-service analysis, custom dashboard,

visualization (export) of analysis results, user / role permission control, and an SQL editor is integrated for SQL editing and query.

## 1.6 Data Model Definition

---

*Develop and provide a brief description of the Druid Ingestion Spec (equivalent to DDL) to define your descriptive data analytics (DW) schema. Druid SQL to define the schema of your DW application*

截图 ingestion spec, 解释每张表格的含义, 解释表格再整个 DW 中的作用


*For 1.4 provide the Druid Ingestion Spec used to populate the database of your DW*

图换一下

Below is the table of “tower id” in our dataset, it records the id of each tower in the researching area. Different id represents different towers and this helps us to identify which tower the target received the signal from, therefore helping us to calculate the relationship between signal and distance. And it also contains the definition of each parameter in this entity, like the type of data, the name of each column. The type of “latitude” and “longitude” are double and the type of “tower\_id” is long. This can assist us to link each tower with an accurate location.

Here is our table about “4G” which is the comparison entity. We use this to see the difference between 5G and 4G. The type of each element in this entity is long.




Load data
Ingestion
Datasources
Segments
Services
Query

Connect and parse raw data

Transform data and configure schema

Tune parameters

Verify and submit

Start
Connect
Parse data
Parse time
Transform
Filter
Configure schema
Partition
Tune
Publish
Edit spec

```

{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMSHare/in/LumosSG-SG.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    }
  },
  "dataSource": {
    "dataSource": "LumosSG-SG",
    "timestampSpec": {
      "column": "!!!_no_such_column_!!!",
      "missingValue": "2010-01-01T00:00:00Z"
    },
    "dimensionsSpec": {
      "dimensions": [
        {
          "type": "long",
          "name": "run_num"
        },
        {
          "type": "long",
          "name": "seq_num"
        },
        {
          "name": "nrStatus"
        }
      ]
    },
    "granularitySpec": {
      "queryGranularity": "none",
      "rollup": false,
      "segmentGranularity": "day"
    }
  }
}

```

Connect and parse raw data

Transform data and configure schema

Tune parameters

Start

Connect

Parse data

Parse time

Transform

Filter

Configure schema

Partition

Tune

Publ

```

{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMShare/in/Lumos5G-4G.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    }
  },
  "dataSchema": {
    "dataSource": "Lumos5G-4G",
    "timestampSpec": {
      "column": "lte_rssnr",
      "format": "posix"
    },
    "dimensionsSpec": {
      "dimensions": [
        {
          "type": "long",
          "name": "run_num"
        },
        {
          "type": "long",
          "name": "seq_num"
        },
        {
          "type": "long",
          "name": "lte_rssi"
        },
        {
          "type": "long",
          "name": "lte_rsrp"
        },
        {
          "type": "long",
          "name": "lte_rsrq"
        }
      ]
    },
    "granularitySpec": {
      "queryGranularity": "none",
      "rollup": false
    }
  }
}

```

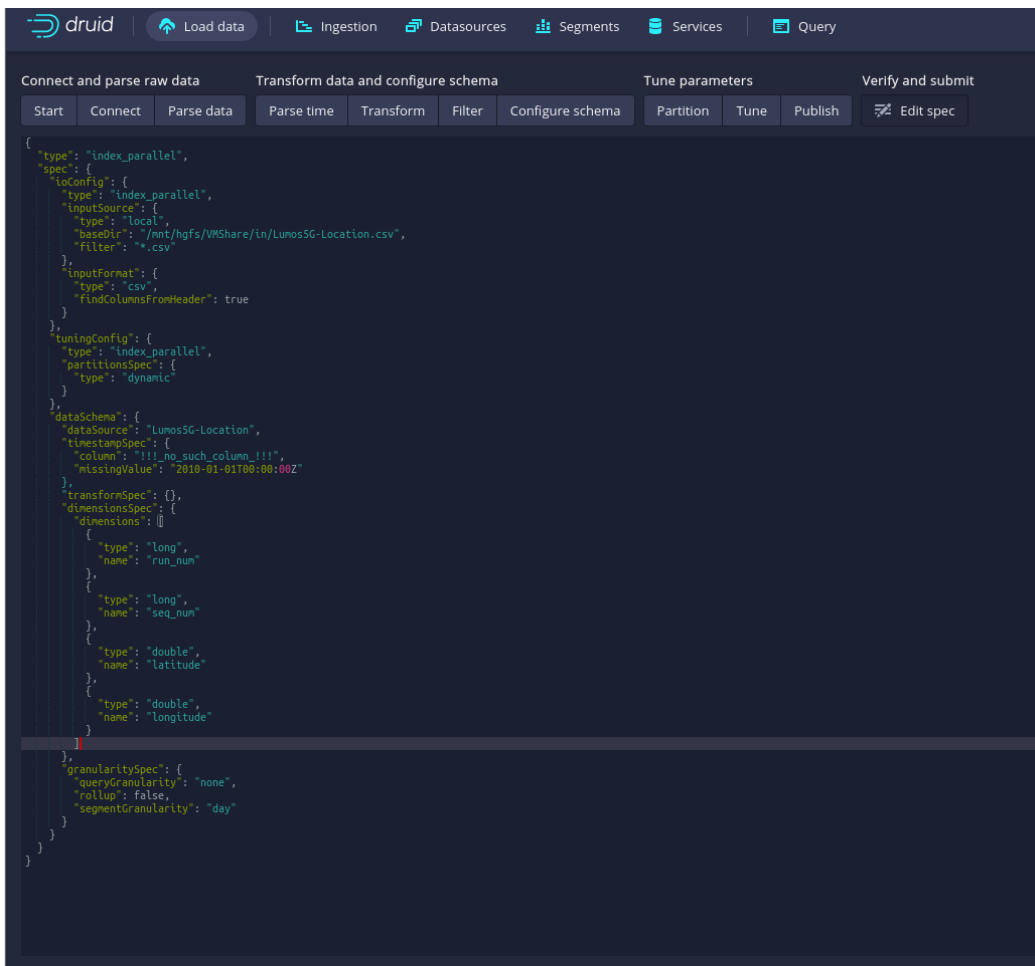
The next one is the “Throughput” table which records the throughput from our 5G data. It contains the id of each record, just like others, and it records the exact value of the throughput of each “5G” record. This is a very important table because it has directly relationship with the topic we want to research because throughput is a key feature of signal quality and we need it to see what relationship between it and the signal sources’ locations. The type for “latitude” and “longitude” are double, which is more accurate than long. The type for “throughput” is long.

The screenshot shows the Apache Druid web interface. The top navigation bar includes links for Load data, Ingestion, Datasources, Segments, Services, and Query. Below this, there are four main tabs: Connect and parse raw data, Transform data and configure schema, Tune parameters, and Verify and submit. The 'Transform data and configure schema' tab is active, showing a sequence of steps: Start, Connect, Parse data, Parse time, Transform, Filter, Configure schema, Partition, Tune, Publish, and Edit spec. The 'Configure schema' step is currently selected.

The main area displays a JSON configuration for a new table. A 'Software Updater' tooltip is visible over the 'parallel' field. The configuration is as follows:

```
{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMSHare/in/Lumos5G-throughput.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "parallel": "parallel",
    "partitionSpec": {
      "type": "dynamic"
    },
    "dataSchema": {
      "dataSource": "Lumos5G-throughput",
      "timestampSpec": {
        "column": "!!!no_such_column_!!!",
        "missingValue": "2010-01-01T00:00:00Z"
      },
      "dimensionsSpec": {
        "dimensions": [
          {
            "type": "long",
            "name": "run_num"
          },
          {
            "type": "long",
            "name": "seq_num"
          },
          {
            "type": "long",
            "name": "Throughput"
          }
        ]
      },
      "granularitySpec": {
        "queryGranularity": "none",
        "rollup": false,
        "segmentGranularity": "day"
      }
    }
  }
}
```

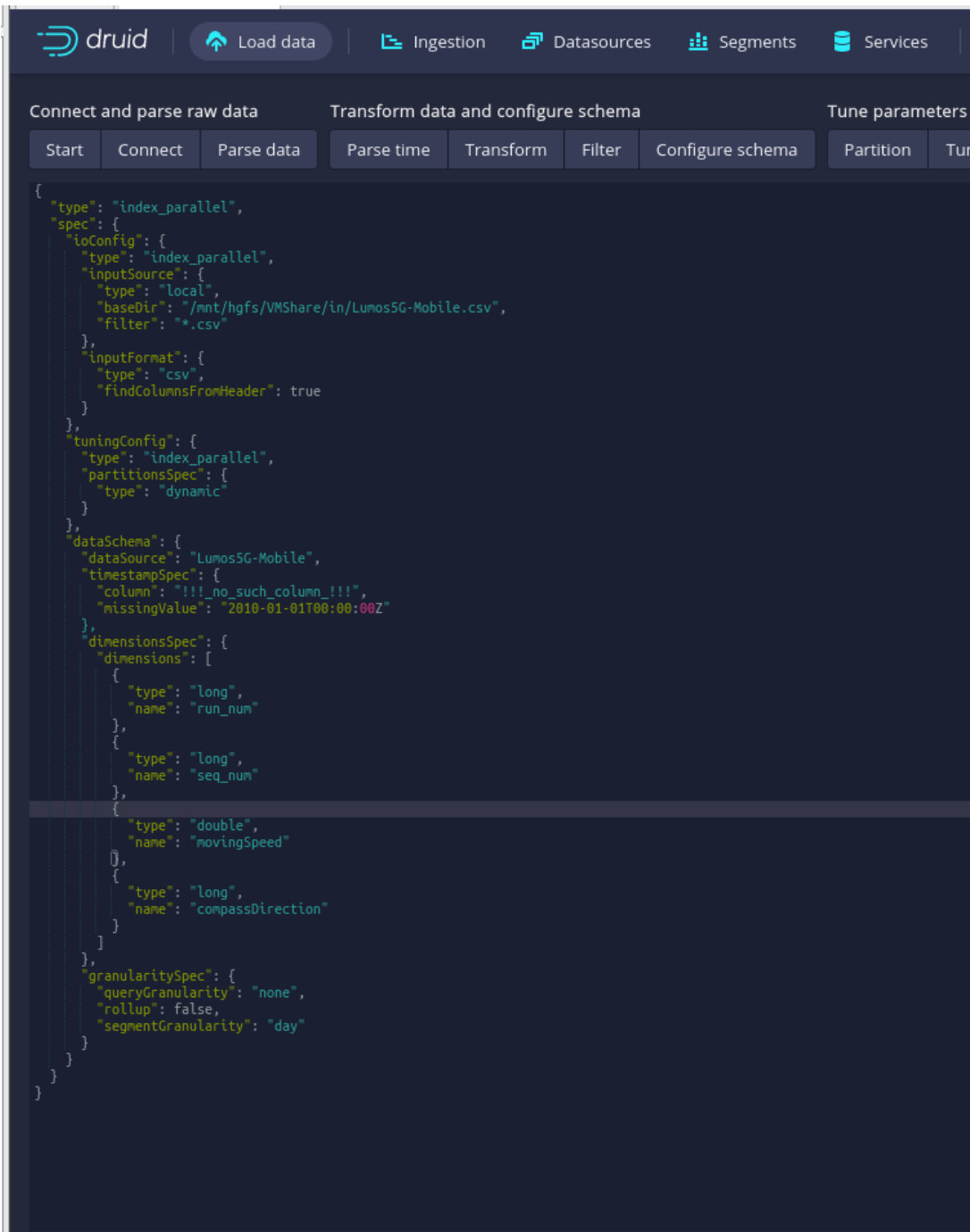
In “Location” table accurate latitude and longitude are included. It is another entity we will use in our research. So we assign type “double” for longitude and latitude.




The screenshot shows the Druid web console interface. At the top, there are navigation tabs: Load data, Ingestion, Datasources, Segments, Services, and Query. Below these, there are four main sections: Connect and parse raw data, Transform data and configure schema, Tune parameters, and Verify and submit. The 'Connect and parse raw data' section is active, showing a 'Start' button and a 'Connect' button. The 'Transform data and configure schema' section is also visible, showing buttons for Parse time, Transform, Filter, Configure schema, Partition, Tune, Publish, and Edit spec. The main area displays a JSON query specification for a CSV file named 'Lumos5G-Location.csv'.

```
{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMShare/In/Lumos5G-Location.csv",
        "filter": "*.csv"
      },
      "outputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    },
    "dataSchema": {
      "dataSource": "Lumos5G-Location",
      "timestampSpec": {
        "column": "!!!_no_such_column_!!!",
        "missingValue": "2010-01-01T00:00:00Z"
      },
      "transformSpec": {},
      "dimensionsSpec": {
        "dimensions": [
          {
            "type": "long",
            "name": "run_num"
          },
          {
            "type": "long",
            "name": "seq_num"
          },
          {
            "type": "double",
            "name": "latitude"
          },
          {
            "type": "double",
            "name": "longitude"
          }
        ]
      }
    },
    "granularitySpec": {
      "queryGranularity": "none",
      "rollup": false,
      "segmentGranularity": "day"
    }
  }
}
```

“Mobile” is the table we used to record the movement of data sources. The type of “movingSpeed” is double and the type for compass direction is long.



In table “ReceivedPower” not too many parameters are included. Except id it contains only one which is “nr\_ssRsrp”, this tells us the received power of each 5G record. We will use this with locations of signal resources in our research and the type of this data is long.


Load data
Ingestion
Datasources
Segments

Connect and parse raw data

Transform data and configure schema

Start

Connect

Parse data

Parse time

Transform

Filter

Configure schema

```

{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMShare/in/Lumos5G-ReceivedPower.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    },
    "schema": {
      "dataSource": "Lumos5G-ReceivedPower",
      "timestampSpec": {
        "column": "!!!_no_such_column_!!!",
        "missingValue": "2010-01-01T00:00:00Z"
      },
      "dimensionsSpec": {
        "dimensions": [
          {
            "type": "long",
            "name": "run_num"
          },
          {
            "type": "long",
            "name": "seq_num"
          },
          {
            "type": "long",
            "name": "nr_ssRsrp"
          }
        ]
      },
      "granularitySpec": {
        "queryGranularity": "none",
        "rollup": false,
        "segmentGranularity": "day"
      }
    }
  }
}

```

Settings

The type of “nr\_ssRsrp” and “nr\_ssRinr” is long, this two parameters represent signal received power and signal signal-to-noise ratio about our records of 5G signal. These two value decides how good or bad the signal is and we assign both of them with long data type.


Load data
Ingestion
Datasources

Connect and parse raw data

Transform data and configure schema

Start

Connect

Parse data

Parse time

Transform

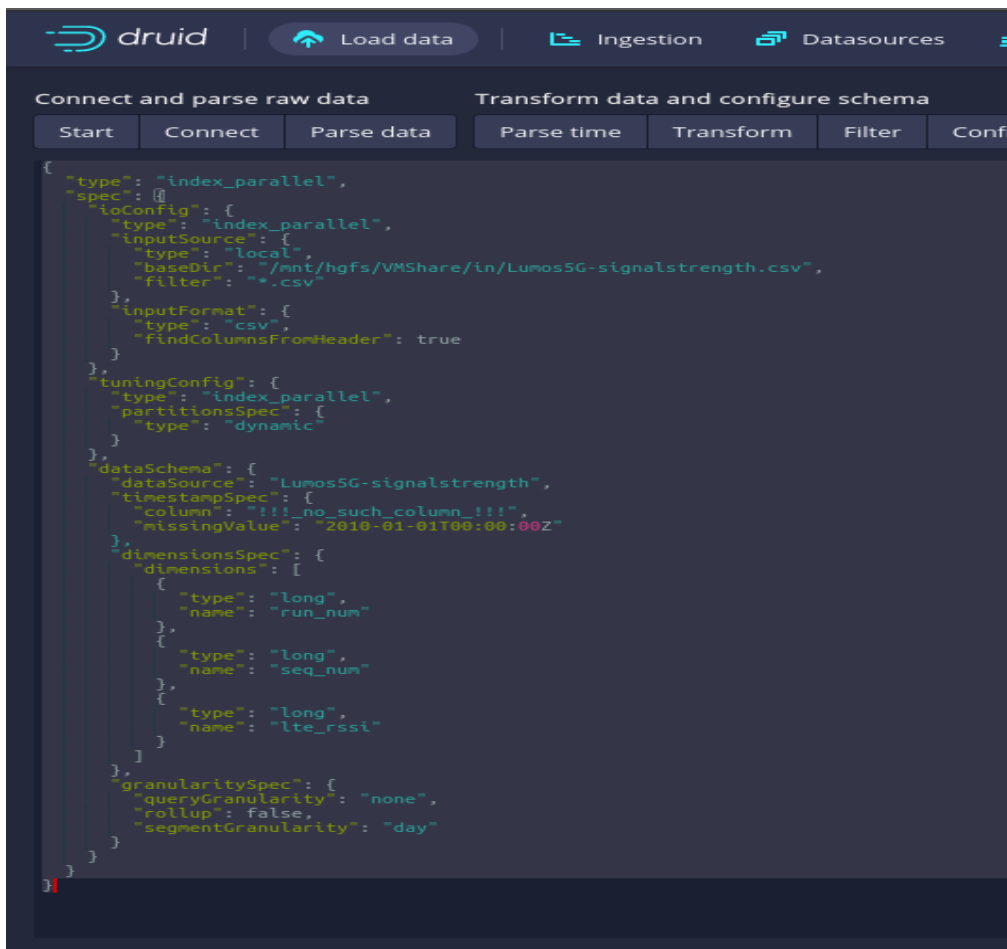
Filter

...

```

{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMShare/in/Lumos5G-Signalquality.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    },
    "dataSchema": {
      "dataSource": "Lumos5G-Signalquality",
      "timestampSpec": {
        "column": "!!!_no_such_column!!!",
        "missingValue": "2010-01-01T00:00:00Z"
      },
      "dimensionsSpec": {
        "dimensions": [
          {
            "type": "long",
            "name": "run_num"
          },
          {
            "type": "long",
            "name": "seq_num"
          },
          {
            "type": "long",
            "name": "nr_ssRsrq"
          },
          {
            "type": "long",
            "name": "nr_ssSnr"
          }
        ]
      },
      "granularitySpec": {
        "queryGranularity": "none",
        "rollup": false,
        "segmentGranularity": "day"
      }
    }
  }
}

```



Above is the table of signal strength which contains id and an important feature - "lte\_rssi". It records the value of Received Signal Strength Indication (RSSI) in dBm. The type of this parameter is long.

## 1.7 Data Warehouse Population (ETL)

*Provide brief description of how you used Airflow-based ETL to populate and maintain your DW and provide all supporting material.*

对应作业的 1.5 1.6

For 1.6 provide the ETL supported functionalities used to populate and maintain your DW

For 1.5 provide a description of the full ETL elements as implemented using Airflow.



Permission on Views/Menus

Tree Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code

DAG Docs

2021-12-01T22:02:07Z Runs 25 Update

DruidOperator PythonOperator

queued running success failed up\_for\_retry up\_for\_reschedule upstream\_failed skipped scheduled deferred no\_status

Auto-refresh

[DAG] extract transform load

Dec 01, 17:00

Tree Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code

DAG Docs

Settings 2021-12-01T22:00:32Z Runs 25 Run Layout Left > Right Update No DAG runs yet. Find Task...

DruidOperator PythonOperator

queues running success failed up\_for\_retry up\_for\_reschedule upstream\_failed skipped scheduled deferred no\_status

Auto-refresh

extract → transform → load

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions
9014dag day9014	airflow	1	None	2021-12-01, 22:02:07		3	

## 1.8 Build and Materialize Cubes

*Provide a brief description of applying Druid SQL to build and materialize your selected Cubes (as needed) and justify your answer. Provide samples*

For 1.7 provide Druid SQL statements to build the materialized Cubes in your DW

Add lookup

Name

movingmethod

Tier

\_\_default

Version

2021-11-29T00:11:58.843Z

Set to current ISO time

Form

JSON

```
{
  "type": "map",
  "map": {
    "1": "driving",
    "2": "driving",
    "3": "driving",
    "4": "walking",
    "5": "driving",
    "6": "driving",
    "7": "walking",
    "8": "walking",
    "9": "walking",
    "10": "walking",
    "11": "walking",
    "12": "walking",
    "13": "walking",
    "14": "driving",
    "15": "walking",
    "16": "walking",
    "17": "walking",
    "18": "walking",
    "19": "walking",
    "20": "walking",
    "21": "driving",
    "22": "driving"
  }
}
```

Close

Submit



Key	Value
1	driving
2	driving
3	driving
4	walking
5	driving
6	driving
7	walking
8	walking
9	walking
10	walking
11	walking
12	walking
13	walking
14	driving
15	walking

Previous Page 1 of 6 20 rows Next

Actions Close

Key	Value
1	CW
2	CW
3	CW
4	ACW
5	CW
6	CW
7	ACW
8	CW
9	CW
10	ACW
11	CW
12	ACW
13	ACW
14	CW
15	CW

Previous Page 1 of 6 20 rows Next

Actions Close

1	SELECT *
2	FROM Lookup.movingmethod

▶ Run
...
Auto limit
Live query: Auto
100+ results in 0.01s

k	v
1	driving
2	driving
3	driving
4	walking
5	driving
6	driving
7	walking
8	walking
9	walking
10	walking
11	walking
12	walking

1	SELECT *
2	FROM lookup.novdir

▶ Run
...
Auto limit
Live query: Auto
100+ results in 0.02s

k	v
1	CW
2	CW
3	CW
4	ACW
5	CW
6	CW
7	ACW
8	CW
9	CW
10	ACW
11	CW
12	ACW

## 1.9 Data Manipulation: Internal Schema

*Provide samples and description of the required answers or/and reports for your descriptive data analytics problem generated using Druid SQL and Superset as OLAP with visualization*

For 1.8 provide the Druid SQL statements and corresponding copies of the visualization produced by Superset representing the reports for your descriptive data analytics problem

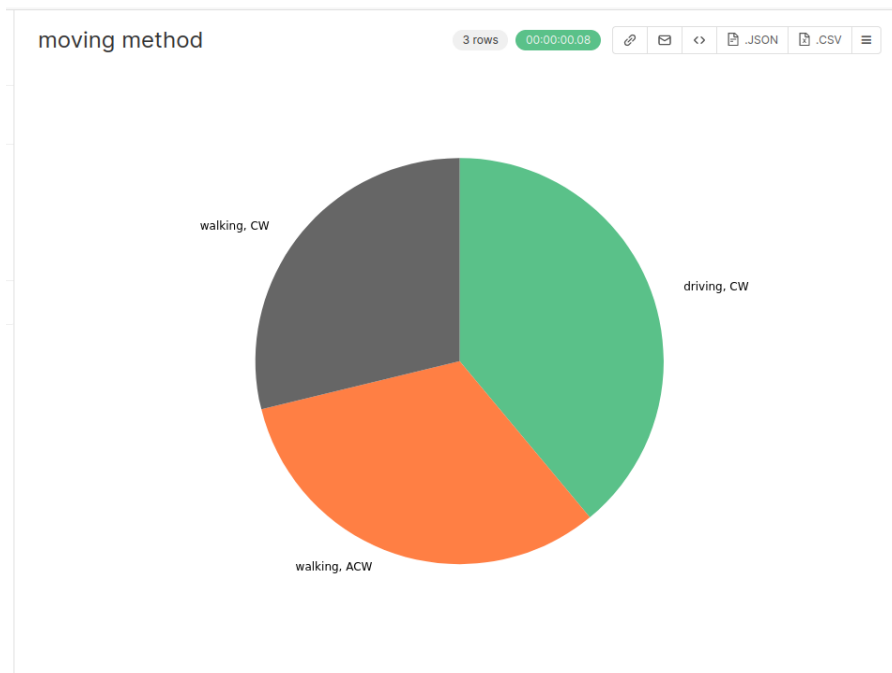
```
SELECT "__time",
       "mobility_mode",
       "run_num",
       "trajectory_direction"
FROM "druid"."Lumos5G-MovingMode"
```

The screenshot displays the Apache Druid web console interface. The top navigation bar includes links for 'Load data', 'Ingestion', 'Datasources', 'Segments', 'Services', and 'Query'. The main content area is divided into two sections: 'Supervisors' and 'Tasks'.

The 'Supervisors' section shows a table with columns: Datasource, Type, Topic/Stream, Status, and Actions. It currently displays 'No supervisors'.

The 'Tasks' section shows a table with columns: Task ID, Group ID, Type, Datasource, Location, Created time, Status, Duration, and Actions. It displays two tasks:

Task ID	Group ID	Type	Datasource	Location	Created time	Status	Duration	Actions
index_parallel_Lumos5G-MovingMode_hhdhghd_2021-12-01T21:01:46.422Z	index_parallel_Lumos5G-MovingMode_hhdhghd...	index_parallel	Lumos5G-M...	localhost:8100	2021-12-01T21:01:46.431Z	RUNNING		
index_parallel_Lumos5G-throughput_fmcbaj_2021-12-01T20:58:51.456Z	index_parallel_Lumos5G-throughput_fmcbaj...	index_parallel	Lumos5G-th...	localhost:8100	2021-12-01T20:58:51.467Z	SUCCESS	0:00:14	



```
SELECT "Throughput",
      "__time",
      "run_num",
      "seq_num"
FROM "druid"."Lumos5G-thoughtput"
```

druid

Load data Ingestion Datasources Segments Services Query

### Supervisors

Refresh Columns (5/5)

Datasource	Type	Topic/Stream	Status	Actions
No supervisors				

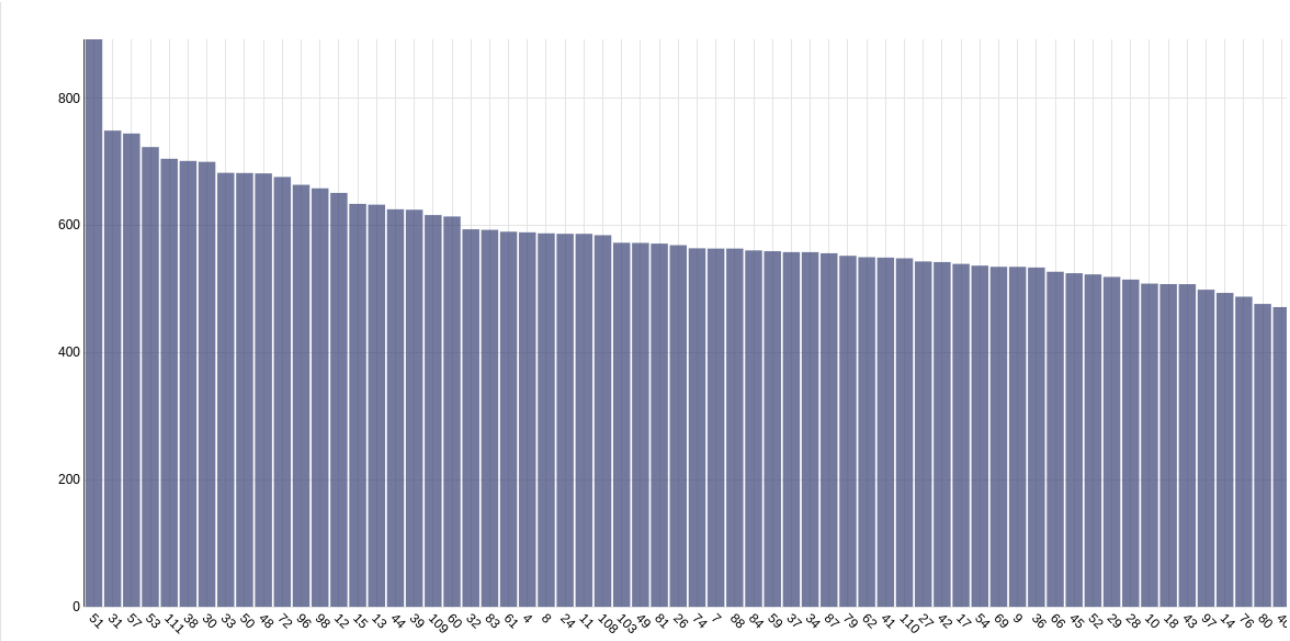
Software Updater Previous Page 1 of 1 20 rows Next

### Tasks

Group by None Group ID Type Datasource Status Refresh Columns (9/9)

Task ID	Group ID	Type	Datasource	Location	Created time	Status	Duration	Active
index_parallel_Lumos5G-thoughtput_fm...	index_parallel_Lumos5G-thoughtput_fmbr							
index_parallel_Lumos5G-thoughtput_fm...	index_parallel_Lumos5G-thoughtput_fm...	index_parallel	Lumos5G-th...	localhost:8100	2021-12-01T20:58:51.569Z	RUNNING		

Previous Page 1 of 1 20 rows Next





---

## 2 Individual Deliverables

*Opening paragraph describing the overall extensions (with one demission per extension) and the list of the individual names responsible for each extension*

---

### 2.1 Extension 1: throughput of moving signal

---

#### 2.1.1 Main Data Analytics Elements/Concepts

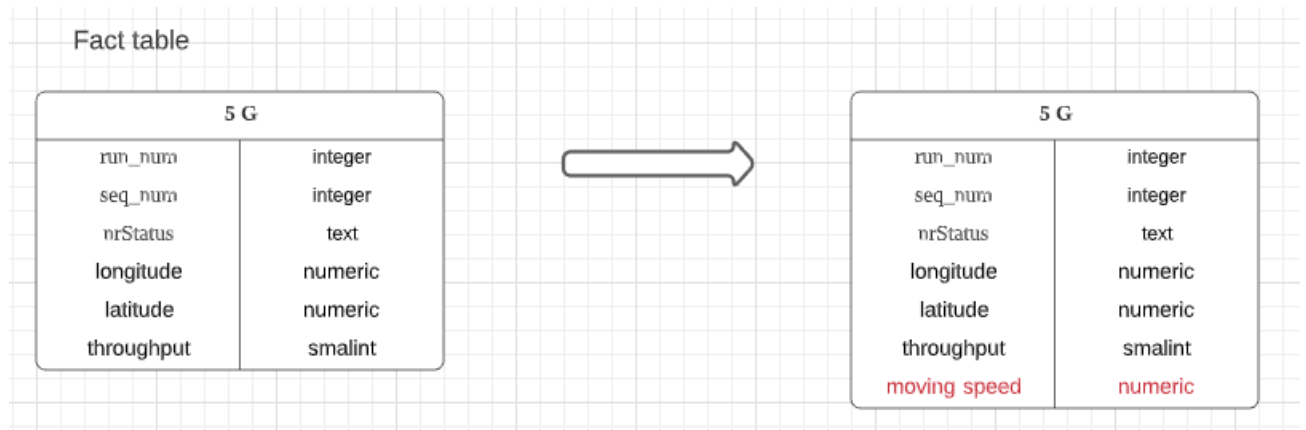
*Provide a brief description of the proposed expansion to your group descriptive data analytics with one analytical (decision) variables of interest that will enable within your application domain problem to produce high rich “information”, which in turn can be a base to identify insights about the context of the application domain. It can be based on the extension of your entities and the associated relations in “Deliverable 1”*

The proposed expansion is “throughput with respect to the moving speed. The one analytical variable will be the moving signal. The moving signal is combined with the moving speed, ite\_rsrq(signal quality), ite\_rsrp(signal power). With the help of this analytical variable, we can know the change of the throughput when the moving speed changes as well.

#### 2.1.2 Data Model Definition

*Develop and provide a brief description of the Druid Ingestion Spec (equivalent to DDL to define your new Dimension/Fact table(s) for the group’s descriptive data analytics (DW) schema*

Uploading all data needed to the druid, Druid Ingestion Spec will make those data as the datasource.



```
{
  "type": "inline",
  "data": "run_num,seq_num,nrStatus,movingSpeed,lte_rsrp,lte_rsrq,longitude,latitude,Throughput\\n1,1,NOT_RESTRICTED",
},
"inputFormat": {
  "type": "csv",
  "findColumnsFromHeader": true
},
"appendToExisting": false
},
},
"tuningConfig": {
  "type": "index_parallel",
  "partitionsSpec": {
    "type": "dynamic"
  }
},
"dataSource": "5G-extension 1",
"timestampSpec": {
  "column": "!!!_no_such_column!!!",
  "missingValue": "2010-01-01T00:00:00Z"
},
"dimensionsSpec": {
  "dimensions": [
    {
      "type": "double",
      "name": "latitude"
    },
    {
      "type": "double",
      "name": "longitude"
    },
    {
      "type": "long",
      "name": "lte_rsrp"
    },
    {
      "type": "long",
      "name": "lte_rsrq"
    },
    {
      "type": "double",
      "name": "movingSpeed"
    },
    "nrStatus",
    {
      "type": "long",
      "name": "run_num"
    },
    {
      "type": "long",
      "name": "seq_num"
    },
    {
      "type": "long",
      "name": "Throughput"
    }
  ]
}
```

### 2.1.3 Data Warehouse Population (ETL)

*Provide brief description of how you used Airflow-based ETL to populate and maintain [your new Dimension/Fact table\(s\)](#).*

The ETL function is combined with three functions: `extract_task`, `transform_task` and `load_task`.

`extract_task`: data will be accessed at PostgreSQL and then saved as a file to be found by the `transform_task`. So the extension data will be created at PostgreSQL.

```
extract_task = PythonOperator(
    task_id='extract',
    python_callable=extract,
    op_kwargs = {
        "sql": "extract_query",
        "postgres_conn_id": "pgoltp",
        "pandas_sql_params": None,
        "csv_path": '/tmp/filesystem/extract.csv',
        "csv_sep": ",",
    }
)
extract_task.doc_md = dedent(
    """\
```

`transform_task`: in this task, the data file extracted from the last step will be performed and transformed. The transformed will be ready for the next step-`load_task`.

```
transform_task = PythonOperator(
    task_id='transform',
    python_callable=transform,
    op_kwargs = {
        "csv_path": '/tmp/filesystem/transformed.csv',
        "csv_sep": ",",
    }
)
transform_task.doc_md = dedent(
    """\
```

`load_task`: Finally, the transformed data will be loaded to the druid as the datasource.

```
load_task = DruidOperator(task_id='load', json_index_file='json_index.json', druid_ingest_conn_id='druid_ingest_default')
```

## 2.1.4 Build and Materialize Cubes

*Provide a brief description of applying Druid SQL to build and metalize [your additional Cubes](#) (as needed) and justify your answer.*





```

1  -- Note: Unless you save your query, these tabs will NOT pers
2
3  select "Throughput","movingSpeed","lte_rsrq","lte_rsrp"
4  from "5G-extension 1"
5

```

**RUN**    LIMIT: 1 000    00:00:00.18

RESULTS    QUERY HISTORY    PREVIEW: `5G`    PREVIEW: `MOVING SIGNAL`

 **EXPLORE**   
  **DOWNLOAD TO CSV**   
  **COPY TO CLIPBOARD**   
  **Fi**

19 rows returned

Throughput	movingSpeed	lte_rsrq	lte_rsrp
116	7.6398712	-16	-95
4	7.5100057	-16	-95
13	7.3809295	-16	-95
75	7.2591074	-16	-95
110	7.3278816	-16	-95
4	7.48102235	-16	-95
8	6.68715865	-16	-95
124	5.6157547	-16	-95
86	4.45283425	-16	-95
116	3.67905355	-16	-95
87	3.97776975	-16	-95

## 2.1.5 Data Manipulation: Internal Schema

*Provide samples and description of the [your new answers or/and reports](#) for your descriptive data analytics problem generated using Druid SQL and Superset as OLAP with visualization*

Using the superset to make a table about the moving speed and throughput data. We can see the relation between moving speed and the throughput from the chart.

Settings

- untitled

19 rows 00:00:00.17

Search 19 records...

movingSpeed	Throughput
0.094889398	78
0.87663361	117
2.22579805	110
3.1807259	108
3.67905355	116
3.7512345	100
3.97776975	87
4.0833311	124
4.10702345	108
4.13076805	109
4.45283425	86
5.6157547	124
6.68715865	8
7.2591074	75
7.3278816	110
7.3809295	13
7.48102235	4
7.5100057	4
7.6398712	116

## 2.1.6 Predictive Data Analytics problem

*Provide a brief description of a predictive data analytics problem based on the Application domain of your chosen dataset to produce insights about the context of the application domain in the form of*

- either producing predictions based on new instances of the application domain*
- or discovering hidden properties related to the application domain that can a base to identify new insights*

## 2.1.7 Predictive Data Model

*Develop and provide a description of an adequate data model capturing the data elements required for your predictive data analytics*

## 2.1.8 Predictive Data Analytics Solution Pipeline

*Develop and provide a description of the data predictive pipeline (using IBM Cloud Pak for Data and Spark ML) required to solve your predictive data analytics problem described above. It should be integrated with*

*your data warehouse system developed for Deliverable 2 of the project as the data source. Provide a brief description of each component and its usage*

#### **2.1.8.1 Samples of the Predictive Data Analytics Solutions**

*Provide samples of the desirable answers of your predictive data analytics problem described above*

#### **2.1.8.2 Predictive Data Analytics Solution: Accuracy & Performance**

*Provide your analysis of the produced analysis in terms of accuracy and performance.*

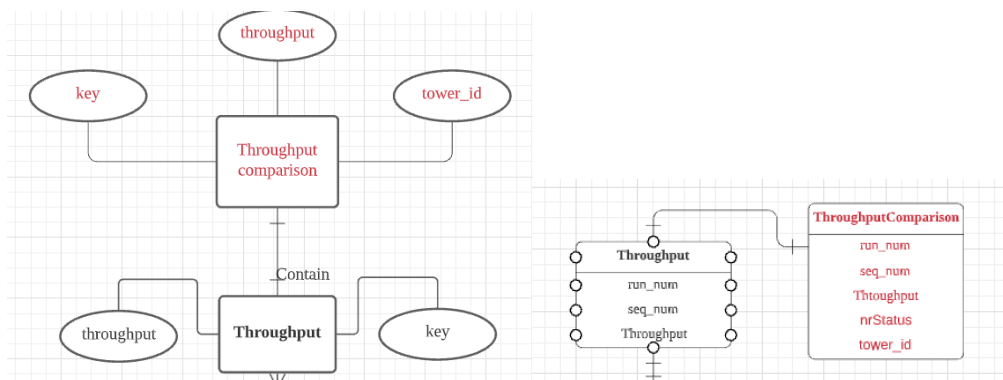
### 2.2.1 Main Data Analytics Elements/Concepts

In this extension I want to study the throughput difference between ‘connected to 5G’ and ‘connected to 4G’ in terms that the sources all accept signal from the same signal tower. By doing this research I want to show users the advantages 5G signal has over 4G because the value of ‘throughput’ can directly show the actual signal transmission rate. This comparison will help users to decide whether it is worthwhile to use 5G.

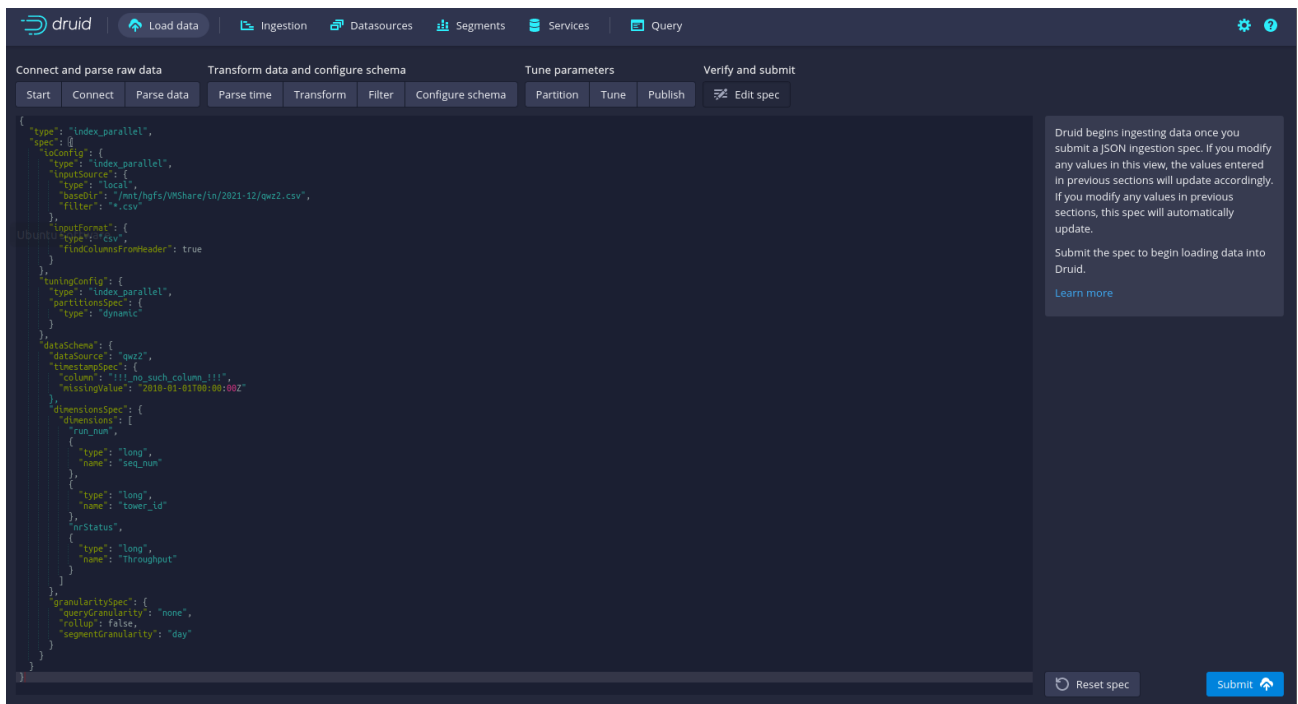
The main data elements I need to use are ‘run\_num’, ‘seq\_num’, ‘tower\_id’, ‘nrStatus’ and ‘Throughput’. The combination of ‘run\_num’ and ‘seq\_num’ is the key value I need to identify each record as before. By using ‘tower\_id’ I want to make sure that all the records I use are from the same tower, in case the results are influenced by ability of different signal tower. The key value of ‘Throughput’ will help me to decide how good or bad the signal received by the record source is. ‘Throughput’ is also the value I want to compare this this extension.

### 2.2.2 Data Model Definition

Here is the conceptual and logical model of my extension.



Here is the background of my extend entity in druid. It provides detailed information about the data in my entity. I load data from a csv file I created in my first project. We can see that the type of two important variables is ‘long’.



Then in another window druid shows me that there are total 5 columns in my entity.

[illegible]

### 2.2.3 Data Warehouse Population (ETL)

Here are the ETL codes I used in my extension.

Extract the original data from the data source.



```

extract_task = PythonOperator(
    task_id='extract',
    python_callable=extract,
    op_kwargs = {
        "sql": "extract_query",
        "postgres_conn_id": "pgoltp",
        "pandas_sql_params": None,
        "csv_path": '/tmp/filesystem/extract.csv',
        "csv_sep": ",",
    }
)

```

Then transform extracted data.

```

transform_task = PythonOperator(
    task_id='transform',
    python_callable=transform,
    op_kwargs = {
        "csv_path": '/tmp/filesystem/transformed.csv',
        "csv_sep": ",",
    }
)

```

Finally load the data to the data warehouse.

```

load_task = DruidOperator(task_id='load', json_index_file='json_index.json', druid_ingest_conn_id='druid_ingest_default')

```

To maintain my data warehouse it requires the cooperation of Druid, Ariflow and Superset.

Apache Airflow is to manage increasingly complex data management tools, scripts, and analysis tools, and provides a solution for building batch workflows. From a functional point of view, this is a scalable distributed workflow scheduling system that allows workflows to be modeled as directed acyclic graphs (DAGs), which simplifies the creation and orchestration of various processing steps in the data pipeline And monitoring.

My apache druid is running one the localhost:8888 and I just add the data I need to the druid, but it is hard to use the data in druid. So I need to connect apache druid and airflow. By fill the information of my druid in airflow they can be successfully connected.

DAGs
Security
Browse
Admin
Docs

Changed Row

List Connection

Search

+

Actions

-

	Conn Id	Conn Type	Description	Host	Port	Is Encrypted	Is Extra Encrypted
<input type="checkbox"/>	druid	druid		192.168.0.22	8888	False	False

Airflow's WebUI is the embodiment of its task scheduling visualization, and you can monitor the real-time and historical data of almost all task scheduling operations on this WebUI.

DAG: 9014dag

Schedule: None

Next Run: None

Tree

Graph

Calendar

Task Duration

Task Tries

Landing Times

Gantt

Details

<> Code

DAG Docs

2021-12-14T20:43:07Z

Runs

25

Update

DruidOperator

PythonOperator

queued

running

success

failed

up\_for\_retry

up\_for\_reschedule

upstream\_failed

skipped

scheduled

deferred

no\_status

[DAG]

extract

transform

load

Dec 01, 17:04

Dec 14, 15:4

Auto-refresh

Refresh

DruidOperator

PythonOperator

queued

running

success

failed

up\_for\_retry

up\_for\_reschedule

upstream\_failed

skipped

scheduled

deferred

no\_status

Auto-refresh

Refresh

extract

transform

load

## 2.2.4 Build and Materialize Cubes

I have added data of entity to Druid and the task is successfully running. So, I want to check if the data can be used or see by users in druid. I type the SQL in Query to make sure the data is available.

```
SELECT "Throughput", "nrStatus", "seq_num"
from "inline_data"
```

I use this command to check the value of three parameters ‘Throughput’, ‘nrStatus’ and ‘seq\_num’.

```
1 SELECT "Throughput", "nrStatus", "seq_num"
2 from "inline_data"
```

Run Auto limit Live query: Auto

Throughput	nrStatus	seq_num
1150	CONNECTED	386
1020	CONNECTED	389
110	CONNECTED	421
1100	NOT_RESTRICTED	385
98	NOT_RESTRICTED	420

And I also run this in Superset. Use some SQL in superset to select elements.

```
1 SELECT "Throughput", "nrStatus", "tower_id"
2 FROM "druid"."inline_data"
3 LIMIT 100
```

RUN LIMIT: 1 000 00:00:00.22

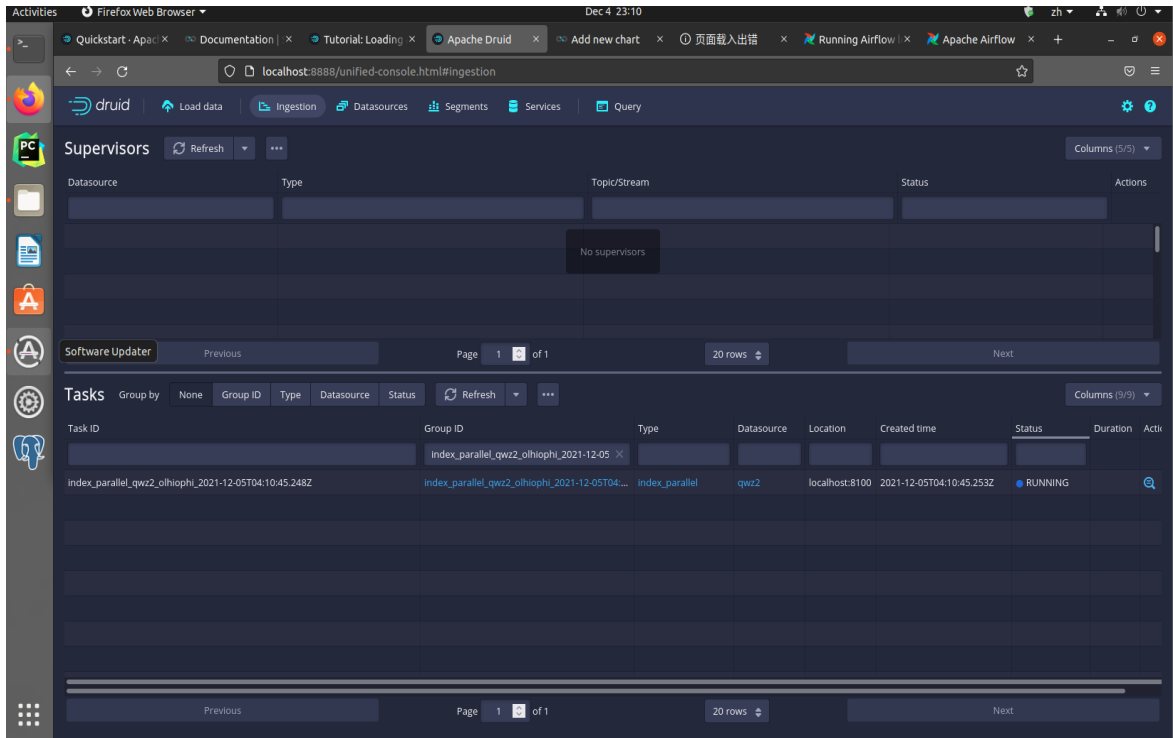
RESULTS QUERY HISTORY

EXPLORE DOWNLOAD TO CSV COPY TO CLIPBOARD Filter

27 rows returned

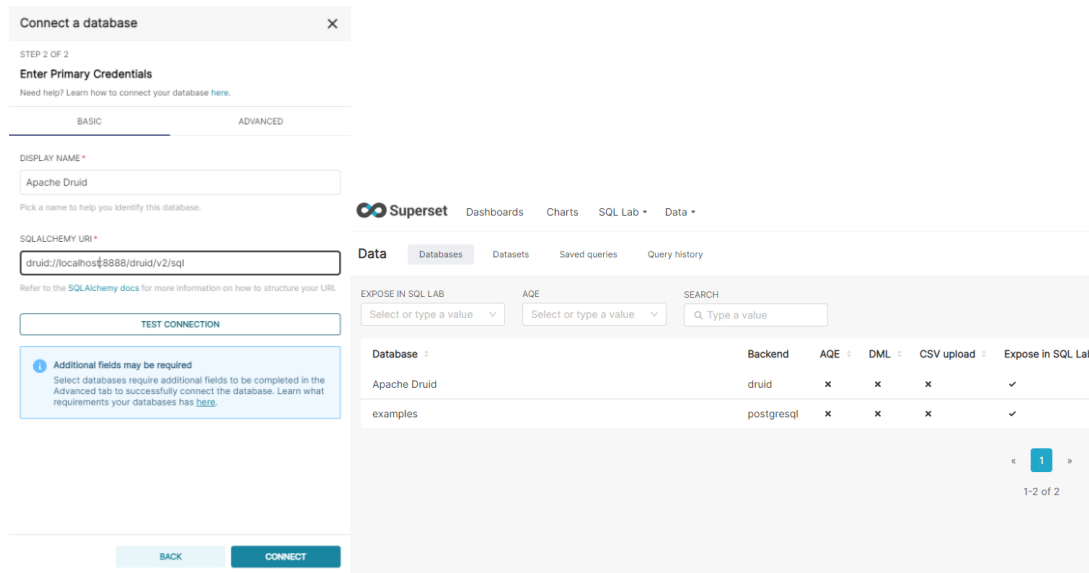
Throughput	nrStatus	tower_id
549	CONNECTED	17
559	CONNECTED	17
575	CONNECTED	17
568	CONNECTED	17
538	CONNECTED	17
398	CONNECTED	17
111	CONNECTED	17

## 2.2.5 Data Manipulation: Internal Schema



Firstly, make sure I have loaded the csv file to druid. As shown above, the task is running.

Then connect 'Superset' and 'Apache Druid'. This operation is to make sure that Superset can reach to the data I saved in druid. To do this I need to add Druid as a database to my Superset.

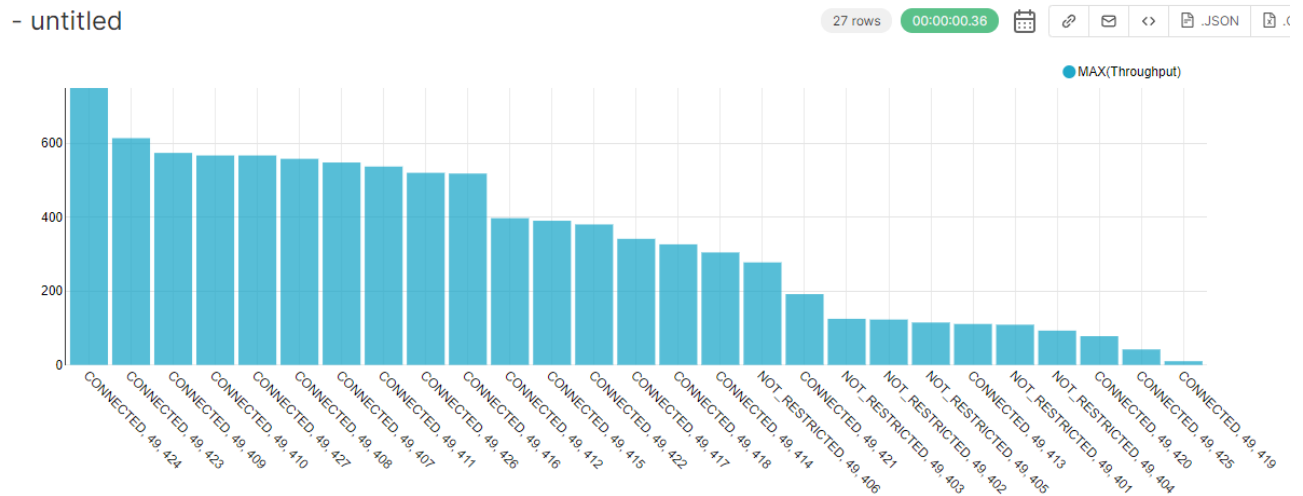


Here I have successfully connected Superset and Druid. Then I need to add the data set I created in Druid into Superset so it can use the value in that dataset.



VIEW RESULTS	VIEW SAMPLES	27 rows retrieved	
nrStatus	run_num	seq_num	MAX(Throughput)
CONNECTED	49	424	751
CONNECTED	49	423	615
CONNECTED	49	409	575
CONNECTED	49	410	568
CONNECTED	49	427	568
CONNECTED	49	408	559
CONNECTED	49	407	549

The most important function of Superset is to visualize data. Here I set up a coordinate system to display the Bar Chart.



The abscissa is the key and connection status of each record. The ordinate is the value of throughput each record.

Here we can clearly see that when the signal source is connected to 5G, that is, nrStatus is CONNECTED, the signal throughput is very large, and the number will basically remain above three hundred. In contrast, if the signal source is connected to 4G, which means nrStatus is NOT\_RESTRICTED, its throughput is very small compared to 5G signals. The recording error and the influence of different signal towers are excluded here. Because their run\_num and tower\_id is the same, it means that they are all data from the same tower in the same data record.

So, my answer to this data analytics problem is: Compared with 4G signal, the throughput of 5G signal is greatly improved. And it is very worthy to choose 5G.

I think this research is very convincing. First of all, the improvement in throughput of 5G signals compared to 4G signals is very large (about 3 times). And the throughput records the actual transmission rate of the signal rather than the theoretical one, which means that even under the influence of external factors, such as inclement weather and equipment quality, the transmission rate of 5G signals has been greatly improved.

## 2.2.6 Predictive Data Analytics problem

---

In my extension, I studied the relationship between throughput and signal connection status, and I found that the throughput of 5G signals is much greater than that of 4G signals. In my previous research, I concluded that 5G signals are greatly improved compared to 4G signals, and it is very worthwhile for users to choose to use 5G signals.

What I want to study next is whether the throughput of a signal source in an area is related to its moving angle in the area. Different moving directions meet the signal tower to form different angles. I want to know whether this angle will have an impact on throughput. I want to know if the moving angle will influence the accuracy of the record.

## 2.2.7 Predictive Data Model

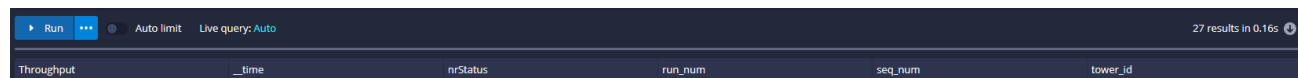
I chose to use decision tree as a predictive model. Because I need to compare different numbers of 'Throughput' and 'compassDirection'. If it moving direction does have an effect on throughput, the predicted result will be very accurate. Otherwise the prediction number will be far from real ones.

The decision tree is composed of nodes and directed edges. There are two types of nodes: internal nodes and leaf nodes. The internal node represents a feature or attribute, and the leaf node represents a category or a certain value. When using a decision tree for classification or regression tasks, start from the root node, test a certain feature of the sample, and assign the sample to its child nodes according to the test results; meanwhile, each child node corresponds to one of the features values. Use "mean squared error (MSE)" as a criterion for feature selection. In this way, the samples are tested and distributed recursively until the leaf node is reached. In fact, the decision tree is a method of dividing the space with hyperplanes. Each time it is divided, the current space is divided according to the value of the feature, so that each leaf node is a different part of the space. When making a decision, the intersecting area will be based on the value of each dimension of the input sample, step by step, and finally make the sample fall into one of the N areas (assuming there are N leaf nodes).

## 2.2.8 Predictive Data Analytics Solution Pipeline

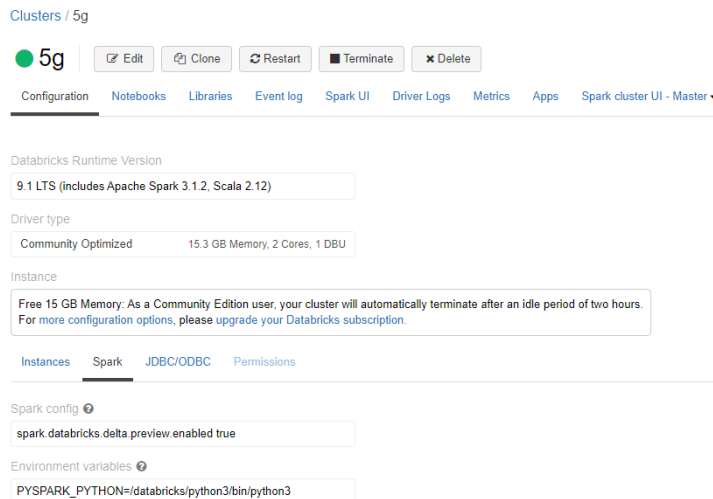
*Develop and provide a description of the data predictive pipeline (using IBM Cloud Pak for Data and Spark ML) required to solve your predictive data analytics problem described above. It should be integrated with your data warehouse system developed for Deliverable 2 of the project as the data source. Provide a brief description of each component and its usage*

To get the answer of my problem I need to export my dataset in druid. Download this as a csv file.

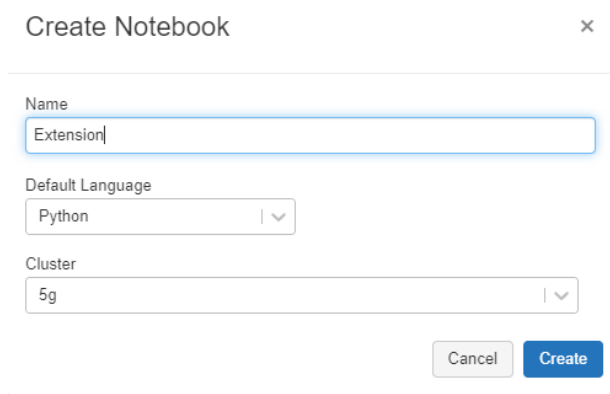


Throughput	_time	nrStatus	run_num	seq_num	tower_id

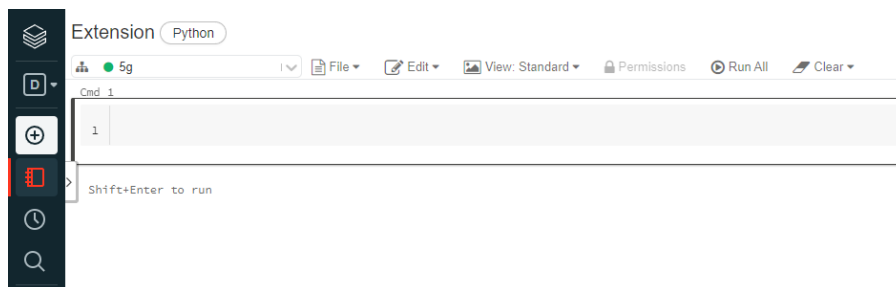
Then import this csv file to databricks. This csv file will act as a data source. Create a 'Cluster' by using this file and I name it '5g'.



To do operation to this cluster I need a workspace to type code, which in databricks is 'Notebook'. I created a notebook called 'extension' and attached it to my '5g' cluster. About programming language, I choose Python.



After creating notebook, here I have a workspace to do programming. This is very similar to 'Google cloud' I used before. This platform allows me to program and execute code on cloud rather than on local cpu. Its running space is much faster than cpu and that will save me a lot of time when I deal with a large amount of data. Besides, it won't occupy too much memory of my laptop.



In the white space I can type code to analysis my data or do some prediction which is what I want to do, as long as my code is in right form.

Firstly, import the modular I need to create an Apache Spark machine learning model.



```

from pyspark.sql import SparkSession
from pyspark.ml.regression import DecisionTreeRegressor
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.feature import VectorIndexer
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml import Pipeline
from pyspark.ml.tuning import ParamGridBuilder, CrossValidator

spark = SparkSession.builder.master('local').appName('DecisionTreeRegression').getOrCreate()

```

The data I need to use are 'Throughput' and 'compassDeriction'. Set an array to save these two features.

```

data1 = data.select(
    data["compassDirection"].cast("Double"),
    data["Throughput"].cast("Double"), |

featuresArray = ["compassDirection", "Throughput"]
assembler = VectorAssembler().setInputCols(featuresArray).setOutputCol("features")

```

Randomly divide the data, this data is used in the regression model.

```

trainingData, testData = data1.randomSplit([0.7, 0.3])

```

The next steps are very important steps. First, the feature conversion involved in the decision tree regression model and the model training group are loaded with a pipeline. Then train the decision tree regression model to predict the value of the decision tree regression model. Finally, the prediction result is evaluated. Here I use r2 to evaluate the model.

```

model = pipeline.fit(trainingData)
predictions = model.transform(testData)
evaluator = RegressionEvaluator().setLabelCol("label").setPredictionCol("prediction").setMetricName("r2")
score = evaluator.evaluate(predictions)
print("score:", score)

```

### 2.2.8.1 Samples of the Predictive Data Analytics Solutions

To show the result I use some code to draw a graph.

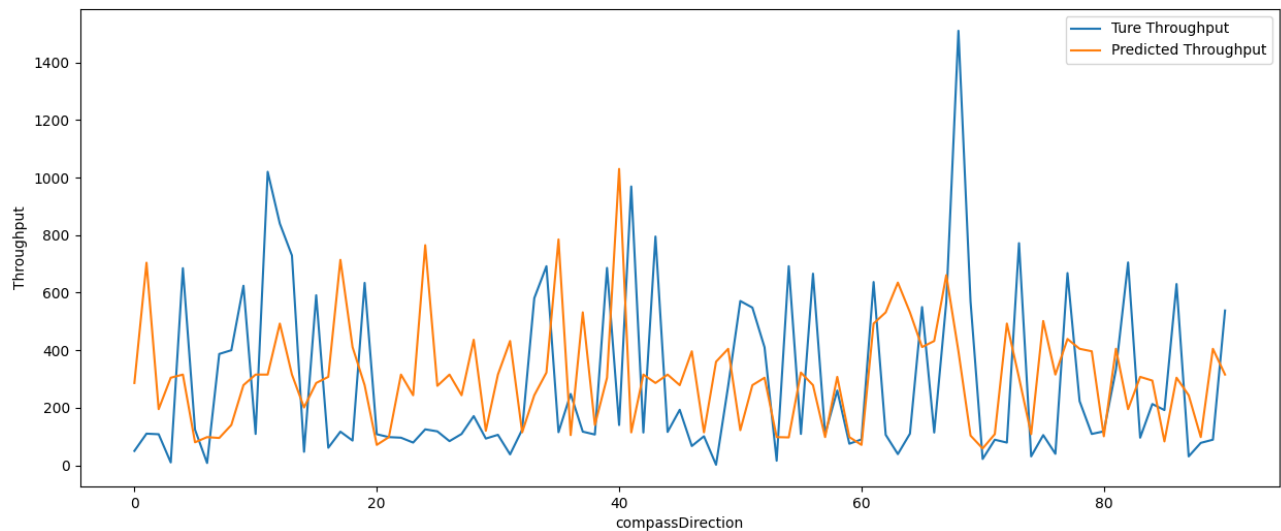
```

plt.figure(figsize=(15,6))
plt.xlabel(u'time')
plt.ylabel(u'throughput')
plt.title('score: %f'%score)

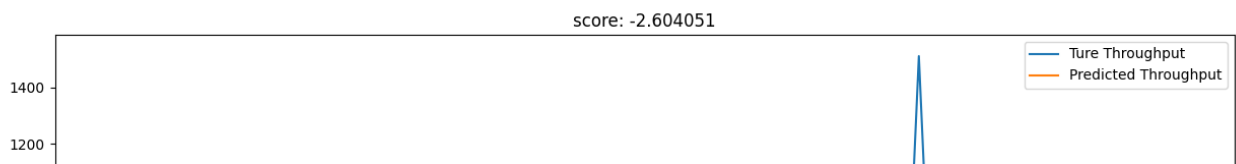
plt.scatter(testData.compassDirection, testData.Throughput, color='blue', label= 'True throughput')
plt.plot(testData.compassDirection, predictions.Throughput, color='red', label='Predicted throughput')

```

After I ran the model, I visualized the results and got a picture representing the decision tree regression results. As can be seen from the figure, we cannot predict 'Throughput' accurately by using 'compassDirection'.



### 2.2.8.2 Predictive Data Analytics Solution: Accuracy & Performance



The accuracy of decision tree regression prediction is really low ( $r^2$  score is very bad). The model did not perform good. But I don't think it is the problem of the decision tree. The 'Throughput' is affected by a lot of factors and this result shows that the 'compassDirection' won't have an influence on throughput.

This gives me the answer for my problem: The throughput of a signal source has nothing to do with moving direction.

## Extension 3: Name of the Extension

---

### 2.2.6 Main Data Analytics Elements/Concepts

*Provide a brief description of the proposed expansion to your group descriptive data analytics with **one analytical (decision) variables of interest** that will enable within your application domain problem to produce high rich “information”, which in turn can be a base to identify insights about the context of the application domain. It can be **based on the extension of your entities and the associated relations in “Deliverable 1”***

### 2.2.7 Data Model Definition

*Develop and provide a brief description of the Druid Ingestion Spec (equivalent to DDL to define **your new Dimension/Fact table(s)** for the group’s descriptive data analytics (DW) schema*

Below is the table of “tower id” in my dataset, it records the id of each connected tower in the different rounds. Different id represents different towers helping us to identify which tower signal received from, therefore helping us to calculate the relationship between signal and distance. And it also contains the definition of each parameter in this entity, like the type of data, the name of each column. The type of “latitude” and “longitude” are double and the type of “tower\_id” is long. This can assist us to link each tower with an accurate location.

```

{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMShare/in/Lumos5G-tower_id.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    },
    "dataSchema": {
      "dataSource": "Lumos5G-tower_id",
      "timestampSpec": {
        "column": "!!!_no_such_column_!!!",
        "missingValue": "2010-01-01T00:00:00Z"
      },
      "dimensionsSpec": {
        "dimensions": [
          {
            "type": "double",
            "name": "latitude"
          },
          {
            "type": "double",
            "name": "longitude"
          },
          "nrStatus",
          {
            "type": "long",
            "name": "tower_id"
          }
        ]
      },
      "granularitySpec": {
        "queryGranularity": "none",
        "rollup": false,
        "segmentGranularity": "hour"
      }
    }
  }
}

```

## 2.2.8 Data Warehouse Population (ETL)

*Provide brief description of how you used Airflow-based ETL to populate and maintain [your new Dimension/Fact table\(s\)](#).*



Add lookup

NameTowerStu

Tier\_\_default ▾

Version2021-12-05T04:53:07.391ZSet to current ISO time

FormJSON

Typemap ▾

Map

```
{  
  "1": "CONNECTED"  
  "0": "NOT_RESTRICTED"  
}
```

CloseSubmit

[illegible]

Provide samples and description of the **your new answers or/and reports** for your descriptive data analytics problem generated using Druid SQL and Superset as OLAP with visualization

The screenshot shows the Apache Superset web interface. At the top, there's a navigation bar with tabs for 'Quickstart', 'Apache Druid', 'Superset', and others. Below the navigation bar, a green notification banner states 'Task submitted successfully. Going to task view...'. The main content area is divided into two sections. The top section, titled 'Supervisors', has a 'Refresh' button and a 'Columns (5/5)' dropdown. It contains a table with headers 'Datasource', 'Type', 'Topic/Stream', 'Status', and 'Actions', but it is currently empty with a 'No supervisors' message. The bottom section, titled 'Tasks', has a 'Settings' button, a 'Group by' dropdown set to 'None', and a 'Columns (9/9)' dropdown. It contains a table with headers 'Task ID', 'Group ID', 'Type', 'Datasource', 'Location', 'Created time', 'Status', 'Duration', and 'Actions'. One task is listed with the following details: Task ID 'index\_parallel\_Lumos5G-throughput\_gkijhfm\_2021-12-05T04:24:00.949Z', Group ID 'index\_parallel\_Lumos5G-throughput\_gkijhfm\_...', Type 'index\_parallel', Datasource 'Lumos5G-th...', Location 'localhost:8100', Created time '2021-12-05T04:24:00.954Z', and Status 'RUNNING'.

```
SELECT "Throughput",
       "nrStatus",
       "run_num",
       "seq_num",
       "tower_id"
FROM "druid"."qwz2"
```

tower_id ^	nrStatus ^	COUNT(nrStatus) ^
1	NOT_RESTRICTED	89
2	NOT_RESTRICTED	2
3	NOT_RESTRICTED	8.05k
3	CONNECTED	3.95k
4	NOT_RESTRICTED	15
5	NOT_RESTRICTED	4
6	NOT_RESTRICTED	15
7	CONNECTED	44
7	NOT_RESTRICTED	39
8	CONNECTED	21
8	NOT_RESTRICTED	8
9	CONNECTED	1.52k
9	NOT_RESTRICTED	138
9	NONE	77
10	NOT_RESTRICTED	611
10	CONNECTED	423
11	CONNECTED	2.7k
11	NOT_RESTRICTED	601
12	CONNECTED	542
12	NOT_RESTRICTED	199
13	CONNECTED	3.3k

### 2.3.6 Predictive Data Analytics problem

*Provide a brief description of a predictive data analytics problem based on the Application domain of your chosen dataset to produce insights about the context of the application domain in the form of*

- *either producing predictions based on new instances of the application domain*
- *or discovering hidden properties related to the application domain that can a base to identify new insights*

### 2.3.7 Predictive Data Model

*Develop and provide a description of an adequate data model capturing the data elements required for your predictive data analytics*

### 2.3.8 Predictive Data Analytics Solution Pipeline

*Develop and provide a description of the data predictive pipeline (using IBM Cloud Pak for Data and Spark ML) required to solve your predictive data analytics problem described above. It should be integrated with your data warehouse system developed for Deliverable 2 of the project as the data source. Provide a brief description of each component and its usage*



### 2.3.8.1 Samples of the Predictive Data Analytics Solutions

*Provide samples of the desirable answers of your predictive data analytics problem described above*

### 2.3.8.2 Predictive Data Analytics Solution: Accuracy & Performance

*Provide your analysis of the produced analysis in terms of accuracy and performance.*

## Extension 4: Name of the Extension

---

### 2.2.11 Main Data Analytics Elements/Concepts

*Provide a brief description of the proposed expansion to your group descriptive data analytics with one analytical (decision) variables of interest that will enable within your application domain problem to produce high rich "information", which in turn can be a base to identify insights about the context of the application domain. It can be based on the extension of your entities and the associated relations in "Deliverable 1"*

### 2.2.12 Data Model Definition

*Develop and provide a brief description of the Druid Ingestion Spec (equivalent to DDL to define your new Dimension/Fact table(s) for the group's descriptive data analytics (DW) schema*

```

{
  "type": "index_parallel",
  "spec": {
    "ioConfig": {
      "type": "index_parallel",
      "inputSource": {
        "type": "local",
        "baseDir": "/mnt/hgfs/VMShare/in/LumosSG-Absthroughput.csv",
        "filter": "*.csv"
      },
      "inputFormat": {
        "type": "csv",
        "findColumnsFromHeader": true
      }
    },
    "tuningConfig": {
      "type": "index_parallel",
      "partitionsSpec": {
        "type": "dynamic"
      }
    },
    "dataSchema": {
      "dataSource": "LumosSG-Absthroughput",
      "timestampSpec": {
        "column": "!!!_no_such_column_!!!",
        "missingValue": "2010-01-01T00:00:00Z"
      },
      "dimensionsSpec": {
        "dimensions": [
          {
            "type": "long",
            "name": "abstractSignalStr"
          },
          {
            "type": "double",
            "name": "latitude"
          },
          {
            "type": "double",
            "name": "longitude"
          },
          {
            "type": "long",
            "name": "Throughput"
          }
        ]
      },
      "granularitySpec": {
        "queryGranularity": "none",
        "rollup": false,
        "segmentGranularity": "day"
      }
    }
  }
}

```

### 2.2.13 Data Warehouse Population (ETL)

*Provide brief description of how you used Airflow-based ETL to populate and maintain [your new Dimension/Fact table\(s\)](#).*



druid

Load data Ingestion Datasources Segments

Task submitted successfully. Going to task view...

### Supervisors

Refresh

Columns (5/5)

Datasource	Type	Topic/Stream	Status	Actions
No supervisors				

Software Updater Previous Page 1 of 1 20 rows Next

### Tasks

Group by None Group ID Type Datasource Status Refresh

Columns (9/9)

Task ID	Group ID	Type	Datasource	Location	Created time	Status	Duration	Acti
Index_parallel_Lumos5G-Absthroughput...	Index_parallel_Lumos5G-Absthroughput...	Index_parallel	Lumos5G-Ab...	localhost:8100	2021-12-06T03:38:50.463Z	RUNNING		

Previous Page 1 of 1 20 rows Next

- untitled

10k rows

00:00:02.55



Show 200 entries

latitude	longitude	MAX(Throughput)
44.97550505	-93.26247285	1.92k
44.9753824	-93.2595931	1.89k
44.975453	-93.26230665	1.89k
44.97551085	-93.26243925	1.89k
44.975529	-93.2624469	1.89k
44.9755295	-93.26247935	1.89k
44.97491925	-93.26108595	1.88k
44.97602895	-93.26103545	1.88k
44.97535575	-93.259554	1.87k
44.97544125	-93.2623024	1.87k
44.9755003	-93.2624282	1.87k
44.9755377	-93.2624928	1.87k
44.9755496	-93.262583	1.87k
44.9757434	-93.2602787	1.87k
44.9743037	-93.2614081	1.86k
44.9744671	-93.2612637	1.86k
44.9749021	-93.260908	1.86k
44.9753488	-93.259534	1.86k
44.9753784	-93.2595888	1.86k

1

2

3

4

5

6

7

...

50

## 2.1.6 Predictive Data Analytics problem

*Provide a brief description of a predictive data analytics problem based on the Application domain of your chosen dataset to produce insights about the context of the application domain in the form of*

- either producing predictions based on new instances of the application domain*
- or discovering hidden properties related to the application domain that can a base to identify new insights*

## **2.1.7 Predictive Data Model**

*Develop and provide a description of an adequate data model capturing the data elements required for your predictive data analytics*

## **2.1.8 Predictive Data Analytics Solution Pipeline**

*Develop and provide a description of the data predictive pipeline (using IBM Cloud Pak for Data and Spark ML) required to solve your predictive data analytics problem described above. It should be integrated with your data warehouse system developed for Deliverable 2 of the project as the data source. Provide a brief description of each component and its usage*

### **2.1.8.1 Samples of the Predictive Data Analytics Solutions**

*Provide samples of the desirable answers of your predictive data analytics problem described above*

### **2.1.8.2 Predictive Data Analytics Solution: Accuracy & Performance**

*Provide your analysis of the produced analysis in terms of accuracy and performance.*