

Explorer : Data visualisation Dashboarding

ISSUED BY

EL KORCHI MOHAMED

SUPERVISED BY

Prof EL HAJJI Mohamed

EX1:

1. Make a synthesis of the main concepts presented in this paper

This paper presents many concepts that are concerned by dashboarding, also it's been done on a corpus of dashboards collected from different sources in order to categorise the dashboards into different types and figure out how to make a good dashboard and what's different between each dashboard in terms of purpose and the audiences this dashboard is presented to.

Also this paper discusses some challenges that the dashboards are facing in terms of data or social impact it has, and also it talks about the privacy of the people and the security concerns about the data that the dashboard is showing to the public.

And then finally the paper concluded with some research challenges in data visualisation and dashboarding that the world of research is facing right now.

2. What are the main challenges set out in this paper?

- data design

data design goes in this direction. Indeed, data design means “the design of data”, and design can be defined as placing the user at the heart of the solution to a problem.

- metrics

Choosing the right metrics poses challenges for those less versed in analytics culture, particularly where trustworthy communication is the primary purpose such as in social organisations and public outreach. Most social organisations lack both organisational and data literacy to build up metrics and representations .

- Impoverished data vocabulary

the need for more sophisticated metrics and representations to express the nuances of complex problem solving. Such constrained “scorecard” views and quantitative models amenable to traditional data analytics can impose mechanistic, unsophisticated decision- making regimes that are susceptible to misinterpretation.

- * Adaptivity

Different tasks require different levels of data: both richer access to data through drill-down or roll-up and task-appropriate data filtering. For dashboards used in critical applications (e.g. healthcare), timely access to the “right data” is particularly important as decisions are made collaboratively across different situational and personnel contexts. For personal analytics applications, users tend to want control to select the data that matters for their personal goals.

- Social Impact

dashboards were often the primary interface to how people use “big data:” consequently, dashboard use is often a proxy for the social impacts of big data practices. Concerns raised included data in the workplace, trust, and issues related to “ data limited thinking , unintended reactions to privacy who controls the data democratisation:”to see what. These issues widen the discussion of dashboard design, introducing challenges of representation and framing relevant to both designers and researchers.

- Data-driven thinking

Data views are not neutral although they convey the impression of objectivity and “truth”: how they are framed and visualized and

perspectives of the authors and determines their. People interpreted the data as “trustworthy and definitive” because the visual organisation emphasised those associations as objective.

Dashboard use often restricts the scope of interpretation and decision making.

- Social data as context

How the data framed and represented present challenges. Comparisons that emphasize competition can be highly motivational.

For example, “the overriding element that helped to persuade some to remain engaged was the ability to share and compare energy consumption”. Yet such competition can be de-motivating when the user feels the thresholds of performance are out of reach.

- Sharing, security, and trust

When dashboards become portals to the information system, there are questions of an agency. Who can see the data at all levels? Is access ? Authoring models for data can be both unclear and insecure, leading to issues of trust.

- Privacy

The concerns of surveillance and anxiety about what data were being captured, who had access, and how they were interpreted.

3. Classify and cite the main rules for better dashboard design.

- 1. Consider your audience

Concerning dashboard best practices in design, your audience is one of the most important factors you have to take into account. You need to know who's going to use the dashboard and for what purpose they will use it in order to create the best analytical tool for them.

- **2. Determine your goals**

The next dashboard UI design principle has a direct relationship between the user's needs and the purpose of the dashboard, which is to establish your ultimate goals. Whether you are creating a client dashboard or an internal report, each dashboard that you create will serve a purpose and answer key questions through the data. Here, it is important to consider that not all the data available will be useful for the analysis process and that getting this part of the process wrong can render your further efforts meaningless.

- **3. Tell a story with your data**

Following the workflow for effective dashboard design UX comes the moment to start building your data storytelling.

To put it simply, dashboard storytelling is the process of presenting data in a visual manner that will depict the whole narrative of the data analysis process in order to efficiently understand business strategies and goals. In other words, efficient storytelling will help you communicate your message in the clearest way possible.

- **4. Provide context**

Without providing context, how will you know whether those numbers are good or bad, or if they are typical or unusual? Without comparison values, numbers on a dashboard are meaningless for the users. And more importantly, they won't know whether any action is required. For example, a management dashboard design will focus on high-level metrics that are easy to compare and, subsequently, offer a visual story.

- **5. Don't try to place all the information on the same page**

The next in our rundown of dashboard design tips is a question of information. This most golden of dashboard design principles refers to both precision and the right audience targeting.

- **6. Select the right type of dashboard**

Another best practice to consider is to be aware of the type of dashboard that you want to build based on its analytical purpose. As mentioned in previous points, each dashboard should be designed for a particular user group with the specific aim of assisting recipients in the business decision-making process. Information is valuable only

when it is directly actionable. The receiving user must be able to employ the information in his own business strategies and goals. As a dashboard designer who uses only the best dashboard design principles, make sure you can identify the key information, and separate it from the inessential one to enhance users' productivity.

- **7. Use the right type of chart**

We can't stress enough the importance of choosing the right data visualization types. You can destroy all of your efforts with a missing or incorrect chart type. It's important to understand what type of information you want to convey and choose a data visualization that is suited to the task.

- **8. Choose your layout carefully**

Dashboard best practices in design concern more than just good metrics and well-thought-out charts. The next step is the placement of charts on a dashboard. If your dashboard is visually organized, users will easily find the information they need. Poor layout forces users to think more before they grasp the point, and nobody likes to look for data in a jungle of charts and numbers

- **9. Prioritize simplicity**

One of the best practices for dashboard design focuses on simplicity. Nowadays, we can play with a lot of options in chart creation and it's tempting to use them all at once. However, try to use those frills sparingly. Frames, backgrounds, effects, gridlines... Yes, these options might be useful sometimes, but only when there is a reason for applying them.

- **10. Round your numbers**

Continuing on simplicity, rounding the numbers on your dashboard design should be also one of the priorities since you don't want your audience to be flooded with numerous decimal places. Yes, you want to present details but, sometimes, too many details give the wrong impression. If you want to present your conversion rate with 5 more decimal places, it would make sense to round the number and avoid too many number-specific factors.

- **11. Be careful with colors - choose a few and stick to them**

Without a shadow of a doubt, this is one of the most important of all dashboard design best practices.

This particular point may seem incongruous to what we have said up to this point, but there are options to personalise and customise your creations to your preferences.

- **12. Don't go over the top with real-time data**

Next on our list of good dashboard design tips refers to insight: *don't overuse real-time data*. In some cases, information displayed in too much detail only serves to lead to distraction. Unless you're tracking some live results, most dashboards don't need to be updated continually. Real-time data serves to paint a picture of a general situation or a trend. Most project management dashboards must only be updated periodically – on a weekly, daily, or hourly basis. After all, it is the right data that counts the most.

- **13. Be consistent with labeling and data formatting**

Number 12 on our list of tips on how to design a dashboard is focused on clarity and consistency. Above all else, in terms of functionality, the main aim of a data dashboard is to gain the ability to extract important insights at a swift glance. It's critical to make sure that your labeling and formatting are consistent across KPIs, tools, and metrics.

- **14. Use interactive elements**

Any comprehensive dashboard worth its salt will allow you to dig deep into certain trends, metrics, or insights with ease. When considering what makes a good dashboard, factoring drill-downs, click-to-filter, and time interval widgets into your design is vital.

- **15. Additionally, use animation options**

Animation options can be one of the dashboard elements that give an additional neat visual impression where you select the appearance of the specific element on the dashboard and assign an animation option. The result is a simple, yet effective automated movement based on the desired speed (slow, medium, or fast, e.g.) and types such as linear, swing, ease-in, or ease-out.

- **16. Double up your margins**

One of the most subtle yet essential dashboard guidelines, this principle boils down to balance. White space – also referred to as negative space – is the area of blankness between elements featured on a dashboard design.

- **17. Optimize for multiple devices**

Optimization for mobile or tablet is another critical point in the dashboard development process. By offering remote access to your most important insights, you can answer critical business questions on the go, without the need for a special office meeting. Benefits such as swift decision-making and instant access ensure everyone can look at the data on the fly.

- **18. Consider the use in terms of exports vs. digital**

In the process of dashboard designing, you also need to think about exports. You can use the dashboard itself and share it, but if you plan on regularly using exports, you might want to consider optimizing towards printing bounds, fewer colors, and different types of line styles to make sure everything is readable even on a black-and-white printout. Hence, when you plan your data dashboard design, you also need to look into the future uses and how to optimize towards different exporting options or simply share the dashboard itself with all its features and options.

- **19. Keep graphical integrity**

It might seem like an obvious point, but it is worth mentioning as it is one of the most important dashboard design trends. Graphical integrity basically refers to keeping the truth about the data. This means being objective about the values and not making them look a certain way that will benefit the analysis.

- **20. White label and embed if you need to**

Another critical point when considering your workflow for modern dashboard design is the opportunity to white label and embed the dashboard into your own application or intranet.

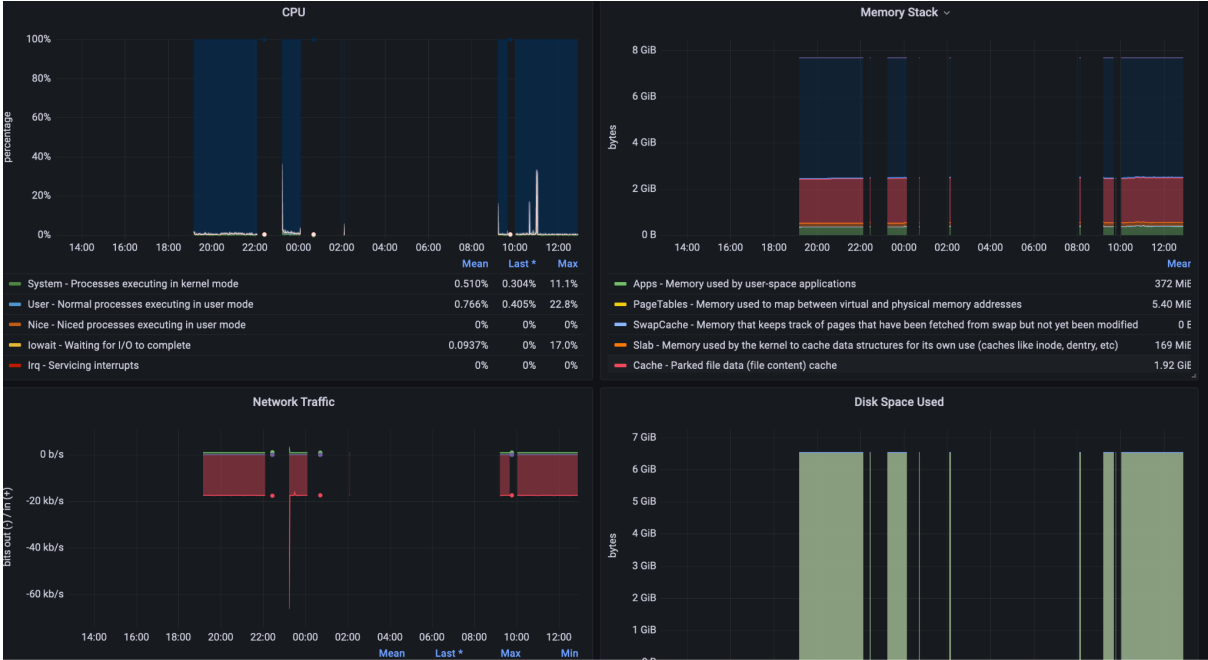
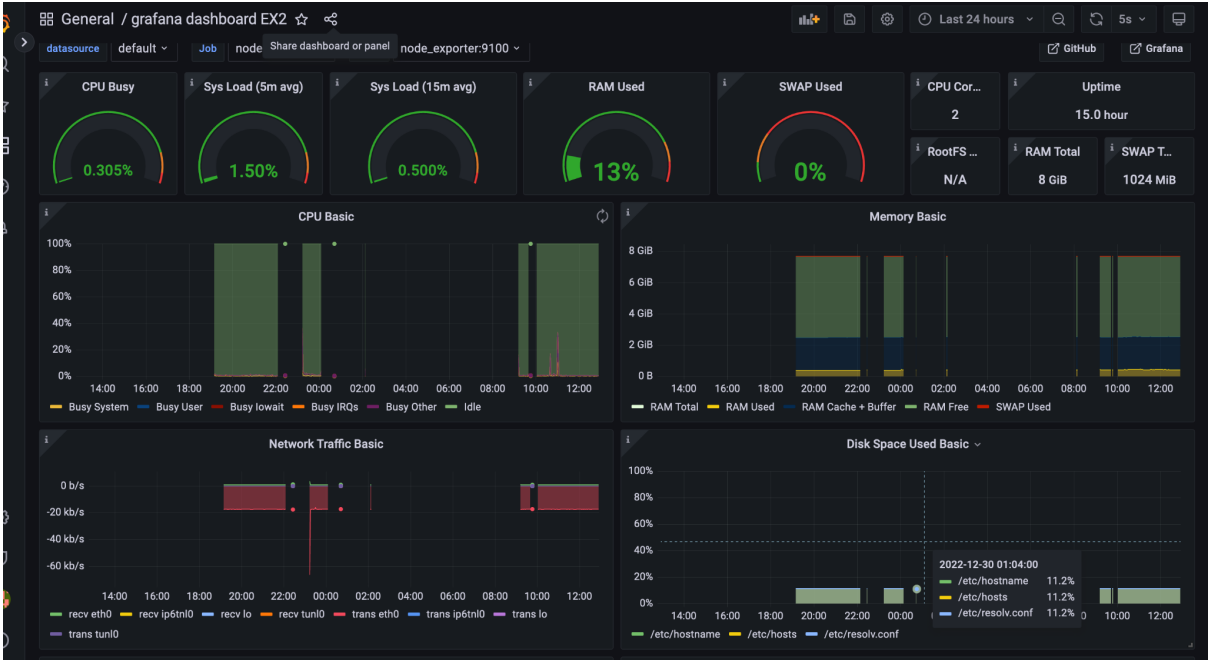
- **21. Avoid common data visualization mistakes**

Data visualization has evolved from simple static presentations to modern interactive software that takes visual perception to the next level. It also enabled average business users and advanced analysts to create stunning visuals that tell a clear data story to any potential audience profile, from beginners in a field to seasoned analysts and strategists.

EX2:

In this exercise I use docker compose to setup my environment (grafana,prometheus and node_exporter) as below:

```
Users > m > Desktop > dashboard lab > docker-compose.yml
1  version: "3"
2
3  services:
4    grafana:
5      image: grafana/grafana:9.3.2-ubuntu
6      container_name: grafana
7      ports:
8        - "3001:3000"
9      volumes:
10       - ./grafana-data:/var/lib/grafana
11      restart: unless-stopped
12
13   prometheus:
14     image: prom/prometheus:v2.41.0
15     container_name: prometheus
16     ports:
17       - 9090:9090
18     volumes:
19       - ./config
20       - ./prometheus-data:/prometheus
21     command:
22       - --config.file=/config/prometheus.yml
23       - --storage.tsdb.retention.time=1d
24     restart: unless-stopped
25
26   node_exporter:
27     image: prom/node-exporter:v1.5.0
28     container_name: node_exporter
29     ports:
30       - 9100:9100
31     restart: unless-stopped
```



Process

1. Producer

```
producer.ipynb > from kafka import KafkaProducer

+ Code + Markdown | ▶ Run All | ✖ Clear Outputs of All Cells | ⚠ Go To | 🔄 Restart | 📄 Variables | 📖 Outline | ⋮
Python 3.10.7

from kafka import KafkaProducer
import json
from json import dumps
import time

def json_serializer(data):
    return json.dumps(data).encode("utf-8")

Mybroker= '127.0.0.1:9092'
producer = KafkaProducer(
    bootstrap_servers='127.0.0.1:9092', value_serializer=lambda x: dumps(x).encode('utf-8'))
f = open('IOTData')

# returns JSON object as
# a dictionary
data = json.load(f)

# Iterating through the json
# list
topic="IotData"
for i in data:
    print(data[i])
    producer.send(topic, value=data[i])
    time.sleep(4)
    producer.flush()
    ##for j in data [i]:
        ##time.sleep(4)
        ##producer.flush()
        ##print(j)

# Closing file

##print(data)
f.close()
```

```
{'N': '90', 'P': '42', 'K': '43', 'temperature': '20.87974371', 'humidity': '82.00274423', 'ph': '6.502985292000001', 'rainfall': '202.9355362', 'label': 'rice'}
{'N': '85', 'P': '58', 'K': '41', 'temperature': '21.77046169', 'humidity': '80.31964408', 'ph': '7.038096361', 'rainfall': '226.6555374', 'label': 'rice'}
{'N': '60', 'P': '55', 'K': '44', 'temperature': '23.00445915', 'humidity': '82.3207629', 'ph': '7.840207144', 'rainfall': '263.9642476', 'label': 'rice'}
{'N': '74', 'P': '35', 'K': '40', 'temperature': '26.49109635', 'humidity': '80.15836264', 'ph': '6.980400905', 'rainfall': '242.8640342', 'label': 'rice'}
{'N': '78', 'P': '42', 'K': '42', 'temperature': '20.13017482', 'humidity': '81.60487287', 'ph': '7.628472891', 'rainfall': '262.7173405', 'label': 'rice'}
{'N': '69', 'P': '37', 'K': '42', 'temperature': '23.05804872', 'humidity': '83.37011772', 'ph': '7.073453503', 'rainfall': '251.0549998', 'label': 'rice'}
{'N': '69', 'P': '55', 'K': '38', 'temperature': '22.70883798', 'humidity': '82.63941394', 'ph': '5.70080568', 'rainfall': '271.3248604', 'label': 'rice'}
{'N': '94', 'P': '53', 'K': '40', 'temperature': '20.27774362', 'humidity': '82.89408619', 'ph': '5.718627177999999', 'rainfall': '241.9741949', 'label': 'rice'}
{'N': '89', 'P': '54', 'K': '38', 'temperature': '24.51588066', 'humidity': '83.53521629999999', 'ph': '6.685346424', 'rainfall': '230.4462359', 'label': 'rice'}
{'N': '68', 'P': '58', 'K': '38', 'temperature': '23.22397386', 'humidity': '83.03322691', 'ph': '6.336253525', 'rainfall': '221.2091958', 'label': 'rice'}
{'N': '91', 'P': '53', 'K': '40', 'temperature': '26.52723513', 'humidity': '81.41753846', 'ph': '5.386167788', 'rainfall': '264.6148697', 'label': 'rice'}
{'N': '90', 'P': '46', 'K': '42', 'temperature': '23.97898217', 'humidity': '81.45061596', 'ph': '7.50283396', 'rainfall': '250.0832336', 'label': 'rice'}
{'N': '78', 'P': '58', 'K': '44', 'temperature': '26.80079604', 'humidity': '80.88684822', 'ph': '5.108681786', 'rainfall': '284.4364567', 'label': 'rice'}
{'N': '93', 'P': '56', 'K': '36', 'temperature': '24.01497622', 'humidity': '82.05687182', 'ph': '6.98435366', 'rainfall': '185.2773389', 'label': 'rice'}
{'N': '94', 'P': '50', 'K': '37', 'temperature': '25.66585205', 'humidity': '80.66385045', 'ph': '6.94801983', 'rainfall': '209.5869708', 'label': 'rice'}
{'N': '60', 'P': '48', 'K': '39', 'temperature': '24.28209415', 'humidity': '80.30025587', 'ph': '7.042299068999999', 'rainfall': '231.0863347', 'label': 'rice'}
{'N': '85', 'P': '38', 'K': '41', 'temperature': '21.58711777', 'humidity': '82.7883708', 'ph': '6.2490506560000005', 'rainfall': '276.65524589999995', 'label': 'rice'}
{'N': '91', 'P': '35', 'K': '39', 'temperature': '23.79391957', 'humidity': '80.41817957', 'ph': '6.970859754', 'rainfall': '206.2611855', 'label': 'rice'}
{'N': '77', 'P': '38', 'K': '36', 'temperature': '21.8652524', 'humidity': '80.1923008', 'ph': '5.953933276', 'rainfall': '224.55501690000003', 'label': 'rice'}
{'N': '88', 'P': '35', 'K': '40', 'temperature': '23.57943626', 'humidity': '83.58760316', 'ph': '5.85393208', 'rainfall': '291.2986618000001', 'label': 'rice'}
{'N': '89', 'P': '45', 'K': '36', 'temperature': '21.32504158', 'humidity': '80.47476396', 'ph': '6.442475375', 'rainfall': '185.4974732', 'label': 'rice'}
{'N': '76', 'P': '40', 'K': '43', 'temperature': '25.15745531', 'humidity': '83.11713476', 'ph': '5.070175667', 'rainfall': '231.3843163', 'label': 'rice'}
{'N': '67', 'P': '59', 'K': '41', 'temperature': '21.94766735', 'humidity': '80.97384195', 'ph': '6.012632591', 'rainfall': '213.3560921', 'label': 'rice'}
{'N': '83', 'P': '41', 'K': '43', 'temperature': '21.0525355', 'humidity': '82.67839517', 'ph': '6.254028451', 'rainfall': '233.1075816', 'label': 'rice'}
{'N': '98', 'P': '47', 'K': '37', 'temperature': '23.48381344', 'humidity': '81.33265073', 'ph': '7.375482851', 'rainfall': '224.0581164', 'label': 'rice'}
...
{'N': '31', 'P': '56', 'K': '23', 'temperature': '31.46846241', 'humidity': '35.39454002', 'ph': '5.661826398', 'rainfall': '174.5723999', 'label': 'pigeonpeas'}
```

2. Consumer

```
from kafka import KafkaConsumer
from pymongo import MongoClient
from json import loads

client = MongoClient('localhost:27017')
collection = client.numtest.numtest

consumer = KafkaConsumer(
    'IotData',
    bootstrap_servers=['localhost:9092'],
    auto_offset_reset='earliest',
    enable_auto_commit=True,
    # group_id='IotData',
    value_deserializer=lambda x: loads(x.decode('utf-8')))

for message in consumer:
    message1 = message.value
    #collection.insert_one(message)
    print(message1)
    print(type(message1))
    #print('{} added to {}'.format(message, collection))
```

[2] 32m 1.5s

Python

... Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
{'N': '90', 'P': '42', 'K': '43', 'temperature': '20.87974371', 'humidity': '82.00274423', 'ph': '6.502985292000001', 'rainfall': '202.9355362', 'label': 'rice'}
<class 'dict'>

{'N': '85', 'P': '58', 'K': '41', 'temperature': '21.77046169', 'humidity': '80.31964408', 'ph': '7.038096361', 'rainfall': '226.6555374', 'label': 'rice'}
<class 'dict'>

{'N': '60', 'P': '55', 'K': '44', 'temperature': '23.00445915', 'humidity': '82.3207629', 'ph': '7.840207144', 'rainfall': '263.9642476', 'label': 'rice'}
<class 'dict'>
```

3. Spark processing

```
df = spark.read.csv("Crop_recommendation.csv", header=None)
df.show()
```

[6]

Python

	_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7
	N	P	K	temperature	humidity	ph	rainfall	label
	90	42	43	20.87974371	82.00274423	6.502985292000001	202.9355362	rice
	85	58	41	21.77046169	80.31964408	7.038096361	226.6555374	rice
	60	55	44	23.00445915	82.3207629	7.840207144	263.9642476	rice
	74	35	40	26.49109635	80.15836264	6.980400905	242.8640342	rice
	78	42	42	20.13017482	81.60487287	7.628472891	262.7173405	rice
	69	37	42	23.05804872	83.37011772	7.073453503	251.0549998	rice
	69	55	38	22.70883798	82.63941394	5.70080568	271.3248604	rice
	94	53	40	20.27774362	82.89408619	5.718627177999999	241.9741949	rice
	89	54	38	24.51588066	83.53521629999999	6.685346424	230.4462359	rice
	68	58	38	23.22397386	83.03322691	6.336253525	221.2091958	rice
	91	53	40	26.52723513	81.41753846	5.386167788	264.6148697	rice
	90	46	42	23.97898217	81.45061596	7.50283396	250.0832336	rice
	78	58	44	26.80079604	80.88684822	5.108681786	284.4364567	rice
	93	56	36	24.01497622	82.05687182	6.98435366	185.2773389	rice
	94	50	37	25.66585205	80.66385045	6.94801983	209.5869708	rice
	60	48	39	24.28209415	80.30025587	7.0422990689999995	231.0863347	rice
	85	38	41	21.58711777	82.7883708	6.2490506560000005	276.65524589999995	rice
	91	35	39	23.79391957	80.41817957	6.970859754	206.2611855	rice
	77	38	36	21.8652524	80.1923008	5.953933276	224.55501690000003	rice

4. Mongodb Atlas

Project 0

Data Services

App Services

Charts

DEPLOYMENT

Database

Data Lake

SERVICES

Triggers

Data API

Data Federation

SECURITY

Database Access

Network Access

Advanced

Goto

+ Create Database

Search Namespaces

sample_airbnb

sample_analytics

sample_geospatial

sample_guides

sample_mflix

sample_restaurants

sample_supplies

sample_training

sample_weatherdata

smart_farme

apple

banana

blackgram

smart_farme.apple

STORAGE SIZE: 24KB LOGICAL DATA SIZE: 12.6KB TOTAL DOCUMENTS: 100 INDEXES TOTAL SIZE: 20KB

FindIndexesSchema Anti-PatternsAggregationSearch Indexes

INSERT DOCUMENT

FILTER { field: 'value' }

OPTIONSApplyReset

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('63a30361a799f6179f9700bc')
N: 24
P: 128
K: 196
temperature: 22.75088787
humidity: 90.69489172
ph: 5.521466996
rainfall: 110.4317855
label: "apple"
```

PREVIOUS1-20 of many resultsNEXT

Project 0

Data Services

App Services

Charts

DEPLOYMENT

Database

Data Lake

SERVICES

Triggers

Data API

Data Federation

SECURITY

Database Access

Network Access

Advanced

New On Atlas

Goto

EL KORCHI'S ORG - 2021-06-14 > PROJECT 0

Database Deployments

Find a database deployment...

+ Create

Cluster0

Connect

View Monitoring

Browse Collections

...

FREE

SHARED

Enhance Your Experience

For production throughput and richer metrics, upgrade to a dedicated cluster now!

Upgrade

R 0

W 0

Last 6 hours

1.0/s

Connections 6.0

Last 6 hours

20.0

In 21.3 B/s

Out 216.8 B/s

Last 6 hours

5.2 KB/s

Data Size 338.9 MB

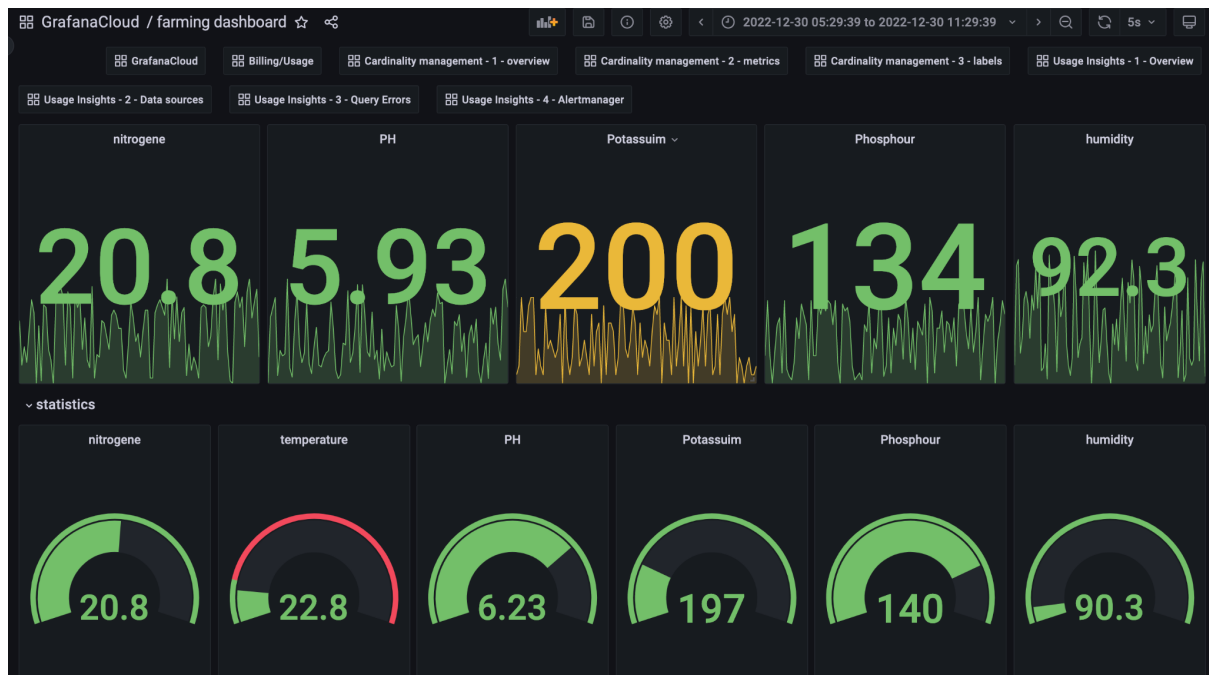
Last 10 days

512.0 MB

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SEARCH
5.0.14	AWS / Paris (eu-west-3)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index

5. Grafana visualization

In this step we have to install mongodb plugin into grafana platform ,next connect out atlas cluster with grafana with a connection string ,after that start creating our dashboard and some panels to visualize what we want to show (some metrics values and graphs)



In this image I visualize first the mean value for each metric for 10 days to let the farmer analyse and take the decision about his farm if he must do something .

Secondly show the metrics value for each 4 hours.



In this image I visualize four graphs:

Air quality:

This graph explains the relationship between the temperature and the humidity for each part of the farm and this to show if this part of the farm needs some ventilation and if the climatization is good for cultivation of this type of fruit.

Rainfall:

This graph shows rainfall for 30 days and the effect on the fruits.

Soil health:

This graph is very important to check soil health. There's a strong relationship between ph, phosphor ,nitrogen and potassium ,they explain how the soil's good for cultivation.

Soil health:

And in this last graph explain the difference between the new and the last value for each metric and the farmer could know if there's a problem or not to fix it to save his farm and his planting quality.

Here's my dashboard link :

<https://mohamedelkorchi.grafana.net/dashboard/snapshot/4Q2hUkLWbzNmkgYFa8dZ8D8jw7hAiyNM>