I535 COURSE PROJECT REPORT FALL 2022

BIG DATA CONCEPTS AND IMPLEMENTATION (rbhowmik)

1. INTRODUCTION

For my project I have selected the Tétouan city power consumption dataset. It's a multivariate and time series dataset. The historical dataset contains every 10 minutes data from 2017-01-01 and 2017-12-3, collected from Supervisory Control and Data Acquisition system. The dataset used in this study is related to three different power distribution networks of Tétouan city which is in north Morocco. The entire dataset has 52417 data points including 9 attributes. The dataset has zone wise power consumption values, including humidity, temperature, wind speed and other attributes leading to the power consumption distribution in the zones. I would be analyzing zone was power consumption with respect to temperature and humidity, and feature engineer multiple columns to check the thresholds.

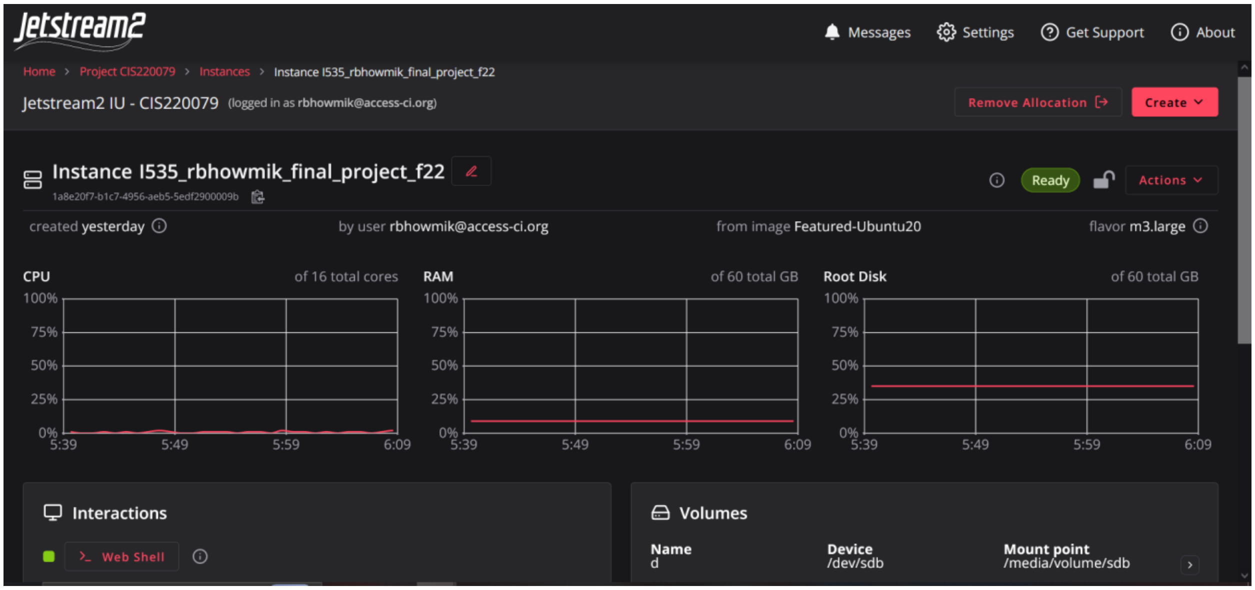
1. BACKGROUND

Power consumption is the most important use of resources in this world. Let it be in the field of manufacturing, IT, pharmaceutical etc., electricity consumption is the lifeline for all our day-to-day work. Predicting electricity power consumption is an important task which provides intelligence to utilities and helps them to improve their systems' performance in terms of productivity and effectiveness, and that's what make this data interesting, as it gives us a picture of the city of Tétouan and its power consumption, and how it varies based on temperature and humidity.

2. METHODS

a. VIRTUAL MACHINE SETUP

I created an ubuntu instance I535_rbhowmik_final_project_f22 under Project CIS220079 directory in Jetstream.



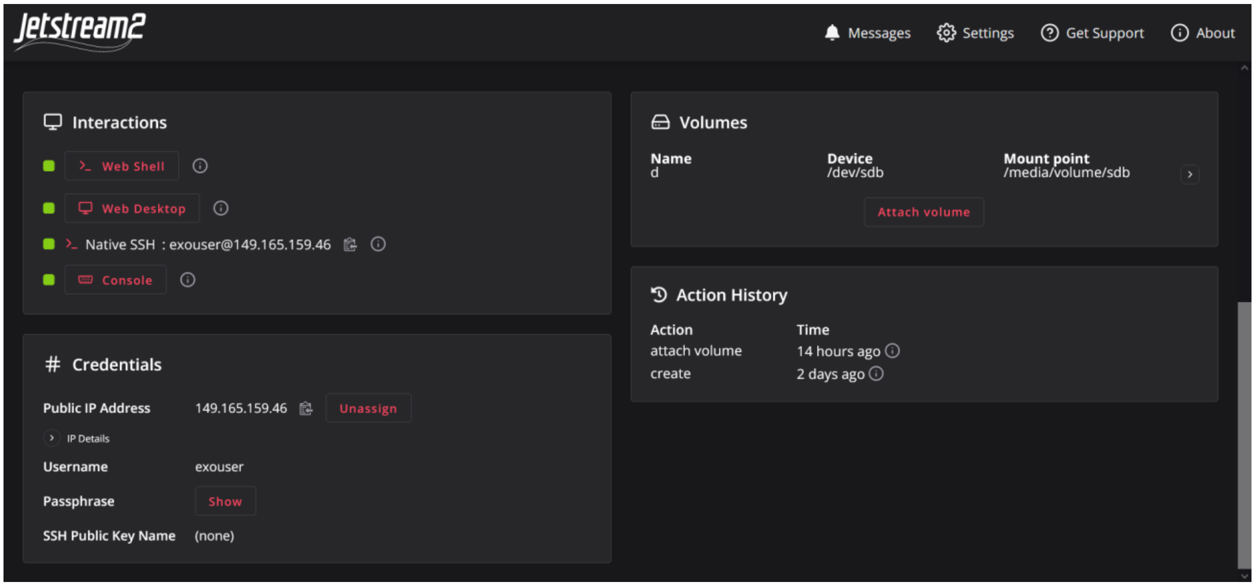
Jetstream2 IU - CIS220079 (logged in as rbhowmik@access-ci.org)

Instance I535_rbhowmik_final_project_f22

1a8e20f7-b1c7-4956-aeb5-5edf2900009b

created yesterday | by user rbhowmik@access-ci.org | from image Featured-Ubuntu20 | flavor m3.large

Name	Device	Mount point
d	/dev/sdb	/media/volume/sdb



Native SSH : exouser@149.165.159.46

Name	Device	Mount point
d	/dev/sdb	/media/volume/sdb

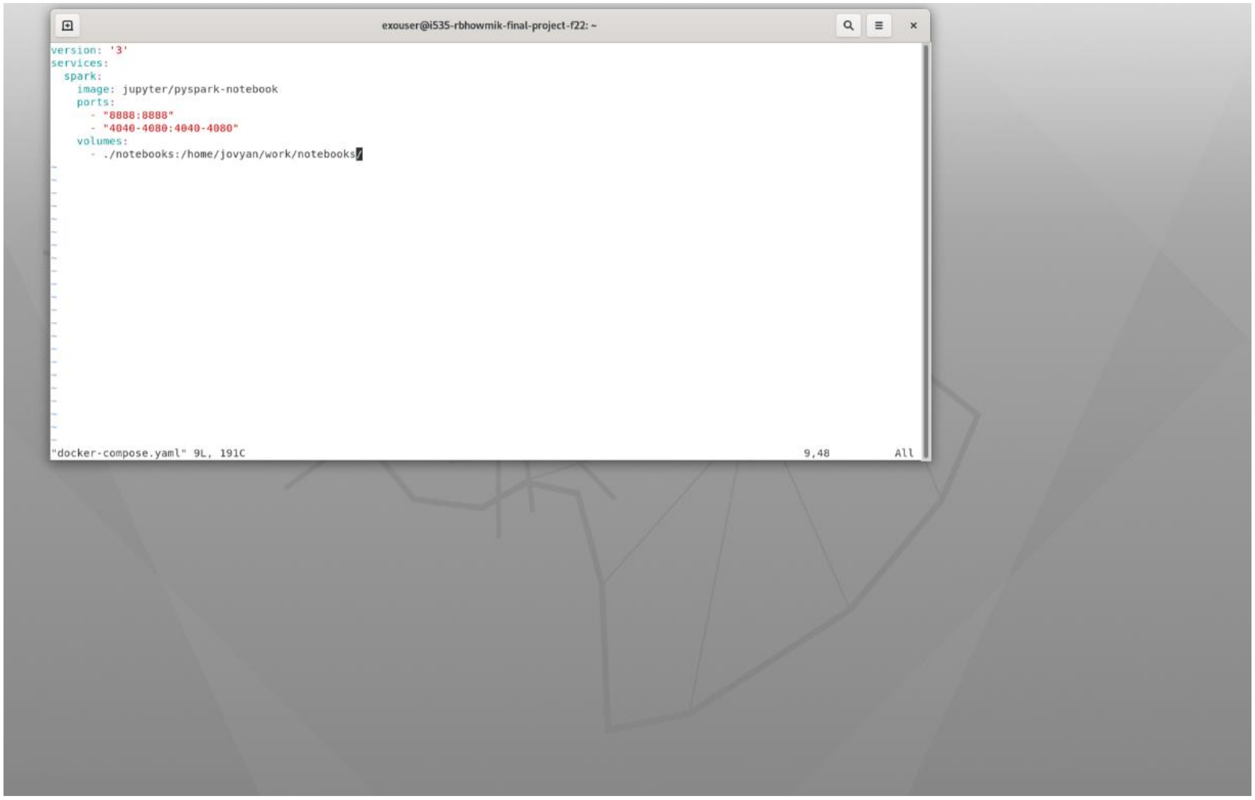
Action	Time
attach volume	14 hours ago
create	2 days ago

Public IP Address	149.165.159.46
Username	exouser
Passphrase	Show
SSH Public Key Name	(none)

I chose the m3.large with 16cores CPU, 60GB disk, to analyze my dataset. I also additionally attached a 10GB volume disk. I triggered the web-desktop and performed the anaconda installation. I have also installed additional packages like pymongo, seaborn and charstudio.

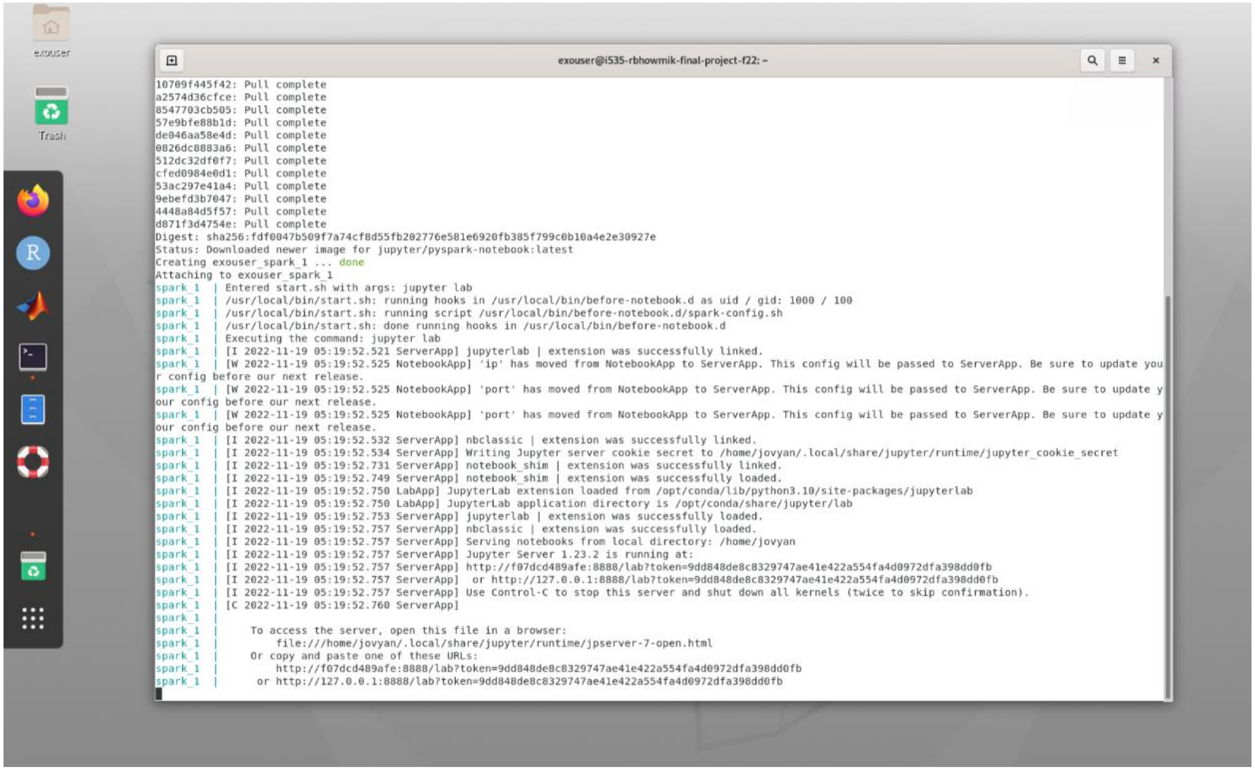
b. SPARK SETUP

Created a separate Spark folder in the instance having a docker engine running. Created a docker-compose.yaml file in the spark folder. In the docker-compose.yaml file I populated the below code for the pyspark to get instantiated.



```
version: '3'
services:
  spark:
    image: jupyter/pyspark-notebook
    ports:
      - "8888:8888"
      - "4040-4080:4040-4080"
    volumes:
      - ./notebooks:/home/jovyan/work/notebooks/
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"docker-compose.yaml" 9L, 191C                                        9,48          All
```

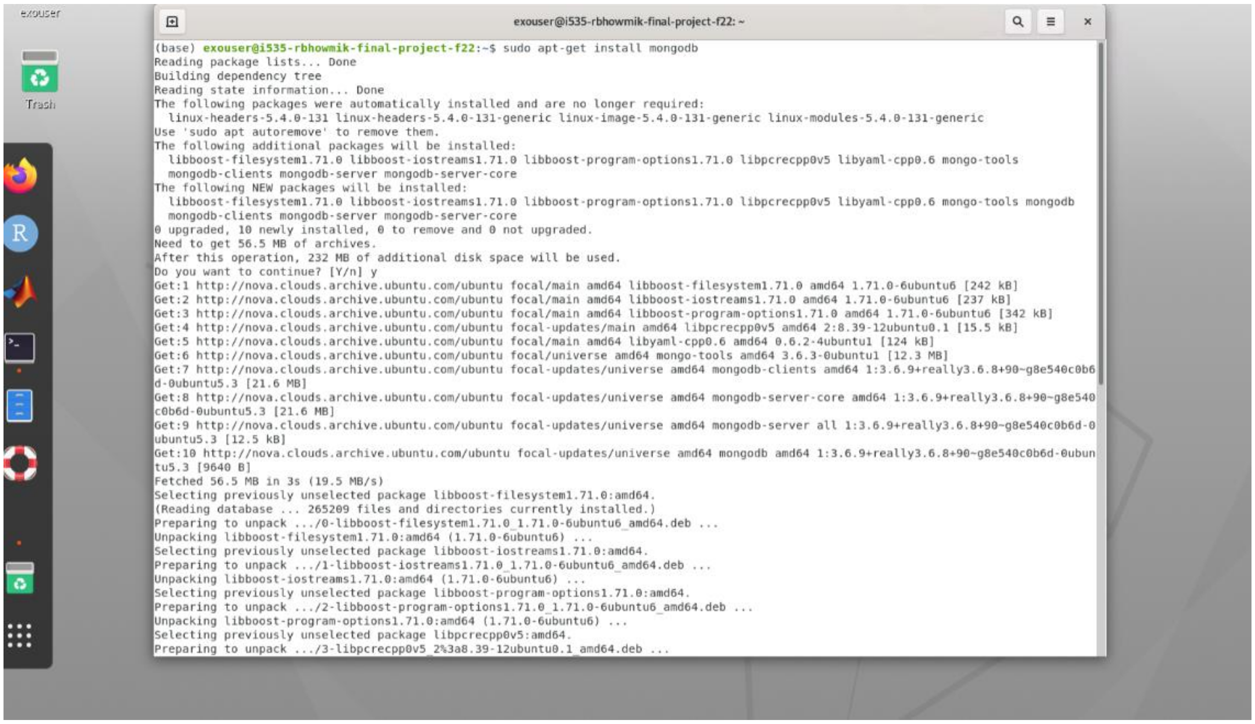
After running the sudo docker-compose command the pyspark notebook was triggered.



```
10709f445f42: Pull complete
a2574d36cfce: Pull complete
8547703cb505: Pull complete
57e9bfe88b1d: Pull complete
de046aa58e4d: Pull complete
0826dc8883a6: Pull complete
512dc32df0f7: Pull complete
cfed0984e0d1: Pull complete
53ac297e41a4: Pull complete
9ebefd3b7047: Pull complete
4448a84d5f57: Pull complete
d871f3d4754e: Pull complete
Digest: sha256:fdf0047b509f7a74cf8d55fb202776e581e6920fb385f799c0b10a4e2e30927e
Status: Downloaded newer image for jupyter/pyspark-notebook:latest
Creating exouser_spark_1 ... done
Attaching to exouser_spark_1
spark_1  | Entered start.sh with args: jupyter lab
spark_1  | /usr/local/bin/start.sh: running hooks in /usr/local/bin/before-notebook.d as uid / gid: 1000 / 100
spark_1  | /usr/local/bin/start.sh: running script /usr/local/bin/before-notebook.d/spark-config.sh
spark_1  | /usr/local/bin/start.sh: done running hooks in /usr/local/bin/before-notebook.d
spark_1  | Executing the command: jupyter lab
spark_1  | [I 2022-11-19 05:19:52.521 ServerApp] jupyterlab | extension was successfully linked.
spark_1  | [W 2022-11-19 05:19:52.525 NotebookApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update you
r config before our next release.
spark_1  | [W 2022-11-19 05:19:52.525 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update y
our config before our next release.
spark_1  | [W 2022-11-19 05:19:52.525 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update y
our config before our next release.
spark_1  | [I 2022-11-19 05:19:52.532 ServerApp] nbclassic | extension was successfully linked.
spark_1  | [I 2022-11-19 05:19:52.534 ServerApp] Writing Jupyter server cookie secret to /home/jovyan/.local/share/jupyter/runtime/jupyter_cookie_secret
spark_1  | [I 2022-11-19 05:19:52.731 ServerApp] notebook_shim | extension was successfully linked.
spark_1  | [I 2022-11-19 05:19:52.749 ServerApp] notebook_shim | extension was successfully loaded.
spark_1  | [I 2022-11-19 05:19:52.750 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.10/site-packages/jupyterlab
spark_1  | [I 2022-11-19 05:19:52.750 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
spark_1  | [I 2022-11-19 05:19:52.753 ServerApp] jupyterlab | extension was successfully loaded.
spark_1  | [I 2022-11-19 05:19:52.757 ServerApp] nbclassic | extension was successfully loaded.
spark_1  | [I 2022-11-19 05:19:52.757 ServerApp] Serving notebooks from local directory: /home/jovyan
spark_1  | [I 2022-11-19 05:19:52.757 ServerApp] Jupyter Server 1.23.2 is running at:
spark_1  | [I 2022-11-19 05:19:52.757 ServerApp] http://f07dcd489afe:8888/lab?token=9dd848de8c8329747ae41e422a554fa4d0972dfa398dd0fb
spark_1  | [I 2022-11-19 05:19:52.757 ServerApp]  or http://127.0.0.1:8888/lab?token=9dd848de8c8329747ae41e422a554fa4d0972dfa398dd0fb
spark_1  | [I 2022-11-19 05:19:52.757 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
spark_1  | [C 2022-11-19 05:19:52.760 ServerApp]
spark_1  |
spark_1  |     To access the server, open this file in a browser:
spark_1  |         file:///home/jovyan/.local/share/jupyter/runtime/jpserver-7-open.html
spark_1  |     Or copy and paste one of these URLs:
spark_1  |         http://f07dcd489afe:8888/lab?token=9dd848de8c8329747ae41e422a554fa4d0972dfa398dd0fb
spark_1  |      or http://127.0.0.1:8888/lab?token=9dd848de8c8329747ae41e422a554fa4d0972dfa398dd0fb
```

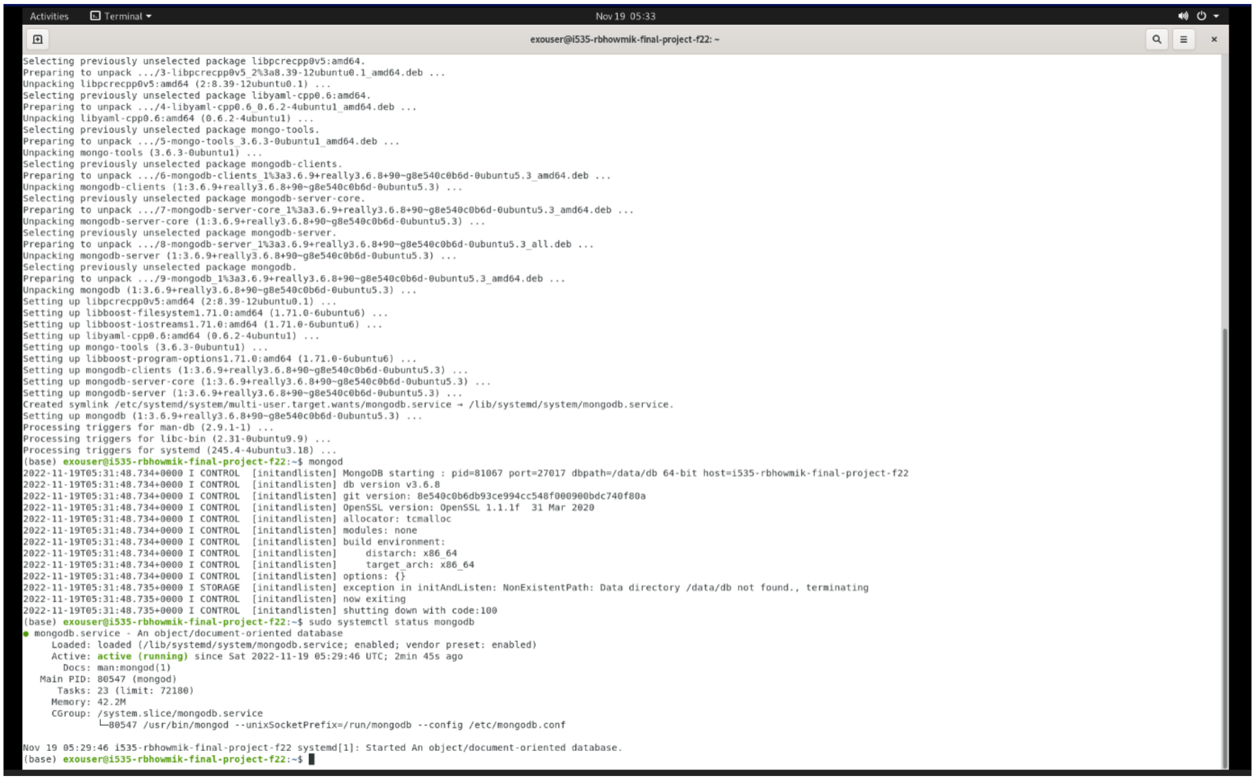
c. <u>MONGODB SETUP</u>

Before starting to analyze the dataset, the dataset is kept in a noSQL database i.e., MongoDB. So we installed the mongoDB using “sudo apt-get install mongodb”.



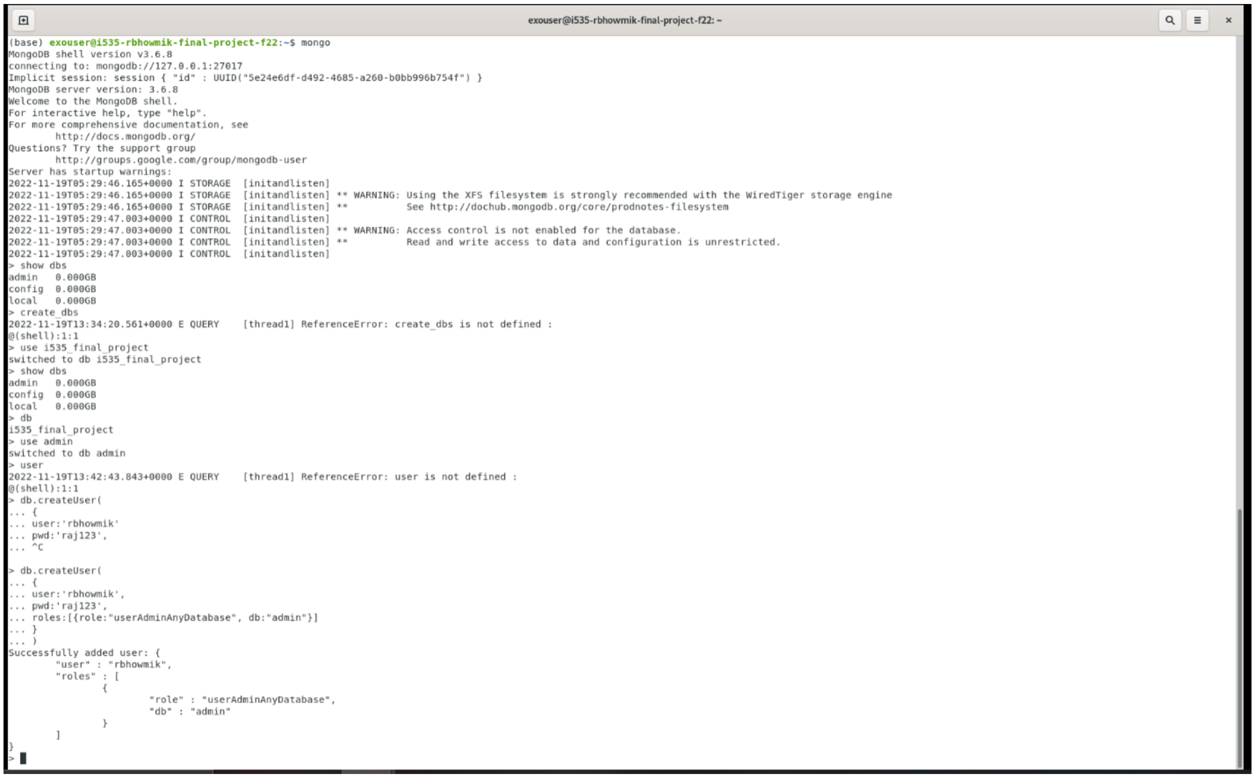
```
(base) exouser@i535-rbhowmik-final-project-f22:~$ sudo apt-get install mongodb
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-131 linux-headers-5.4.0-131-generic linux-image-5.4.0-131-generic linux-modules-5.4.0-131-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libboost-filesystem1.71.0 libboost-iostreams1.71.0 libboost-program-options1.71.0 libpcrecpp0v5 libyaml-cpp0.6 mongo-tools
  mongodb-clients mongodb-server mongodb-server-core
The following NEW packages will be installed:
  libboost-filesystem1.71.0 libboost-iostreams1.71.0 libboost-program-options1.71.0 libpcrecpp0v5 libyaml-cpp0.6 mongo-tools mongodb
  mongodb-clients mongodb-server mongodb-server-core
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 56.5 MB of archives.
After this operation, 232 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://nova.clouds.archive.ubuntu.com/ubuntu focal/main amd64 libboost-filesystem1.71.0 amd64 1.71.0-6ubuntu6 [242 kB]
Get:2 http://nova.clouds.archive.ubuntu.com/ubuntu focal/main amd64 libboost-iostreams1.71.0 amd64 1.71.0-6ubuntu6 [237 kB]
Get:3 http://nova.clouds.archive.ubuntu.com/ubuntu focal/main amd64 libboost-program-options1.71.0 amd64 1.71.0-6ubuntu6 [342 kB]
Get:4 http://nova.clouds.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpcrecpp0v5 amd64 2:8.39-12ubuntu0.1 [15.5 kB]
Get:5 http://nova.clouds.archive.ubuntu.com/ubuntu focal/main amd64 libyaml-cpp0.6 amd64 0.6.2-4ubuntu1 [124 kB]
Get:6 http://nova.clouds.archive.ubuntu.com/ubuntu focal/universe amd64 mongo-tools amd64 3.6.3-0ubuntu1 [12.3 MB]
Get:7 http://nova.clouds.archive.ubuntu.com/ubuntu focal-updates/universe amd64 mongodb-clients amd64 1:3.6.9+really3.6.8+90~g8e540c0b6
d-0ubuntu5.3 [21.6 MB]
Get:8 http://nova.clouds.archive.ubuntu.com/ubuntu focal-updates/universe amd64 mongodb-server-core amd64 1:3.6.9+really3.6.8+90~g8e540
c0b6d-0ubuntu5.3 [21.6 MB]
Get:9 http://nova.clouds.archive.ubuntu.com/ubuntu focal-updates/universe amd64 mongodb-server all 1:3.6.9+really3.6.8+90~g8e540c0b6d-0
ubuntu5.3 [12.5 kB]
Get:10 http://nova.clouds.archive.ubuntu.com/ubuntu focal-updates/universe amd64 mongodb amd64 1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubun
tu5.3 [9640 B]
Fetched 56.5 MB in 3s (19.5 MB/s)
Selecting previously unselected package libboost-filesystem1.71.0:amd64.
(Reading database ... 265209 files and directories currently installed.)
Preparing to unpack .../0-libboost-filesystem1.71.0_1.71.0-6ubuntu6_amd64.deb ...
Unpacking libboost-filesystem1.71.0:amd64 (1.71.0-6ubuntu6) ...
Selecting previously unselected package libboost-iostreams1.71.0:amd64.
Preparing to unpack .../1-libboost-iostreams1.71.0_1.71.0-6ubuntu6_amd64.deb ...
Unpacking libboost-iostreams1.71.0:amd64 (1.71.0-6ubuntu6) ...
Selecting previously unselected package libboost-program-options1.71.0:amd64.
Preparing to unpack .../2-libboost-program-options1.71.0_1.71.0-6ubuntu6_amd64.deb ...
Unpacking libboost-program-options1.71.0:amd64 (1.71.0-6ubuntu6) ...
Selecting previously unselected package libpcrecpp0v5:amd64.
Preparing to unpack .../3-libpcrecpp0v5_2%3a8.39-12ubuntu0.1_amd64.deb ...
```

After starting the mongoDB service, we checked the status using the “mongod” and “sudo systmectl status mongod”, which was running.



```
Selecting previously unselected package libpcrecpp0v5:amd64.
Preparing to unpack .../3-libpcrecpp0v5_2%3a8.39-12ubuntu0.1_amd64.deb ...
Unpacking libpcrecpp0v5:amd64 (2:8.39-12ubuntu0.1) ...
Selecting previously unselected package libyaml-cpp0.6:amd64.
Preparing to unpack .../4-libyaml-cpp0.6_0.6.2-4ubuntu1_amd64.deb ...
Unpacking libyaml-cpp0.6:amd64 (0.6.2-4ubuntu1) ...
Selecting previously unselected package mongo-tools.
Preparing to unpack .../5-mongo-tools_3.6.3-0ubuntu1_amd64.deb ...
Unpacking mongo-tools (3.6.3-0ubuntu1) ...
Selecting previously unselected package mongodb-clients.
Preparing to unpack .../6-mongodb-clients_1%3a3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3_amd64.deb ...
Unpacking mongodb-clients (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Selecting previously unselected package mongodb-server-core.
Preparing to unpack .../7-mongodb-server-core_1%3a3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3_amd64.deb ...
Unpacking mongodb-server-core (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Selecting previously unselected package mongodb-server.
Preparing to unpack .../8-mongodb-server_1%3a3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3_all.deb ...
Unpacking mongodb-server (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Selecting previously unselected package mongodb.
Preparing to unpack .../9-mongodb_1%3a3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3_amd64.deb ...
Unpacking mongodb (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Setting up libpcrecpp0v5:amd64 (2:8.39-12ubuntu0.1) ...
Setting up libboost-filesystem1.71.0:amd64 (1.71.0-6ubuntu6) ...
Setting up libboost-iostreams1.71.0:amd64 (1.71.0-6ubuntu6) ...
Setting up libyaml-cpp0.6:amd64 (0.6.2-4ubuntu1) ...
Setting up mongo-tools (3.6.3-0ubuntu1) ...
Setting up libboost-program-options1.71.0:amd64 (1.71.0-6ubuntu6) ...
Setting up mongodb-clients (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Setting up mongodb-server-core (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Setting up mongodb-server (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mongodb.service → /lib/systemd/system/mongodb.service.
Setting up mongodb (1:3.6.9+really3.6.8+90~g8e540c0b6d-0ubuntu5.3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for systemd (245.4-4ubuntu3.18) ...
(base) exouser@i535-rbhowmik-final-project-f22:~$ mongod
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] MongoDB starting : pid=81067 port=27017 dbpath=/data/db 64-bit host=i535-rbhowmik-final-project-f22
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] db version v3.6.8
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] git version: 8e540c0b6db93ce994cc548f000900bdc740f80a
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1f  31 Mar 2020
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] modules: none
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] build environment:
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten]     distarch: x86_64
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2022-11-19T05:31:48.734+0000 I CONTROL  [initandlisten] options: {}
2022-11-19T05:31:48.735+0000 I STORAGE  [initandlisten] exception in initAndListen: NonExistentPath: Data directory /data/db not found., terminating
2022-11-19T05:31:48.735+0000 I CONTROL  [initandlisten] now exiting
2022-11-19T05:31:48.735+0000 I CONTROL  [initandlisten] shutting down with code:100
(base) exouser@i535-rbhowmik-final-project-f22:~$ sudo systemctl status mongodb
● mongodb.service - An object/document-oriented database
     Loaded: loaded (/lib/systemd/system/mongodb.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-11-19 05:29:46 UTC; 2min 45s ago
       Docs: man:mongod(1)
   Main PID: 80547 (mongod)
      Tasks: 23 (limit: 72180)
     Memory: 42.2M
     CGroup: /system.slice/mongodb.service
             └─80547 /usr/bin/mongod --unixSocketPrefix=/run/mongodb --config /etc/mongodb.conf

Nov 19 05:29:46 i535-rbhowmik-final-project-f22 systemd[1]: Started An object/document-oriented database.
(base) exouser@i535-rbhowmik-final-project-f22:~$ 
```

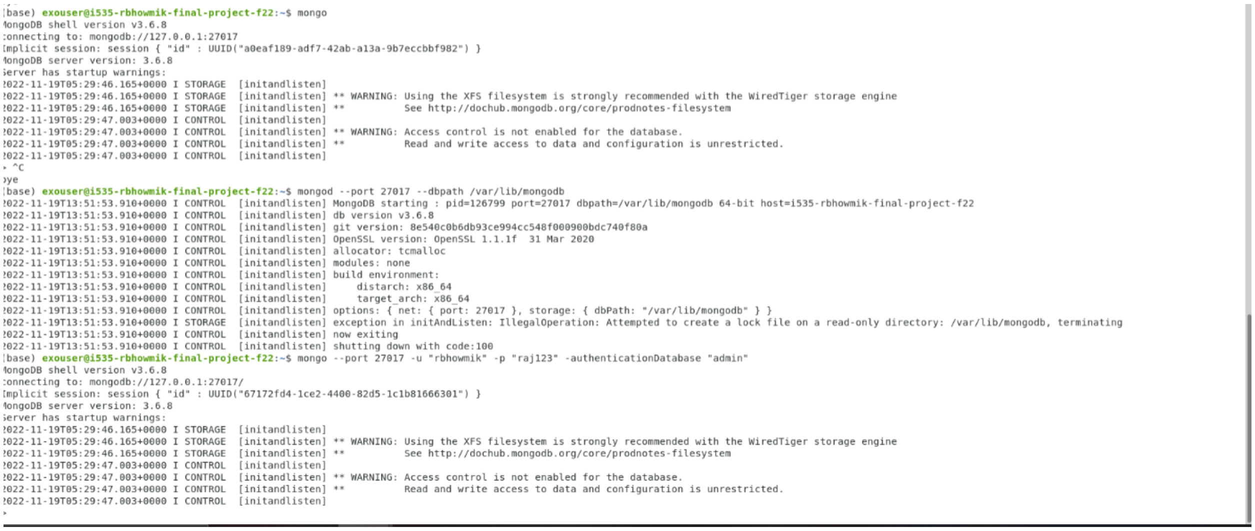
Using the mongo shell I checked for databases available using “show dbs”. I created the “i535_final_project” database dynamically in the code and created a collection “power consumption”. Created user admin by using “use admin”. After creating the user admin, I created my own account using db.createUser with user: rbhowmik, pwd: raj123 and roles:userAdminAnyDatabase.



exouser@i535-rbhowmik-final-project-f22: ~

```
(base) exouser@i535-rbhowmik-final-project-f22:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("5e24e6df-d492-4685-a260-b0bb996b754f") }
MongoDB server version: 3.6.8
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
Server has startup warnings:
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten]
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten]
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten]
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> create_dbs
2022-11-19T13:34:20.561+0000 E QUERY    [thread1] ReferenceError: create_dbs is not defined :
@(shell):1:1
> use i535_final_project
switched to db i535_final_project
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> db
i535_final_project
> use admin
switched to db admin
> user
2022-11-19T13:42:43.843+0000 E QUERY    [thread1] ReferenceError: user is not defined :
@(shell):1:1
> db.createUser(
... {
... user:'rbhowmik'
... pwd:'raj123',
... ^C

> db.createUser(
... {
... user:'rbhowmik',
... pwd:'raj123',
... roles:[{role:"userAdminAnyDatabase", db:"admin"}]
... }
... )
Successfully added user: {
        "user" : "rbhowmik",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ]
}
> █
```

I restarted mongodb to take the effect of new users and tested if it’s opening with my credentials.



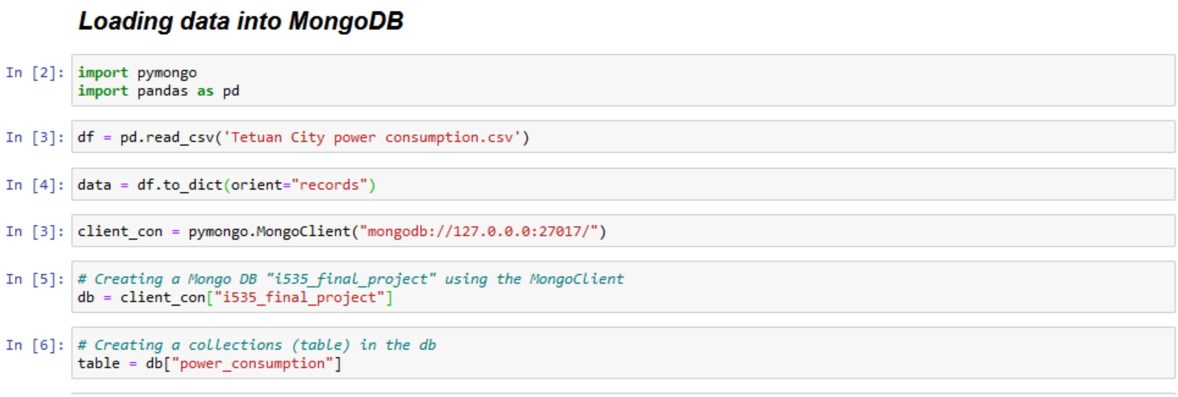
```
(base) exouser@i535-rbhowmik-final-project-f22:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("a0eaf189-adf7-42ab-a13a-9b7eccbbf982") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten]
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten]
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten]
> ^C
bye
(base) exouser@i535-rbhowmik-final-project-f22:~$ mongod --port 27017 --dbpath /var/lib/mongodb
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] MongoDB starting : pid=126799 port=27017 dbpath=/var/lib/mongodb 64-bit host=i535-rbhowmik-final-project-f22
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] db version v3.6.8
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] git version: 8e540c0b6db93ce994cc548f000900bdc740f80a
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1f  31 Mar 2020
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] modules: none
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] build environment:
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten]     distarch: x86_64
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] options: { net: { port: 27017 }, storage: { dbPath: "/var/lib/mongodb" } }
2022-11-19T13:51:53.910+0000 I STORAGE  [initandlisten] exception in initAndListen: IllegalOperation: Attempted to create a lock file on a read-only directory: /var/lib/mongodb, terminating
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] now exiting
2022-11-19T13:51:53.910+0000 I CONTROL  [initandlisten] shutting down with code:100
(base) exouser@i535-rbhowmik-final-project-f22:~$ mongo --port 27017 -u "rbhowmik" -p "raj123" -authenticationDatabase "admin"
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017/
Implicit session: session { "id" : UUID("67172fd4-1ce2-4400-82d5-1c1b81666301") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten]
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-11-19T05:29:46.165+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten]
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-11-19T05:29:47.003+0000 I CONTROL  [initandlisten]
>
```

d. STEPS TAKEN TO ADDRESS THE PROBLEM

I installed MongoDB, Spark, and Anaconda to help me analyze the large dataset of Tétouan city power consumption. Using the parallel processing technique of PySpark i.e., Spark session, Spark Context and Spark SQL I garnered multiple insights from the data. I used transformations, filtering operations, string operations in my data. I generated new features for the data based on a particular threshold for the humidity and temperature features. I classified both the features as: low, medium, and high to classify zone wise power consumption and to check what is the count of humidity and temperature. I used MongoDB to store the data to store the data, so that it can be used to perform ETL in future. The reason for using NoSQL database because data might be unstructured in future, and we just generalized it for future use. Additionally, I implemented a Pipeline to perform one-hot encoding, vector assembler, string indexing and put a linear regression model (algorithm from SparkML library) to pass the pipeline to.

3. RESULTS

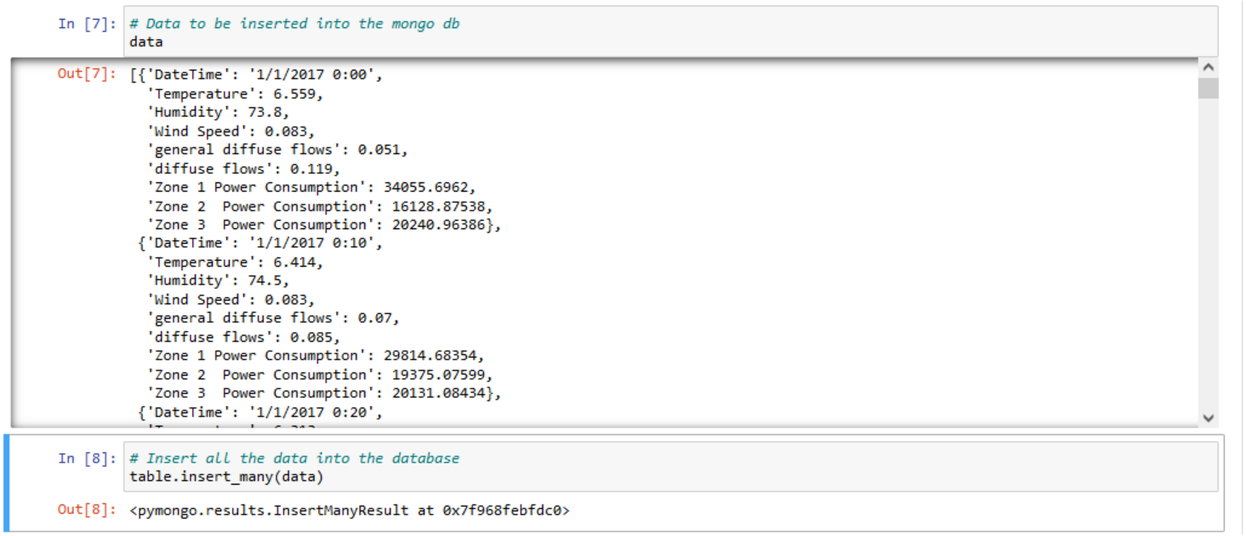
Loading the data using MongoDB



Loading data into MongoDB

```
In [2]: import pymongo
        import pandas as pd

In [3]: df = pd.read_csv('Tetuan City power consumption.csv')

In [4]: data = df.to_dict(orient="records")

In [3]: client_con = pymongo.MongoClient("mongodb://127.0.0.0:27017/")

In [5]: # Creating a Mongo DB "i535_final_project" using the MongoClient
        db = client_con["i535_final_project"]

In [6]: # Creating a collections (table) in the db
        table = db["power_consumption"]
```

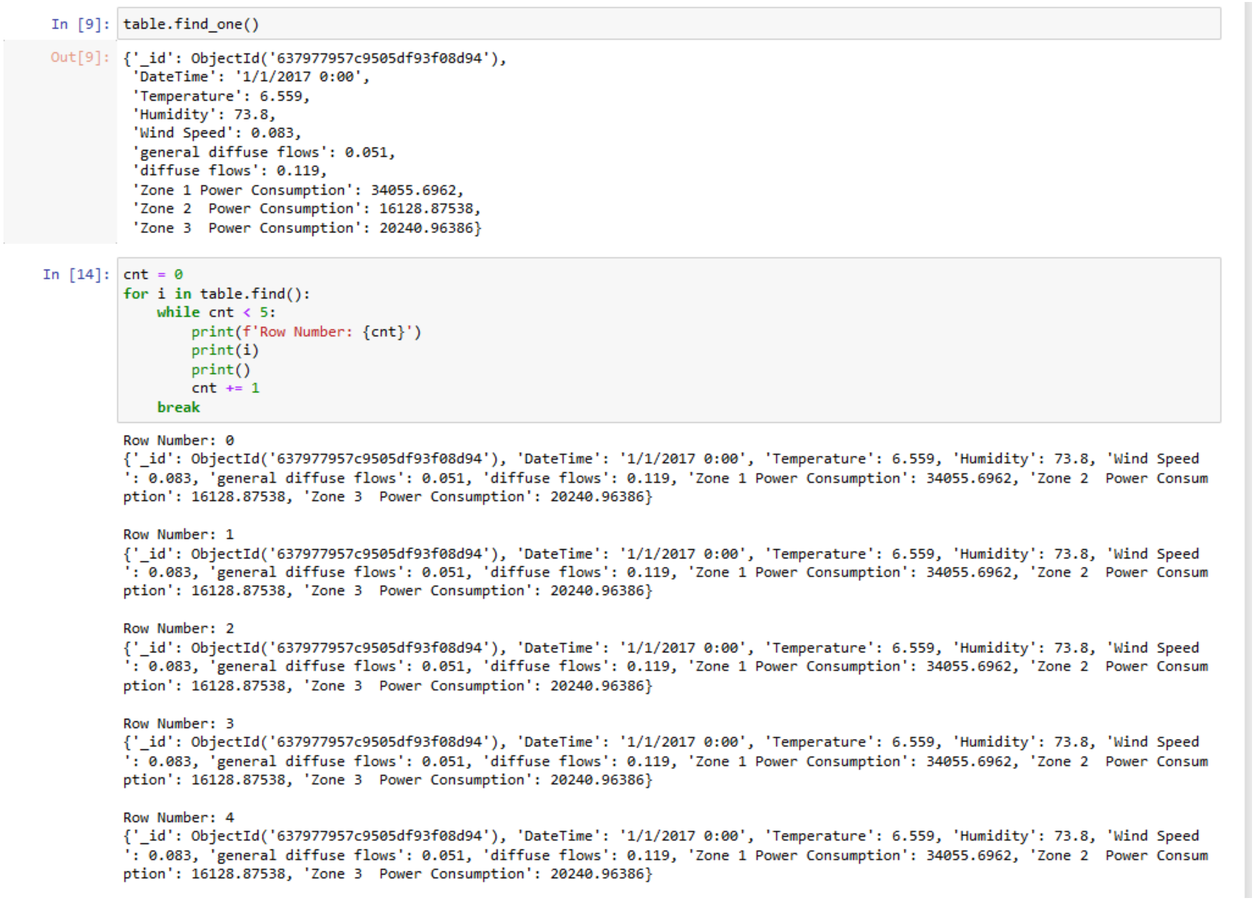
Displaying the data and adding into the database



```
In [7]:  # Data to be inserted into the mongo db
         data

Out[7]:  [{'DateTime': '1/1/2017 0:00',
           'Temperature': 6.559,
           'Humidity': 73.8,
           'Wind Speed': 0.083,
           'general diffuse flows': 0.051,
           'diffuse flows': 0.119,
           'Zone 1 Power Consumption': 34055.6962,
           'Zone 2  Power Consumption': 16128.87538,
           'Zone 3  Power Consumption': 20240.96386},
          {'DateTime': '1/1/2017 0:10',
           'Temperature': 6.414,
           'Humidity': 74.5,
           'Wind Speed': 0.083,
           'general diffuse flows': 0.07,
           'diffuse flows': 0.085,
           'Zone 1 Power Consumption': 29814.68354,
           'Zone 2  Power Consumption': 19375.07599,
           'Zone 3  Power Consumption': 20131.08434},
          {'DateTime': '1/1/2017 0:20',
```

```
In [8]:  # Insert all the data into the database
         table.insert_many(data)

Out[8]:  <pymongo.results.InsertManyResult at 0x7f968febfdc0>
```

Checking one record and multiple records

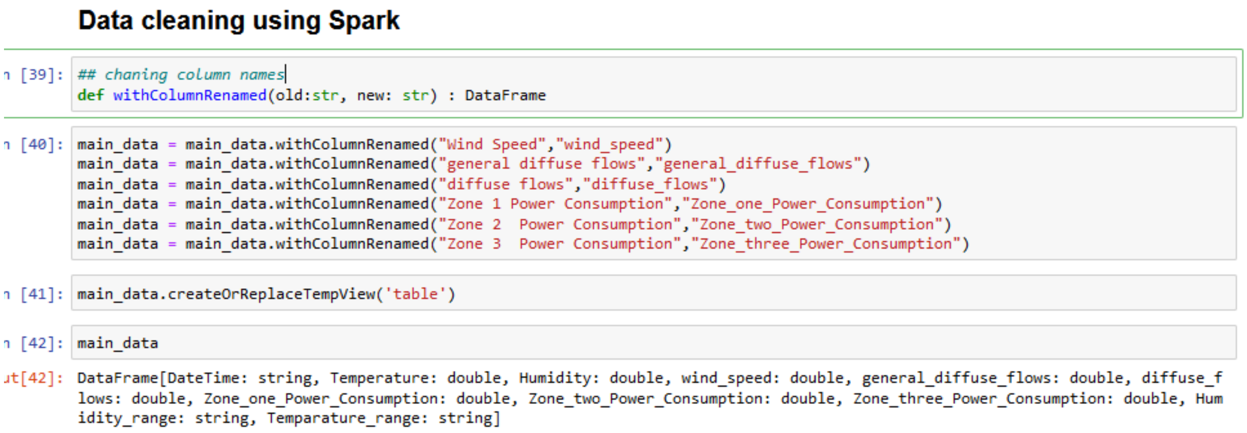


```
In [9]:  table.find_one()

Out[9]:  {'_id': ObjectId('637977957c9505df93f08d94'),
          'DateTime': '1/1/2017 0:00',
          'Temperature': 6.559,
          'Humidity': 73.8,
          'Wind Speed': 0.083,
          'general diffuse flows': 0.051,
          'diffuse flows': 0.119,
          'Zone 1 Power Consumption': 34055.6962,
          'Zone 2  Power Consumption': 16128.87538,
          'Zone 3  Power Consumption': 20240.96386}
```

```
In [14]: cnt = 0
         for i in table.find():
             while cnt < 5:
                 print(f'Row Number: {cnt}')
                 print(i)
                 print()
                 cnt += 1
             break

         Row Number: 0
         {'_id': ObjectId('637977957c9505df93f08d94'), 'DateTime': '1/1/2017 0:00', 'Temperature': 6.559, 'Humidity': 73.8, 'Wind Speed
         ': 0.083, 'general diffuse flows': 0.051, 'diffuse flows': 0.119, 'Zone 1 Power Consumption': 34055.6962, 'Zone 2  Power Consum
         ption': 16128.87538, 'Zone 3  Power Consumption': 20240.96386}

         Row Number: 1
         {'_id': ObjectId('637977957c9505df93f08d94'), 'DateTime': '1/1/2017 0:00', 'Temperature': 6.559, 'Humidity': 73.8, 'Wind Speed
         ': 0.083, 'general diffuse flows': 0.051, 'diffuse flows': 0.119, 'Zone 1 Power Consumption': 34055.6962, 'Zone 2  Power Consum
         ption': 16128.87538, 'Zone 3  Power Consumption': 20240.96386}

         Row Number: 2
         {'_id': ObjectId('637977957c9505df93f08d94'), 'DateTime': '1/1/2017 0:00', 'Temperature': 6.559, 'Humidity': 73.8, 'Wind Speed
         ': 0.083, 'general diffuse flows': 0.051, 'diffuse flows': 0.119, 'Zone 1 Power Consumption': 34055.6962, 'Zone 2  Power Consum
         ption': 16128.87538, 'Zone 3  Power Consumption': 20240.96386}

         Row Number: 3
         {'_id': ObjectId('637977957c9505df93f08d94'), 'DateTime': '1/1/2017 0:00', 'Temperature': 6.559, 'Humidity': 73.8, 'Wind Speed
         ': 0.083, 'general diffuse flows': 0.051, 'diffuse flows': 0.119, 'Zone 1 Power Consumption': 34055.6962, 'Zone 2  Power Consum
         ption': 16128.87538, 'Zone 3  Power Consumption': 20240.96386}

         Row Number: 4
         {'_id': ObjectId('637977957c9505df93f08d94'), 'DateTime': '1/1/2017 0:00', 'Temperature': 6.559, 'Humidity': 73.8, 'Wind Speed
         ': 0.083, 'general diffuse flows': 0.051, 'diffuse flows': 0.119, 'Zone 1 Power Consumption': 34055.6962, 'Zone 2  Power Consum
         ption': 16128.87538, 'Zone 3  Power Consumption': 20240.96386}
```

Data cleaning with spark

- Changing the column names



Data cleaning using Spark

```
n [39]: ## chaning column names
        def withColumnRenamed(old:str, new: str) : DataFrame
```

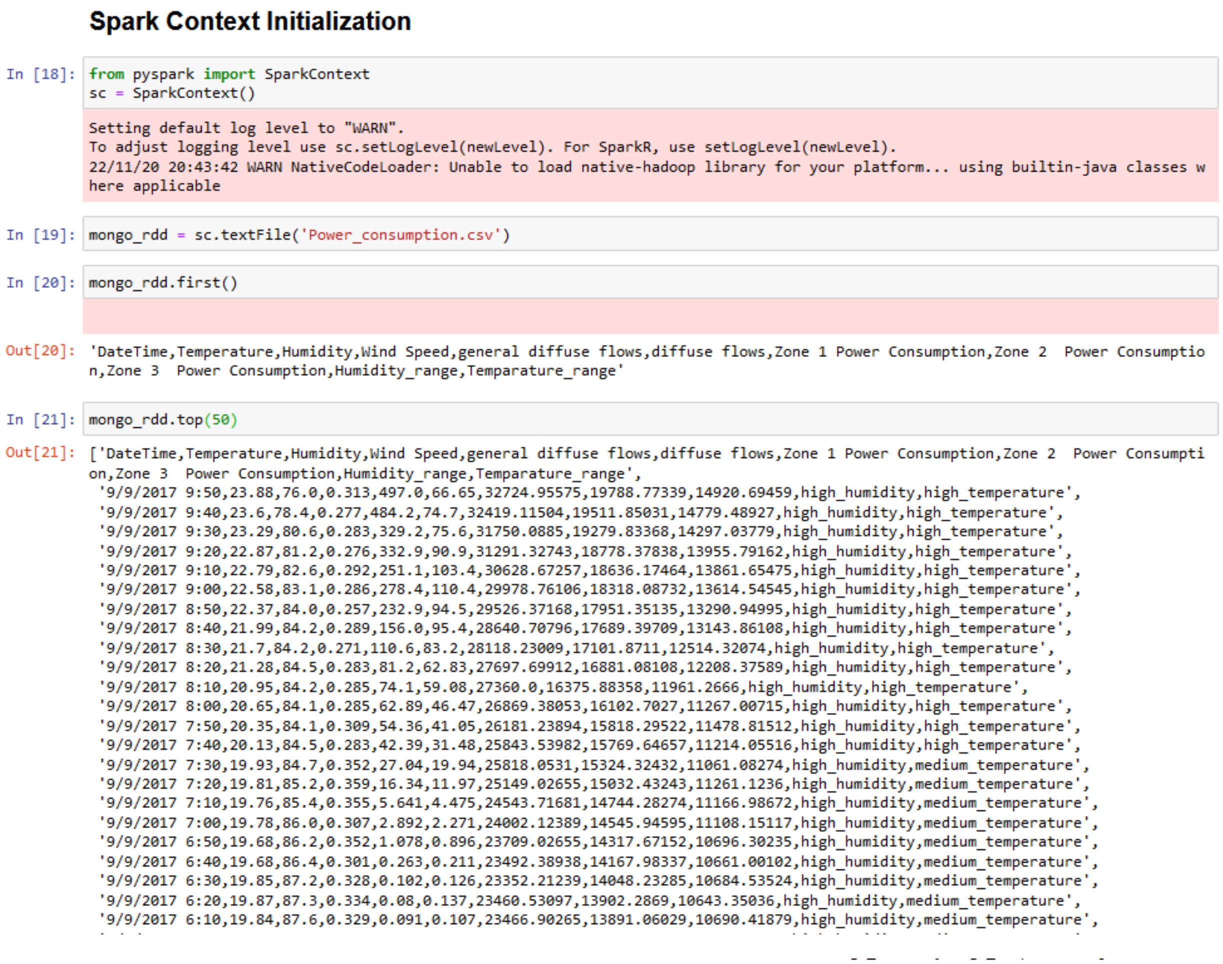
```
n [40]: main_data = main_data.withColumnRenamed("Wind Speed","wind_speed")
        main_data = main_data.withColumnRenamed("general diffuse flows","general_diffuse_flows")
        main_data = main_data.withColumnRenamed("diffuse flows","diffuse_flows")
        main_data = main_data.withColumnRenamed("Zone 1 Power Consumption","Zone_one_Power_Consumption")
        main_data = main_data.withColumnRenamed("Zone 2  Power Consumption","Zone_two_Power_Consumption")
        main_data = main_data.withColumnRenamed("Zone 3  Power Consumption","Zone_three_Power_Consumption")
```

```
n [41]: main_data.createOrReplaceTempView('table')
```

```
n [42]: main_data
```

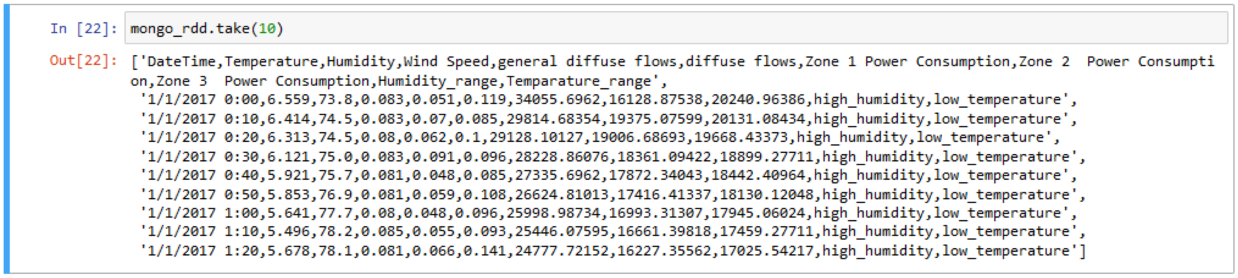
```
ut[42]: DataFrame[DateTime: string, Temperature: double, Humidity: double, wind_speed: double, general_diffuse_flows: double, diffuse_f
        lows: double, Zone_one_Power_Consumption: double, Zone_two_Power_Consumption: double, Zone_three_Power_Consumption: double, Hum
        idity_range: string, Temparature_range: string]
```

Spark distributed processing



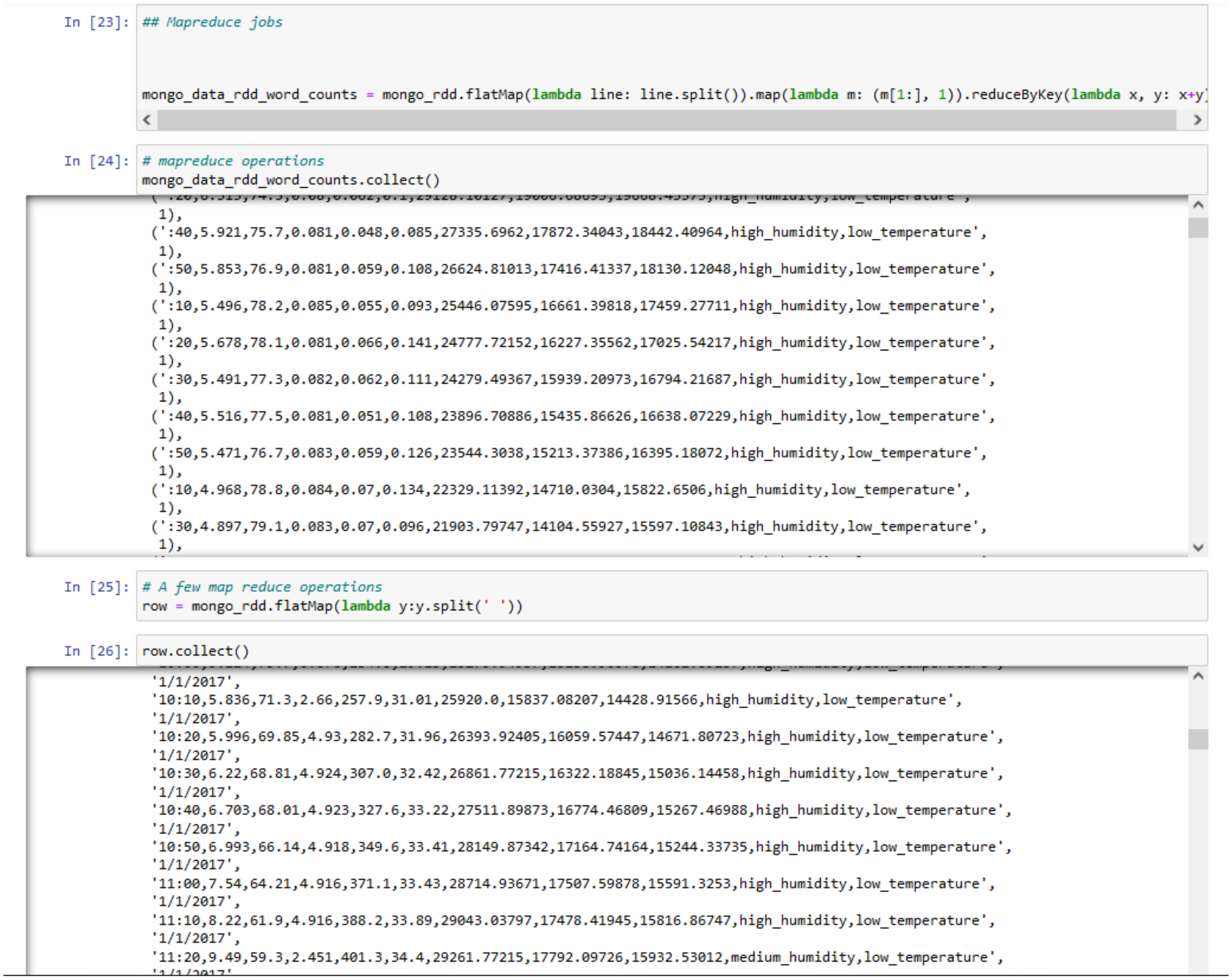
Spark Context Initialization

```
In [18]: from pyspark import SparkContext
         sc = SparkContext()

         Setting default log level to "WARN".
         To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
         22/11/20 20:43:42 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes w
         here applicable

In [19]: mongo_rdd = sc.textFile('Power_consumption.csv')

In [20]: mongo_rdd.first()

Out[20]: 'DateTime,Temperature,Humidity,Wind Speed,general diffuse flows,diffuse flows,Zone 1 Power Consumption,Zone 2  Power Consumptio
         n,Zone 3  Power Consumption,Humidity_range,Temparature_range'

In [21]: mongo_rdd.top(50)

Out[21]: ['DateTime,Temperature,Humidity,Wind Speed,general diffuse flows,diffuse flows,Zone 1 Power Consumption,Zone 2  Power Consumpti
         on,Zone 3  Power Consumption,Humidity_range,Temparature_range',
          '9/9/2017 9:50,23.88,76.0,0.313,497.0,66.65,32724.95575,19788.77339,14920.69459,high_humidity,high_temperature',
          '9/9/2017 9:40,23.6,78.4,0.277,484.2,74.7,32419.11504,19511.85031,14779.48927,high_humidity,high_temperature',
          '9/9/2017 9:30,23.29,80.6,0.283,329.2,75.6,31750.0885,19279.83368,14297.03779,high_humidity,high_temperature',
          '9/9/2017 9:20,22.87,81.2,0.276,332.9,90.9,31291.32743,18778.37838,13955.79162,high_humidity,high_temperature',
          '9/9/2017 9:10,22.79,82.6,0.292,251.1,103.4,30628.67257,18636.17464,13861.65475,high_humidity,high_temperature',
          '9/9/2017 9:00,22.58,83.1,0.286,278.4,110.4,29978.76106,18318.08732,13614.54545,high_humidity,high_temperature',
          '9/9/2017 8:50,22.37,84.0,0.257,232.9,94.5,29526.37168,17951.35135,13290.94995,high_humidity,high_temperature',
          '9/9/2017 8:40,21.99,84.2,0.289,156.0,95.4,28640.70796,17689.39709,13143.86108,high_humidity,high_temperature',
          '9/9/2017 8:30,21.7,84.2,0.271,110.6,83.2,28118.23009,17101.8711,12514.32074,high_humidity,high_temperature',
          '9/9/2017 8:20,21.28,84.5,0.283,81.2,62.83,27697.69912,16881.08108,12208.37589,high_humidity,high_temperature',
          '9/9/2017 8:10,20.95,84.2,0.285,74.1,59.08,27360.0,16375.88358,11961.2666,high_humidity,high_temperature',
          '9/9/2017 8:00,20.65,84.1,0.285,62.89,46.47,26869.38053,16102.7027,11267.00715,high_humidity,high_temperature',
          '9/9/2017 7:50,20.35,84.1,0.309,54.36,41.05,26181.23894,15818.29522,11478.81512,high_humidity,high_temperature',
          '9/9/2017 7:40,20.13,84.5,0.283,42.39,31.48,25843.53982,15769.64657,11214.05516,high_humidity,high_temperature',
          '9/9/2017 7:30,19.93,84.7,0.352,27.04,19.94,25818.0531,15324.32432,11061.08274,high_humidity,medium_temperature',
          '9/9/2017 7:20,19.81,85.2,0.359,16.34,11.97,25149.02655,15032.43243,11261.1236,high_humidity,medium_temperature',
          '9/9/2017 7:10,19.76,85.4,0.355,5.641,4.475,24543.71681,14744.28274,11166.98672,high_humidity,medium_temperature',
          '9/9/2017 7:00,19.78,86.0,0.307,2.892,2.271,24002.12389,14545.94595,11108.15117,high_humidity,medium_temperature',
          '9/9/2017 6:50,19.68,86.2,0.352,1.078,0.896,23709.02655,14317.67152,10696.30235,high_humidity,medium_temperature',
          '9/9/2017 6:40,19.68,86.4,0.301,0.263,0.211,23492.38938,14167.98337,10661.00102,high_humidity,medium_temperature',
          '9/9/2017 6:30,19.85,87.2,0.328,0.102,0.126,23352.21239,14048.23285,10684.53524,high_humidity,medium_temperature',
          '9/9/2017 6:20,19.87,87.3,0.334,0.08,0.137,23460.53097,13902.2869,10643.35036,high_humidity,medium_temperature',
          '9/9/2017 6:10,19.84,87.6,0.329,0.091,0.107,23466.90265,13891.06029,10690.41879,high_humidity,medium_temperature',
```



```
In [22]: mongo_rdd.take(10)

Out[22]: ['DateTime,Temperature,Humidity,Wind Speed,general diffuse flows,diffuse flows,Zone 1 Power Consumption,Zone 2  Power Consumpti
         on,Zone 3  Power Consumption,Humidity_range,Temparature_range',
          '1/1/2017 0:00,6.559,73.8,0.083,0.051,0.119,34055.6962,16128.87538,20240.96386,high_humidity,low_temperature',
          '1/1/2017 0:10,6.414,74.5,0.083,0.07,0.085,29814.68354,19375.07599,20131.08434,high_humidity,low_temperature',
          '1/1/2017 0:20,6.313,74.5,0.08,0.062,0.1,29128.10127,19006.68693,19668.43373,high_humidity,low_temperature',
          '1/1/2017 0:30,6.121,75.0,0.083,0.091,0.096,28228.86076,18361.09422,18899.27711,high_humidity,low_temperature',
          '1/1/2017 0:40,5.921,75.7,0.081,0.048,0.085,27335.6962,17872.34043,18442.40964,high_humidity,low_temperature',
          '1/1/2017 0:50,5.853,76.9,0.081,0.059,0.108,26624.81013,17416.41337,18130.12048,high_humidity,low_temperature',
          '1/1/2017 1:00,5.641,77.7,0.08,0.048,0.096,25998.98734,16993.31307,17945.06024,high_humidity,low_temperature',
          '1/1/2017 1:10,5.496,78.2,0.085,0.055,0.093,25446.07595,16661.39818,17459.27711,high_humidity,low_temperature',
          '1/1/2017 1:20,5.678,78.1,0.081,0.066,0.141,24777.72152,16227.35562,17025.54217,high_humidity,low_temperature']
```

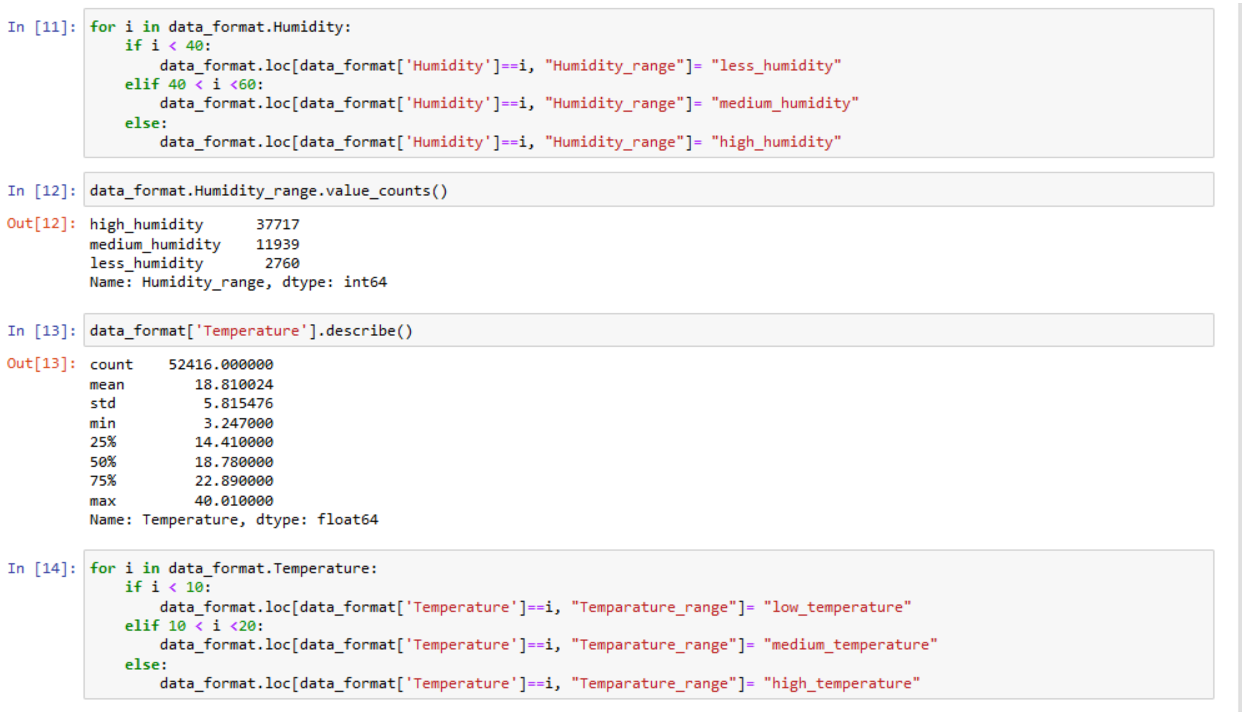
Mapreduce jobs steps



```
In [23]: ## Mapreduce jobs


         mongo_data_rdd_word_counts = mongo_rdd.flatMap(lambda line: line.split()).map(lambda m: (m[1:], 1)).reduceByKey(lambda x, y: x+y)

In [24]: # mapreduce operations
         mongo_data_rdd_word_counts.collect()

          (':20,5.915,74.5,0.08,0.062,0.1,29128.10127,19006.68693,19008.43373,high_humidity,low_temperature',
           1),
          (':40,5.921,75.7,0.081,0.048,0.085,27335.6962,17872.34043,18442.40964,high_humidity,low_temperature',
           1),
          (':50,5.853,76.9,0.081,0.059,0.108,26624.81013,17416.41337,18130.12048,high_humidity,low_temperature',
           1),
          (':10,5.496,78.2,0.085,0.055,0.093,25446.07595,16661.39818,17459.27711,high_humidity,low_temperature',
           1),
          (':20,5.678,78.1,0.081,0.066,0.141,24777.72152,16227.35562,17025.54217,high_humidity,low_temperature',
           1),
          (':30,5.491,77.3,0.082,0.062,0.111,24279.49367,15939.20973,16794.21687,high_humidity,low_temperature',
           1),
          (':40,5.516,77.5,0.081,0.051,0.108,23896.70886,15435.86626,16638.07229,high_humidity,low_temperature',
           1),
          (':50,5.471,76.7,0.083,0.059,0.126,23544.3038,15213.37386,16395.18072,high_humidity,low_temperature',
           1),
          (':10,4.968,78.8,0.084,0.07,0.134,22329.11392,14710.0304,15822.6506,high_humidity,low_temperature',
           1),
          (':30,4.897,79.1,0.083,0.07,0.096,21903.79747,14104.55927,15597.10843,high_humidity,low_temperature',
           1),

In [25]: # A few map reduce operations
         row = mongo_rdd.flatMap(lambda y:y.split(' '))

In [26]: row.collect()

          '1/1/2017',
          '10:10,5.836,71.3,2.66,257.9,31.01,25920.0,15837.08207,14428.91566,high_humidity,low_temperature',
          '1/1/2017',
          '10:20,5.996,69.85,4.93,282.7,31.96,26393.92405,16059.57447,14671.80723,high_humidity,low_temperature',
          '1/1/2017',
          '10:30,6.22,68.81,4.924,307.0,32.42,26861.77215,16322.18845,15036.14458,high_humidity,low_temperature',
          '1/1/2017',
          '10:40,6.703,68.01,4.923,327.6,33.22,27511.89873,16774.46809,15267.46988,high_humidity,low_temperature',
          '1/1/2017',
          '10:50,6.993,66.14,4.918,349.6,33.41,28149.87342,17164.74164,15244.33735,high_humidity,low_temperature',
          '1/1/2017',
          '11:00,7.54,64.21,4.916,371.1,33.43,28714.93671,17507.59878,15591.3253,high_humidity,low_temperature',
          '1/1/2017',
          '11:10,8.22,61.9,4.916,388.2,33.89,29043.03797,17478.41945,15816.86747,high_humidity,low_temperature',
          '1/1/2017',
          '11:20,9.49,59.3,2.451,401.3,34.4,29261.77215,17792.09726,15932.53012,medium_humidity,low_temperature',
          '1/1/2017'
```

Creating new features from existing data

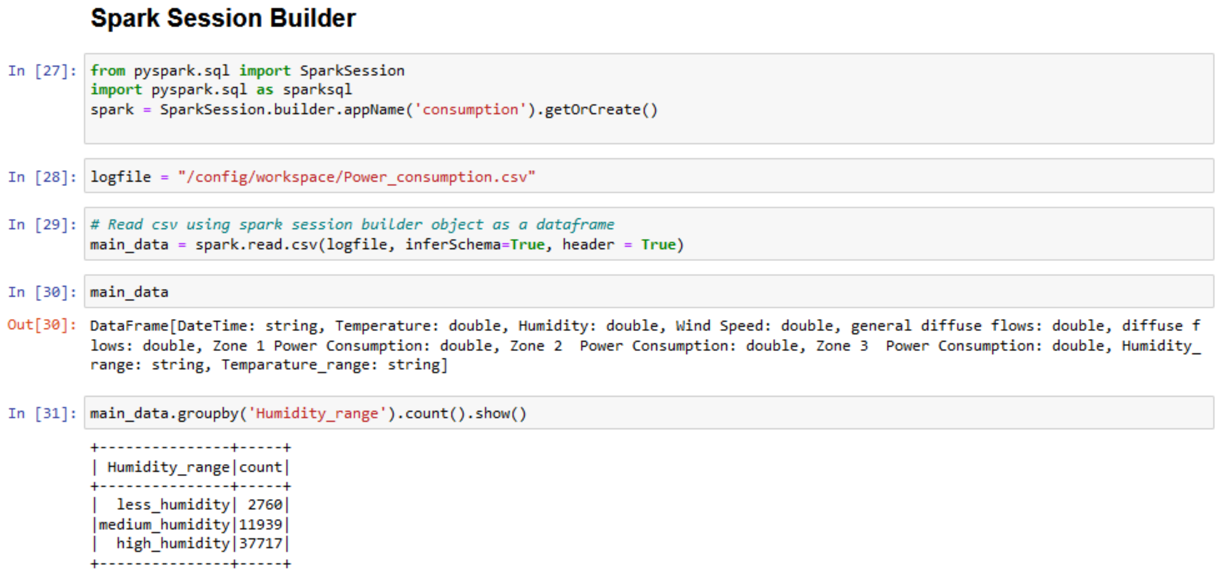
Humidity_range and Temperature_range are the two new attributes created from our power consumption dataset based on a threshold of humidity and temperature as shown in screenshot below.



```
In [11]: for i in data_format.Humidity:
             if i < 40:
                 data_format.loc[data_format['Humidity']==i, "Humidity_range"]= "less_humidity"
             elif 40 < i <60:
                 data_format.loc[data_format['Humidity']==i, "Humidity_range"]= "medium_humidity"
             else:
                 data_format.loc[data_format['Humidity']==i, "Humidity_range"]= "high_humidity"

In [12]: data_format.Humidity_range.value_counts()

Out[12]: high_humidity      37717
         medium_humidity    11939
         less_humidity       2760
         Name: Humidity_range, dtype: int64

In [13]: data_format['Temperature'].describe()

Out[13]: count    52416.000000
         mean        18.810024
         std          5.815476
         min          3.247000
         25%         14.410000
         50%         18.780000
         75%         22.890000
         max         40.010000
         Name: Temperature, dtype: float64

In [14]: for i in data_format.Temperature:
             if i < 10:
                 data_format.loc[data_format['Temperature']==i, "Temparature_range"]= "low_temperature"
             elif 10 < i <20:
                 data_format.loc[data_format['Temperature']==i, "Temparature_range"]= "medium_temperature"
             else:
                 data_format.loc[data_format['Temperature']==i, "Temparature_range"]= "high_temperature"
```

SPARK SESSION

Showing the count of humidity range



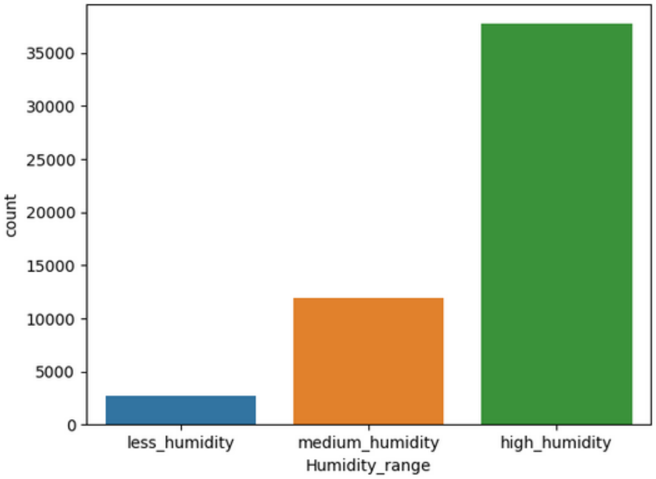
Spark Session Builder

```
In [27]: from pyspark.sql import SparkSession
         import pyspark.sql as sparksql
         spark = SparkSession.builder.appName('consumption').getOrCreate()

In [28]: logfile = "/config/workspace/Power_consumption.csv"

In [29]: # Read csv using spark session builder object as a dataframe
         main_data = spark.read.csv(logfile, inferSchema=True, header = True)

In [30]: main_data

Out[30]: DataFrame[DateTime: string, Temperature: double, Humidity: double, Wind Speed: double, general diffuse flows: double, diffuse f
         lows: double, Zone 1 Power Consumption: double, Zone 2  Power Consumption: double, Zone 3  Power Consumption: double, Humidity_
         range: string, Temparature_range: string]

In [31]: main_data.groupby('Humidity_range').count().show()

         +---------------+-----+
         | Humidity_range|count|
         +---------------+-----+
         |  less_humidity| 2760|
         |medium_humidity|11939|
         |  high_humidity|37717|
         +---------------+-----+
```

Plotting the humidity range

```
In [32]: humidity_plot = main_data.groupby('Humidity_range').count().toPandas()
```

```
In [33]: sns.barplot(data = humidity_plot, x = "Humidity_range", y = "count")
Out[33]: <AxesSubplot: xlabel='Humidity_range', ylabel='count'>
```



Showing the count based on temperature range

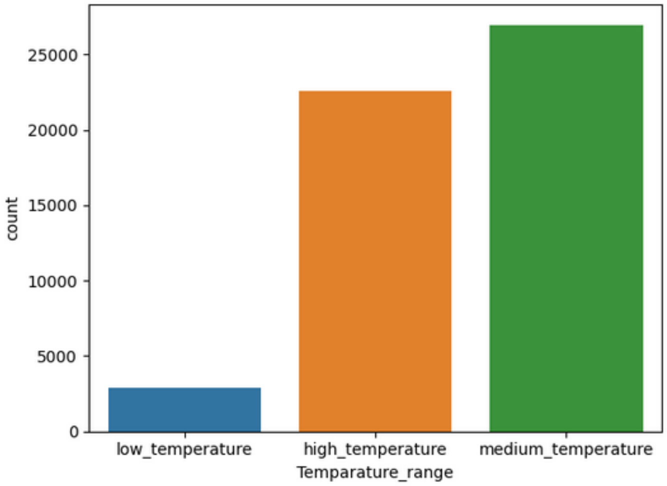
```
In [34]: temparature_plot = main_data.groupby('Temparature_range').count().toPandas()
```

```
In [35]: temparature_plot
Out[35]:
```

	Temparature_range	count
0	low_temperature	2874
1	high_temperature	22565
2	medium_temperature	26977

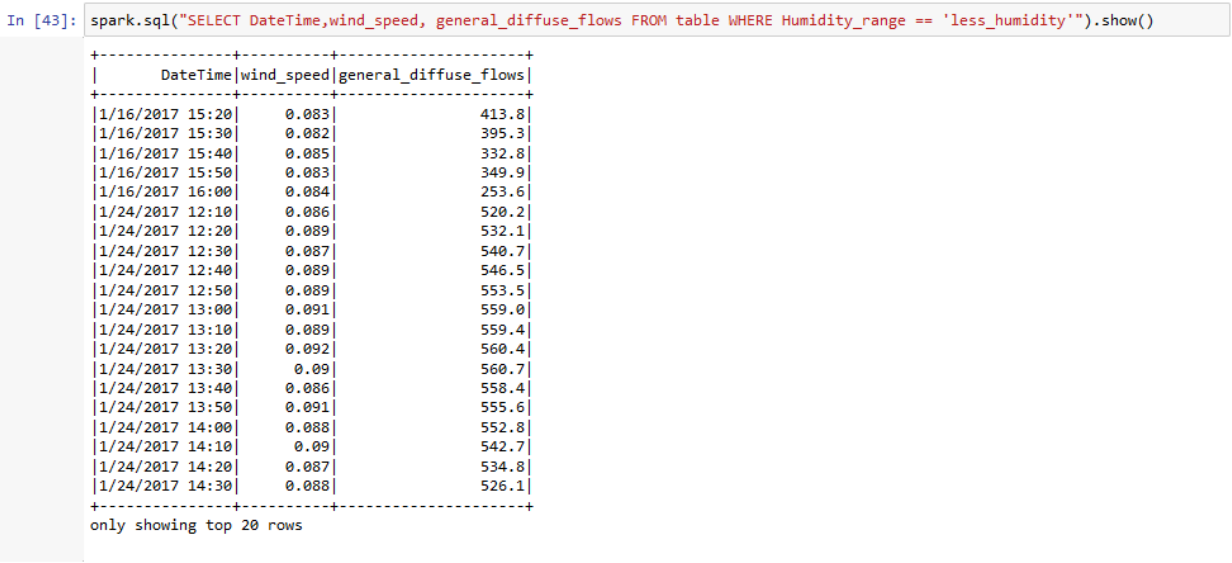
Plotting the temperature range

```
In [36]: sns.barplot(data = temparature_plot, x = "Temparature_range", y = "count")
Out[36]: <AxesSubplot: xlabel='Temparature_range', ylabel='count'>
```

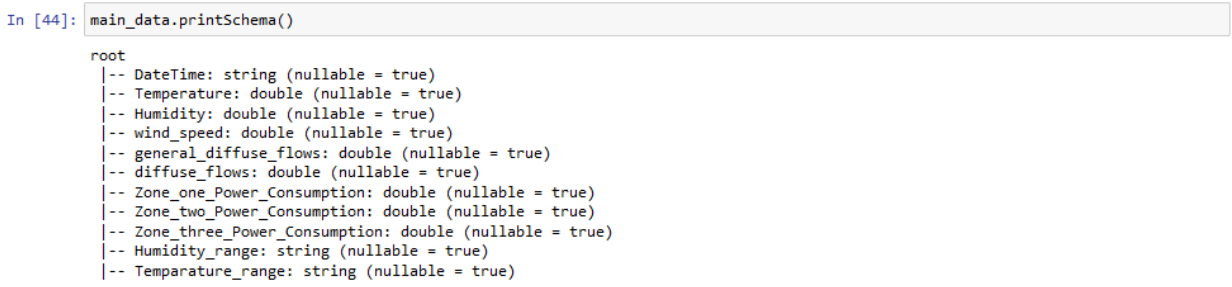


SPARK SQL

- Selecting the Date, wind_speed and diffusion flows based on humidity range and also printing the schema to show the datatypes of the attributes.



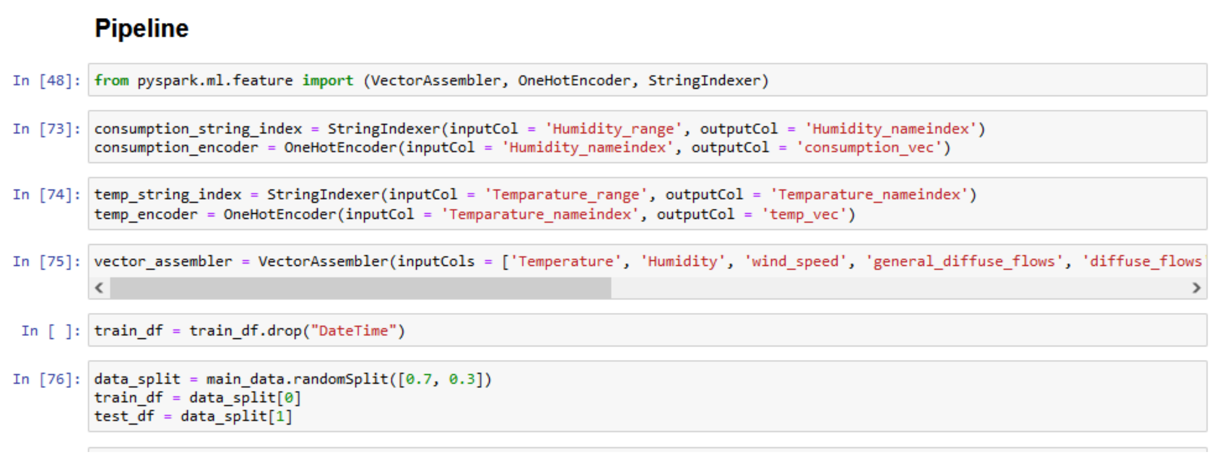
```
In [43]: spark.sql("SELECT DateTime,wind_speed, general_diffuse_flows FROM table WHERE Humidity_range == 'less_humidity'").show()

+---------------+----------+---------------------+
|       DateTime|wind_speed|general_diffuse_flows|
+---------------+----------+---------------------+
|1/16/2017 15:20|     0.083|                413.8|
|1/16/2017 15:30|     0.082|                395.3|
|1/16/2017 15:40|     0.085|                332.8|
|1/16/2017 15:50|     0.083|                349.9|
|1/16/2017 16:00|     0.084|                253.6|
|1/24/2017 12:10|     0.086|                520.2|
|1/24/2017 12:20|     0.089|                532.1|
|1/24/2017 12:30|     0.087|                540.7|
|1/24/2017 12:40|     0.089|                546.5|
|1/24/2017 12:50|     0.089|                553.5|
|1/24/2017 13:00|     0.091|                559.0|
|1/24/2017 13:10|     0.089|                559.4|
|1/24/2017 13:20|     0.092|                560.4|
|1/24/2017 13:30|      0.09|                560.7|
|1/24/2017 13:40|     0.086|                558.4|
|1/24/2017 13:50|     0.091|                555.6|
|1/24/2017 14:00|     0.088|                552.8|
|1/24/2017 14:10|      0.09|                542.7|
|1/24/2017 14:20|     0.087|                534.8|
|1/24/2017 14:30|     0.088|                526.1|
+---------------+----------+---------------------+
only showing top 20 rows
```



```
In [44]: main_data.printSchema()

root
 |-- DateTime: string (nullable = true)
 |-- Temperature: double (nullable = true)
 |-- Humidity: double (nullable = true)
 |-- wind_speed: double (nullable = true)
 |-- general_diffuse_flows: double (nullable = true)
 |-- diffuse_flows: double (nullable = true)
 |-- Zone_one_Power_Consumption: double (nullable = true)
 |-- Zone_two_Power_Consumption: double (nullable = true)
 |-- Zone_three_Power_Consumption: double (nullable = true)
 |-- Humidity_range: string (nullable = true)
 |-- Temparature_range: string (nullable = true)
```

BUILDING THE PIPELINE

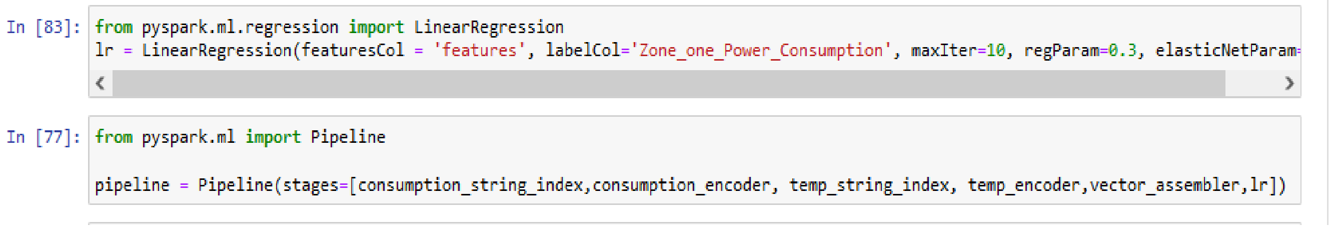
Using the one-hot encoder, string indexer and Vector Assembler. Splitting the data in training: 70% and testing: 30%



Pipeline

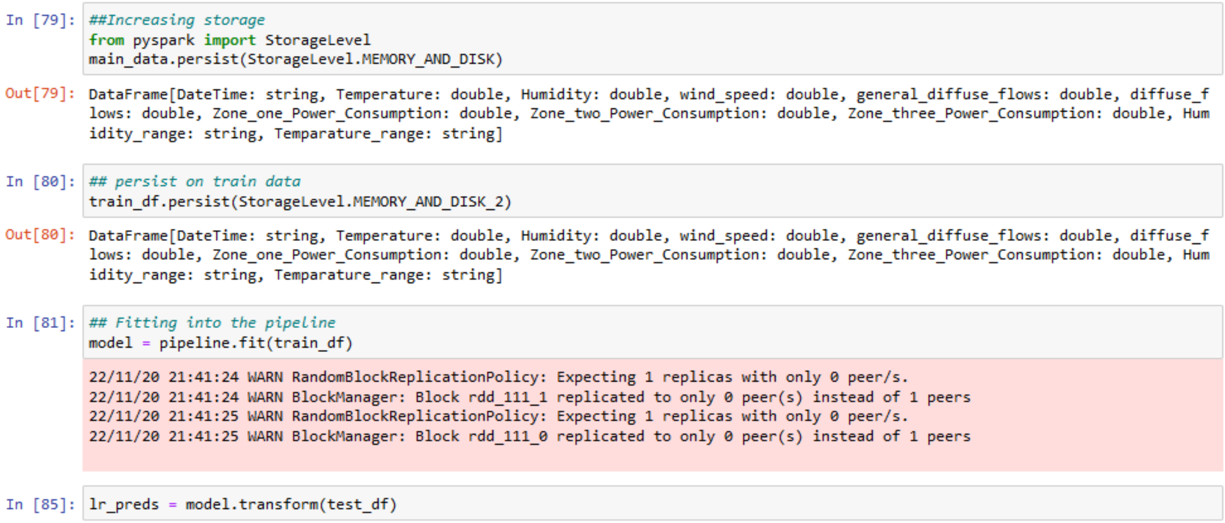
```
In [48]: from pyspark.ml.feature import (VectorAssembler, OneHotEncoder, StringIndexer)

In [73]: consumption_string_index = StringIndexer(inputCol = 'Humidity_range', outputCol = 'Humidity_nameindex')
         consumption_encoder = OneHotEncoder(inputCol = 'Humidity_nameindex', outputCol = 'consumption_vec')

In [74]: temp_string_index = StringIndexer(inputCol = 'Temparature_range', outputCol = 'Temparature_nameindex')
         temp_encoder = OneHotEncoder(inputCol = 'Temparature_nameindex', outputCol = 'temp_vec')

In [75]: vector_assembler = VectorAssembler(inputCols = ['Temperature', 'Humidity', 'wind_speed', 'general_diffuse_flows', 'diffuse_flows

In [ ]: train_df = train_df.drop("DateTime")

In [76]: data_split = main_data.randomSplit([0.7, 0.3])
         train_df = data_split[0]
         test_df = data_split[1]
```

Calling the Linear regression model from pyspark.ml.regression package and fitting it into the Pipeline.



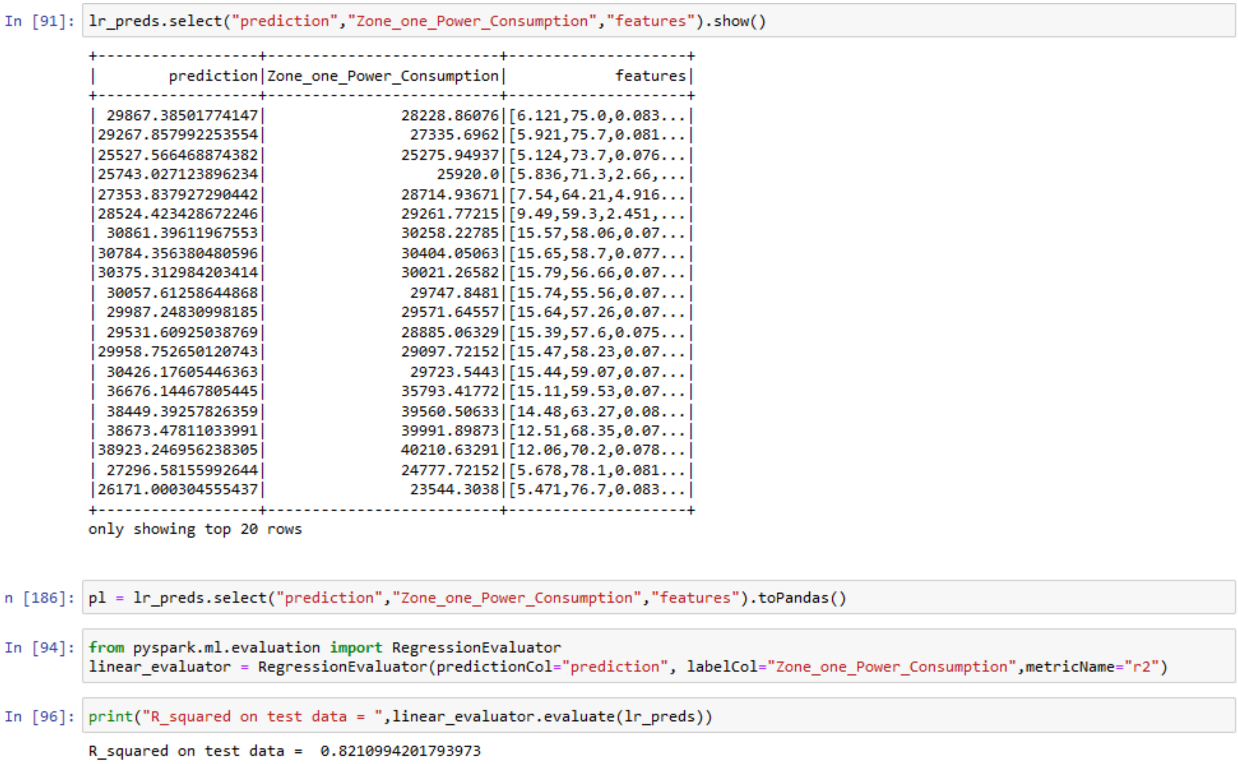
```
In [83]: from pyspark.ml.regression import LinearRegression
         lr = LinearRegression(featuresCol = 'features', labelCol='Zone_one_Power_Consumption', maxIter=10, regParam=0.3, elasticNetParam=

In [77]: from pyspark.ml import Pipeline

         pipeline = Pipeline(stages=[consumption_string_index,consumption_encoder, temp_string_index, temp_encoder,vector_assembler,lr])
```

Increasing the storage level of spark



```
In [79]: ##Increasing storage
         from pyspark import StorageLevel
         main_data.persist(StorageLevel.MEMORY_AND_DISK)

Out[79]: DataFrame[DateTime: string, Temperature: double, Humidity: double, wind_speed: double, general_diffuse_flows: double, diffuse_f
         lows: double, Zone_one_Power_Consumption: double, Zone_two_Power_Consumption: double, Zone_three_Power_Consumption: double, Hum
         idity_range: string, Temparature_range: string]

In [80]: ## persist on train data
         train_df.persist(StorageLevel.MEMORY_AND_DISK_2)

Out[80]: DataFrame[DateTime: string, Temperature: double, Humidity: double, wind_speed: double, general_diffuse_flows: double, diffuse_f
         lows: double, Zone_one_Power_Consumption: double, Zone_two_Power_Consumption: double, Zone_three_Power_Consumption: double, Hum
         idity_range: string, Temparature_range: string]

In [81]: ## Fitting into the pipeline
         model = pipeline.fit(train_df)

         22/11/20 21:41:24 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
         22/11/20 21:41:24 WARN BlockManager: Block rdd_111_1 replicated to only 0 peer(s) instead of 1 peers
         22/11/20 21:41:25 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
         22/11/20 21:41:25 WARN BlockManager: Block rdd_111_0 replicated to only 0 peer(s) instead of 1 peers

In [85]: lr_preds = model.transform(test_df)
```

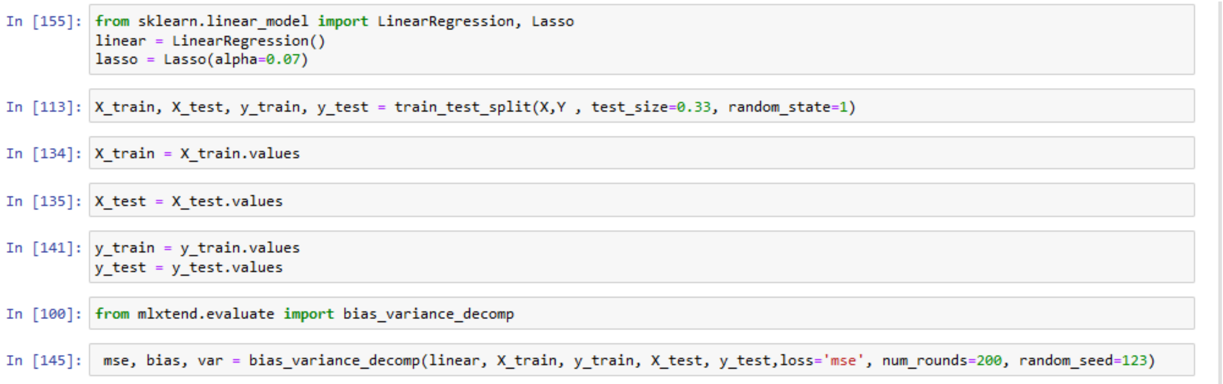
Prediction results and accuracy score



```
In [91]: lr_preds.select("prediction","Zone_one_Power_Consumption","features").show()

         +------------------+--------------------------+--------------------+
         |        prediction|Zone_one_Power_Consumption|            features|
         +------------------+--------------------------+--------------------+
         | 29867.38501774147|               28228.86076|[6.121,75.0,0.083...|
         |29267.857992253554|                27335.6962|[5.921,75.7,0.081...|
         |25527.566468874382|               25275.94937|[5.124,73.7,0.076...|
         |25743.027123896234|                   25920.0|[5.836,71.3,2.66,...|
         |27353.837927290442|               28714.93671|[7.54,64.21,4.916...|
         |28524.423428672246|               29261.77215|[9.49,59.3,2.451,...|
         | 30861.39611967553|               30258.22785|[15.57,58.06,0.07...|
         |30784.356380480596|               30404.05063|[15.65,58.7,0.077...|
         |30375.312984203414|               30021.26582|[15.79,56.66,0.07...|
         | 30057.61258644868|                29747.8481|[15.74,55.56,0.07...|
         | 29987.24830998185|               29571.64557|[15.64,57.26,0.07...|
         | 29531.60925038769|               28885.06329|[15.39,57.6,0.075...|
         |29958.752650120743|               29097.72152|[15.47,58.23,0.07...|
         | 30426.17605446363|                29723.5443|[15.44,59.07,0.07...|
         | 36676.14467805445|               35793.41772|[15.11,59.53,0.07...|
         | 38449.39257826359|               39560.50633|[14.48,63.27,0.08...|
         | 38673.47811033991|               39991.89873|[12.51,68.35,0.07...|
         |38923.246956238305|               40210.63291|[12.06,70.2,0.078...|
         | 27296.58155992644|               24777.72152|[5.678,78.1,0.081...|
         |26171.000304555437|                23544.3038|[5.471,76.7,0.083...|
         +------------------+--------------------------+--------------------+
         only showing top 20 rows

n [186]: pl = lr_preds.select("prediction","Zone_one_Power_Consumption","features").toPandas()

In [94]: from pyspark.ml.evaluation import RegressionEvaluator
         linear_evaluator = RegressionEvaluator(predictionCol="prediction", labelCol="Zone_one_Power_Consumption",metricName="r2")

In [96]: print("R_squared on test data = ",linear_evaluator.evaluate(lr_preds))

         R_squared on test data =  0.8210994201793973
```

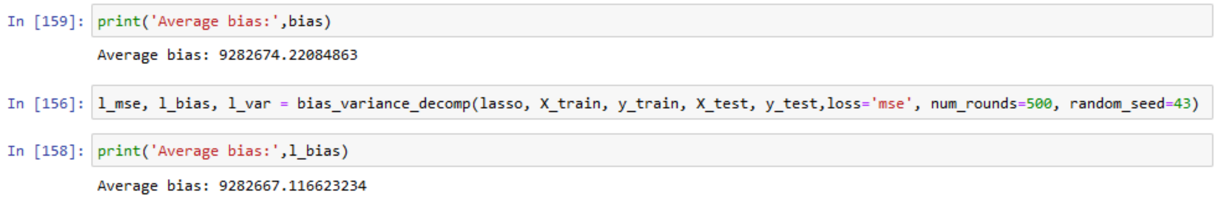
AI FAIRNESS

AI fairness on our dataset to check for bias in the data using the bias_variance_decomposition library



```
In [155]: from sklearn.linear_model import LinearRegression, Lasso
          linear = LinearRegression()
          lasso = Lasso(alpha=0.07)

In [113]: X_train, X_test, y_train, y_test = train_test_split(X,Y , test_size=0.33, random_state=1)

In [134]: X_train = X_train.values

In [135]: X_test = X_test.values

In [141]: y_train = y_train.values
          y_test = y_test.values

In [100]: from mlxtend.evaluate import bias_variance_decomp

In [145]:  mse, bias, var = bias_variance_decomp(linear, X_train, y_train, X_test, y_test,loss='mse', num_rounds=200, random_seed=123)
```

Bias removal values before and after decomposition



```
In [159]: print('Average bias:',bias)

          Average bias: 9282674.22084863

In [156]: l_mse, l_bias, l_var = bias_variance_decomp(lasso, X_train, y_train, X_test, y_test,loss='mse', num_rounds=500, random_seed=43)

In [158]: print('Average bias:',l_bias)

          Average bias: 9282667.116623234
```

4. DISCUSSIONS

Loaded the dataset Tétouan city power consumption into MongoDB, in “i535_final_project” database. Using spark context, we read the file in the spark rdd and performed few map-reduce jobs to check for word count. Using the groupby function of spark we grouped the humidity range and temperature range and found the count of high humidity is maximum, whereas the count of medium temperature is maximum. From **Fig.1** we can infer that the points where the color is darker humidity was more, and 38000 units of power was consumed when temperature hit maximum i.e., 40.

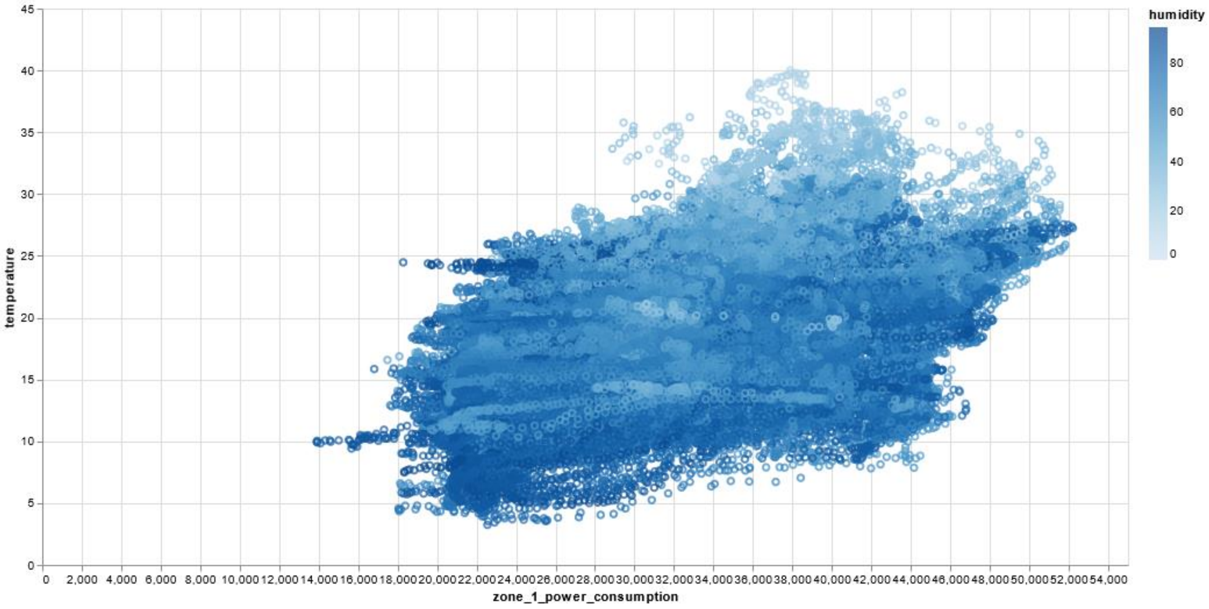


Fig.1: Zone one power consumption

From **Fig.2** we can see that humidity is high where temperature was low. And the points are darker when humidity is low and temperature is high and we can see that the power consumption is 36000 where humidity is 60.

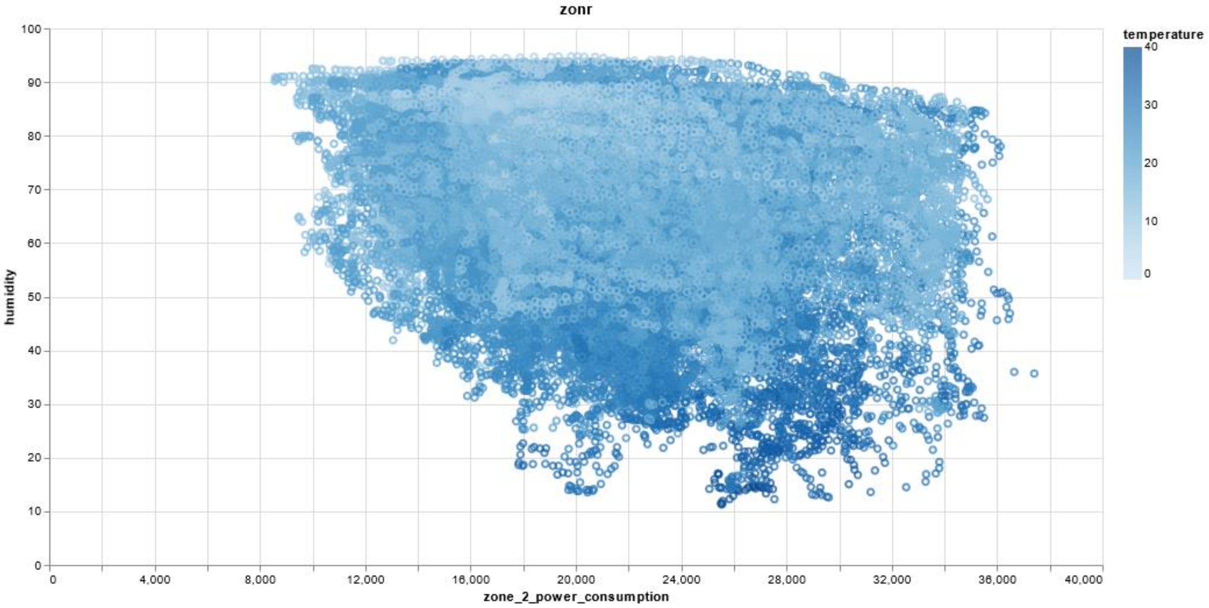


Fig.2: Zone two power consumption

Additionally, for zone two we can see the humidity range and their range of power consumption.

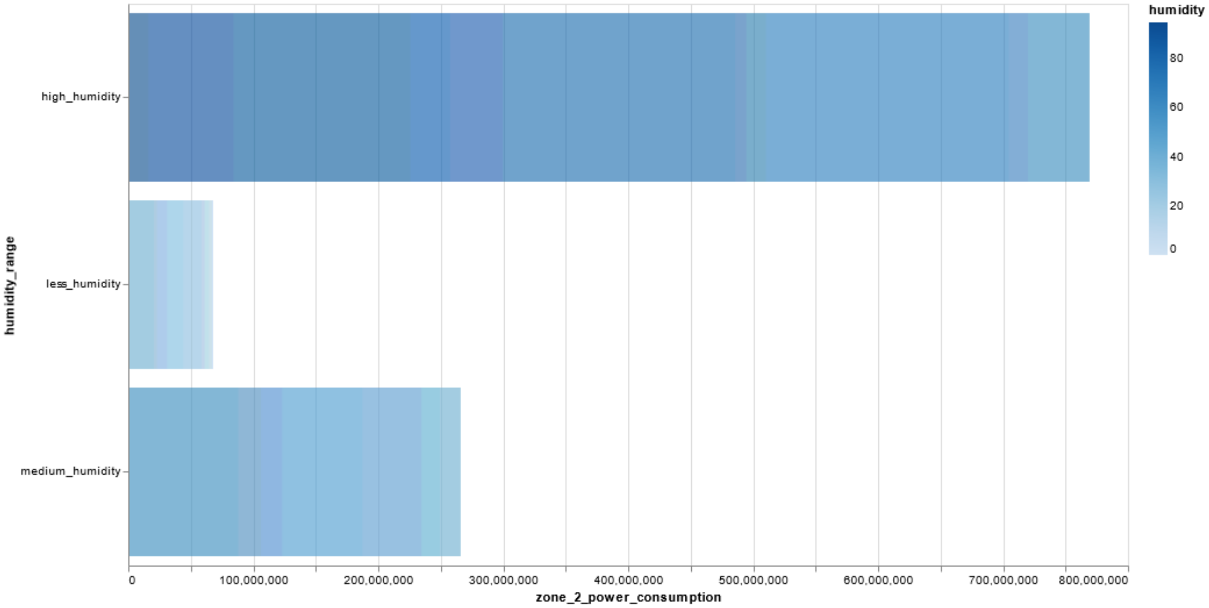
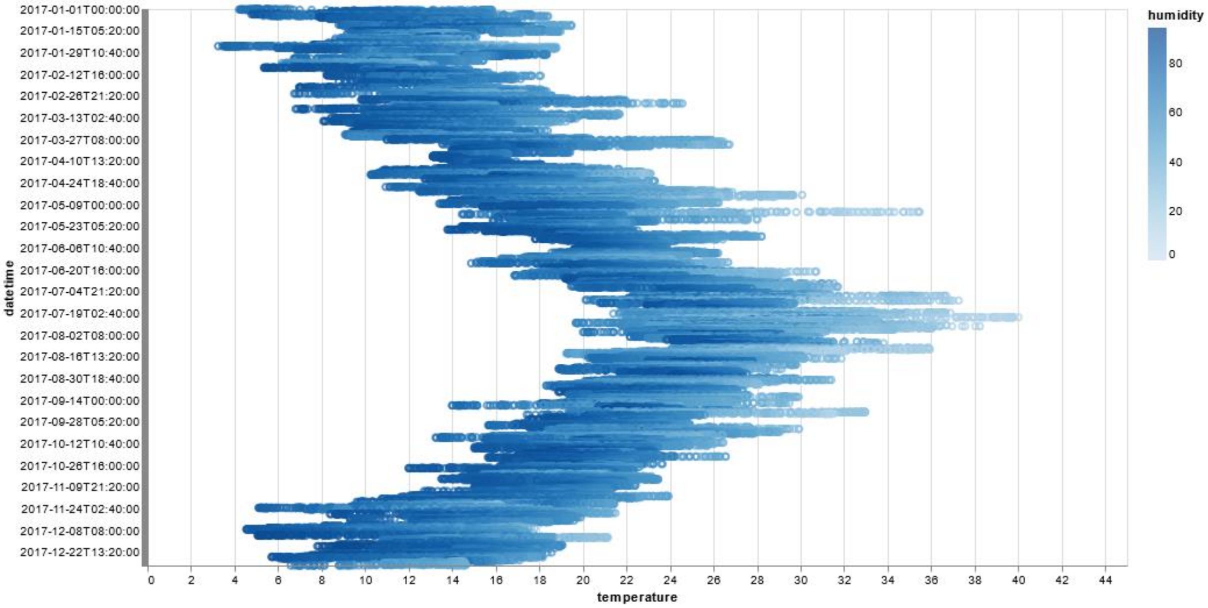


Fig.3 Zone two power consumption vs Humidity range

To see the temperature variations for a year we tried to verify the data and saw that in the month of January the temperature is low and it increases in June and July, and decreases as it approaches December. As it’s a Time series data we can see a trend and a seasonality.



As the data didn’t have any null values and the datatypes were proper, we went ahead and built a pipeline. We used the string indexer of pyspark to map our “humidity_range” and “temperature_range” columns, which are categorical column to map to ML column of label indices. Post that we used the one hot encoder to encode the categorical column and used the vector assembler on our final features. Following we defined a Linear Regression model and passed the model and features to our pipeline.

We predicted the values for zone one, and it gave us a good prediction result with r2 score of 82%.

As my data didn’t have any privileged category or protected category, I removed the bias for the AI Fairness part. By using the “bias_variance_decomposition” of scikit learn library I removed the bias form the entire data. Using lasso regularization, the bias decrement is achieved.

BARRIERS/PROBLEM FACED

I had a problem with my VM, where my VM was crashing. The stability was achieved by closing Firefox and restarting it. The MongoDB wasn’t running, then I restarted using my credentials, which worked fine. While working with my dataset, the SQL queries weren't working due to the spaces in the column names. Once I changed the column names using Spark, it worked fine. I visualized the data using Chart builder, which was very slow, but unfortunately there was no workaround it. So, visualizations took some time.

SKILLS FROM THIS COURSE

Virtualization: I used Jetstream to create VM and finish my project in ubuntu.

Ingest and Storage: I used MongoDB as my storage to store my data.

Processing and Analytics: My analysis was Spark based. So, I used Spark RDD, Spark SQL, MapReduce operations and data preprocessing using Spark

Lifecycles and Pipelines: I created a Pipeline and fit my prediction model into it.

5. CONCLUSION

In this project I implemented a lot of things, right from storing data into a NoSQL database. I also created two features which gave us the range of humidity and temperature, based on which we saw the power consumption of all the three zones. I have predicted the power consumption of zone one and other prediction for zone two and three can be determined and based on natural occurrences of temperature and, wind speed, humidity etc. they can predict the overall power consumption which provides intelligence utilities and helps them to improve their systems' performance in terms of productivity and effectiveness.

6. REFERENCES

[1] **A Comparison Study of Machine Learning Methods for Energy Consumption Forecasting in Industry** Mouad Bahij, M. Labbadi, M. Cherkaoui, Chakib Chatri, S. Lakrit

[2] **https://medium.com/analytics-vidhya/calculation-of-bias-variance-in-python-8f96463c8942**

[3]**https://archive.ics.uci.edu/ml/datasets/Power+consumption+of+Tetouan+city#**

[4]https://sparkbyexamples.com/pyspark-rdd/

[5]https://spark.apache.org/docs/3.1.1/api/python/reference/api/pyspark.SparkContext.html#pyspark.SparkContext