School of Computing
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

Leeds Historical Weather Report

Mithilesh Kumar Ramireddy

Submitted in accordance with the requirements for the degree of MSc Advanced Computer Science

2021/2022

The candidate confirms that the following have been submitted*:*

Items	**Format**	**Recipient(s) and Date**
Deliverables 1	*Report*	*SSO (19/08/2022)*
Deliverables 2	*Azure Account Access*	*Supervisor, assessor (19/08/2022)*
Deliverable 3	*PowerBi Report*	*Supervisor, assessor (19/08/2022)*

Type of Project: Exploratory Software

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) R Mithilesh Kumar

Summary

Using Open Weather data to develop PowerBi reports that can be helpful for users and organisations. Climate change has been a major concern in modern times. Due to the rise of industrialisation and increased pollutants in the atmosphere, researchers are trying to derive results to reduce the pollutants and keep the surroundings a lot cleaner and sustainable for future generation. This report gives a detailed analysis on how the weather has been for the past 12 years. To tag along, the pollution API is also used for the past 2 years to provide how these chemicals that are released in the air are impacting the area.

The location chosen to demonstrate this project is the city of Leeds, West Yorkshire, UK. These reports are designed to provide a historical analysis on how the weather has performed the weather datasets and pollution datasets are JSON format therefore must have been through a data warehousing platform to be finally settle at a database and use them for reporting. In a technical side this project is to explore the field of data warehousing and on the application side, it is to explore the field of weather and pollution APIs to find out its applications.

Acknowledgements

I would like to express my gratitude to God almighty for driving me in the right direction and providing me the strength and patience to resolve and overcome any task put in front of me. I would like to thank my parents for giving me this opportunity to pursue my masters at the University of Leeds. I would like to thank my uncle for their hospitality.

I would like to extend my gratitude to my supervisor Dr.Sebastian Ordyniak , who was really helpful and supportive regarding the domain that I’ve chosen for my project. He has given me great ways to provide fitting solutions for the topic that I’ve chosen. I’m extremely thankful to my assessor Dr.Kristina Vuskovic who helped by providing valuable feedback for my project.

Table of Contents

Chapter 1

Introduction

1.1 Context:

Weather data can be used for a wide range for applications. In most situation it is open to the public data to be glanced and take a forecast what that day/week is going to look like. By using APIs analysis from the past can be made to crack down problems that involve global warming. The API on its own does not hold much value but once these APIs are used as for a new product/service development where every person in the city can use the report to make informed decisions.

1.2 Problem Statement:

Up to now, weather was only used a tool that was viewed daily and forgotten to the public. This report helps in creating awareness to the resident of Leeds regarding weather and pollution. The importance is to visualise how these pollutants are affecting the weather, daily life and what plans must be made to overcome it. The reports give a brief history of how the weather has been from 2010- July 2020 in the city of Leeds. By applying filters, some fluctuations can be noticed which can used as an alert. The pollution dataset is only for the year 2021 and half of 2022. This dataset helps puts out levels for each pollutant on an hourly basis. This helps in noticing when the levels have spiked even by using simple conditional formatting. Result of this project would create awareness and help in forecasting the next day based on the values projected from 2010-2022. Pollution report is to create awareness amongst to public to notice which levels to operate at.

1.3 Project aim:

The aim of the project is to analyse different geographical variables in the UK jurisdiction. This is focused on two segments i.e., weather and pollution. The final report will enable interaction between these variables and provide the best user experience possible to obtain the historical data and the forecasts based on the given data. This project opens a wide array of possibilities as it can be used in the public sector, educational sector and on a single user level too. To produce the exploratory software, the project will use APIs from an open-source website called OpenWeather.org. The geolocation is only fixed to the city of Leeds for this project. The datasets are used to forecast and analyse on the past readings provided. The PowerBi report will be divided into 2 different segments which again contain a multiple set of pages which go through the different environmental variables that give a walkthrough of what factors are used on for the graphs. The software will be later be evaluated based on integrity, effectiveness, and usage scenarios.

1.4 Proposed solution:

To achieve the aim, the proposed exploratory software is developed to display the historical weather and pollution data of the city of Leeds. This application would enable to view all the variables through time and other forms of aggregation. The merit of using PowerBi is that it as interactive capabilities by helping to filter and highlight the necessary visual. The solution will allow to evaluate positives and limitations for the data present.

1.5 Degree of relevance:

The skills that will be required to execute this project are based on three different modules from the Advanced Computer Science MSc course. Data Science (COMP5122M) was used for visualisation skills and dealing with data with a few outliers and data preparations. These skillsets help in understanding the quality of the data, investigate and resolve any issues before the data is being used. Tableau was thought as a visualisation tool and similarly PowerBi will be used for this project. Programming for Data Science (COMP5721M) module teaches function and packages were taught. Accepting data from multiple formats (csv, xlsx and JSON) and converting them into data frames and using regular expressions, and other string manipulation techniques for data analysis where some of the tasks that will be used for this project. Cloud Computing (COMP5850M) had thought about various resources present in the cloud to use as services or platforms. Resource group management and security features were specifically highlighted to ensure safety and integrity.

Chapter 2

Project Planning

This chapter goes through all of action steps for the project. At first there will be a list of tasks that are sequentially listed in the form of events regarding the ways taken to approach the end goal. This will be followed by research methodology which gives an insight on what principles will be used and the influences for the work. Lastly, there will be a risk assessment section that explains the evaluations and limitations of this project.

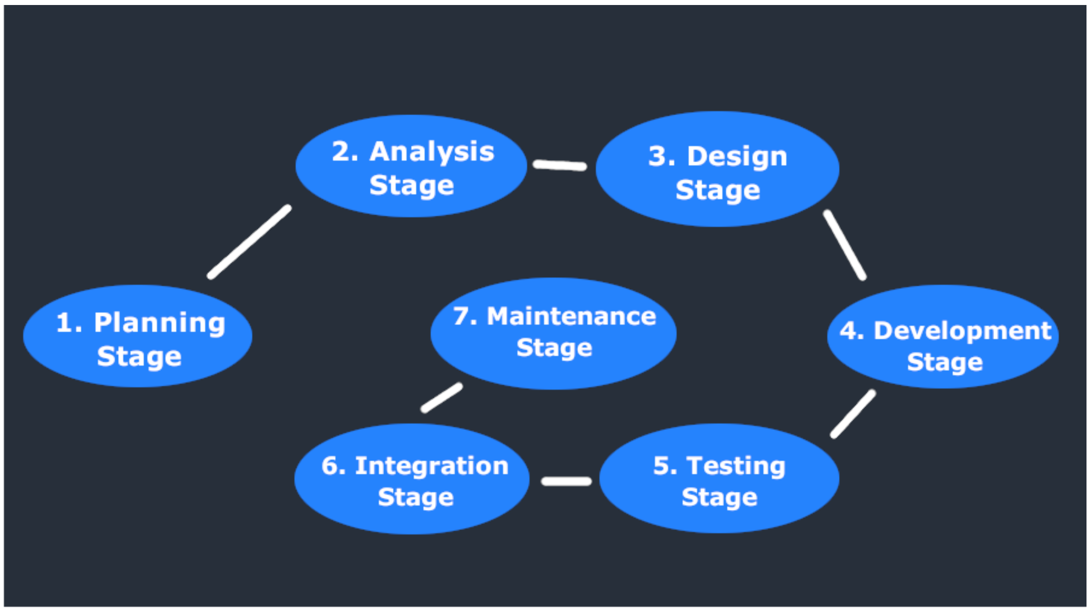


Figure 3.1 *Desired approach for Software Development Life Cycle [12]*

The report is represented in the form of chapters which is very close to the approach taken to the development of this project (Figure 3.1).

Planning Stage

This stage goes through the list of tasks that are taken to approach this project. It also includes to find a motive behind the project and how it would benefit a certain group of people.

Analysis Stage

Business requirements are taken down for this process to establish an understanding to move on to the business stage. The model chosen to be worked on isn't waterfall as there will be change in functional and non-functional requirements after every project review. If the system requirements and the datasets are chosen and complete, then the software design stage comes after. This stage also includes the data understanding stage where the logic behind the 2 separate APIs is to be understood. The approach taken for this life cycle is a spiral model. The weather API includes multiple fields that have to do with temperature, pressure, rain, snow, humidity, etc. The Pollution API just produces a list of values that throw co, no, no2, so2, pm2_5 and pm2_10 values.

Design Stage

After the business understandings are understood and analysis is done what software must be used, the design stage is when the software is put into use to see if the system configurations are compatible with each other. Since this is a project in the data engineering domain, a data warehouse outline must be designed from start to finish to fix to an approach. The data preparation steps must be assigned in this stage as well

Development or Implementation Stage

The cloud technologies need to be integrated with each other to ensure that the dataflow and transformation is possible. Using Databricks and some additional libraries the JSON file will be converted into a CSV. Datafactory is implemented by assigning source and sink (database) and run serverless function to trigger the pipeline.

Integration, Testing and Maintenance stage

All these topics come under the system user testing. The testing phase was split was split into two sections where the experienced professionals tested the data warehouse, and some users tested the front end application.

2.1 Project Tasks

The following tasks are taken for the execution of the project.

1. Read research papers on data warehousing and network security. Read what data variables for weather and pollution would go for analysis.
2. Register to openweather.org and pick a certain set of APIs that help in deriving weather reports.

3. Go through tutorials and read Microsoft documentation about data factories and blob storage accounts.
4. Download Bulk APIs or call the APIs by request.
5. Create and SQL server and a database and configure the database of the local system
6. Configure data factory and storage account and create linked services between them and connect to the database.
7. Develop a system which ensures the travel of information from the storage account to the sql database
8. Use PowerBi to visualise and analyse by projecting useful information
9. Validate the impact of the system from non-experienced users and experienced professionals.
10. Run test cases after every iteration in the developed product.

2.2 Deliverables

At the end of the project, the PowerBi report will be presented along with the final report. The interactive reports show how the data is split through time in general. There will a weather segment and a pollution segment and further drill downs will contain analysis of how the pollution has impacted the change in weather. The report will create awareness amongst the viewers about environmental impacts caused by mankind and weather patterns that stand out during the twelve-year period (2010-present).

2.3 Risk Assessments

- Couldn’t obtain solar radiation to get further analysis on the weather as it was expensive service.
- Spark Cluster isn’t available for long term use for calling APIs as the project uses as Azure student subscription.

Chapter 3

Background Research

This chapter gives a deep focus on the data-warehousing, the old technologies that were associated with it and the new technologies and their advantages. It also discusses the architecture and the general principles behind this solution. The Microsoft documentation regarding the services used and openweather documentation regarding the APIs will be in this section as well. General Data Protection Regulation (GDPR) is a major concern due to the rise in cyber crimes and data privacy. The principles for GDPR will be discussed to ensure that the data used for this project is open sourced and isn't violating any policy regarding national security and safety of its citizens.

3.1 Data Warehousing

Data warehousing an OLAP (on-line analytical processing) are significant features that have become a point of focus in the field of databases. These features are responsible for the flow of data across different platforms and provide storing and transformation solutions which can be helpful of reporting and processing of huge datasets. These enable in making quick decisions based on real time data that benefit the business. There has been an explosive growth in this domain for the past couple of years and various tools have emerged to support this mechanism. These tools are maintained separately from the organisation's database. One of the main reasons why this cant be treated in the same domain is data warehouse supports on-line analytical processing whereas a database supports on-line transaction processing. Data warehouses are targeted to derive historical, summarised, and consolidated information to make the best business decisions possible.

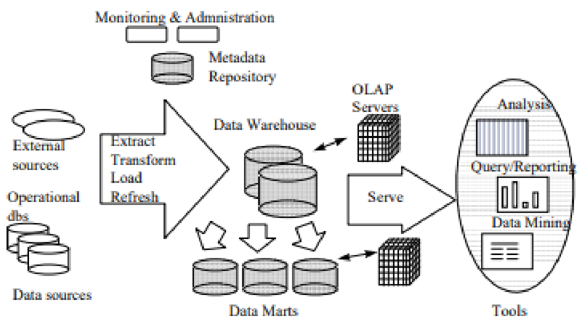


Figure 3.1 *Date Warehouse System Architecture* ***[1]***

The system architecture gives a detailed overview of the design and what services take part in the

3.2 Microsoft Docs

Docs provides the documentation and solutions to all of the services provided to every developer, end user and Professional by Microsoft. It goes through various topics like articles, technical specifications, guides, and tutorials. For this project, docs are used as a guide for every service that was used in the project. The list of software used are

1. Azure Databricks - To help in mounting to the storage accounts and getting APIs
2. Azure storage account - Configure and choosing the right specification
3. Azure SQL server - Configuring database and following an ideal schema for the project.
4. Azure Key vault – To assign access control to the storage account for the databricks service.
5. PowerBi – Helping in applying DAX functions, importing new libraries and advice the best visualisation for graphs.

3.3 Open Weather Map

The online service provides weather data at any region on the planet via APIs. These APIs can be used for energy supply and demand and minimum and maximum risks for efficient farming practices. The website also provides a road risk API which helps in effectively planning routes for fuel consumption. A lot of the APIs used can be retrieved through API keys which give an hourly, minute or week wise forecast for the features offered. The services that will be used for this project are History Bulk API for weather and Pollution API.

3.3 General Data Protection Regulation

When dealing with data with any size or magnitude it is a responsibility to ensure that the personal data used isn’t used for the wrong reasons. GDPR is law passed is 2018 which come under the Data Protection Act that a certain set of principles must be followed when dealing with personal information. Since the data is open source and has nothing to do with personal information, none of these laws are broken and it is safe for usage.

The principles are

- Lawfulness, fairness, and transparency.
- Purpose limitation.
- Data minimisation.
- Accuracy
- Storage limitation

- Integrity and confidentiality
- Accountability

Chapter 4
Software Design

This chapter will go through the design and implementation which would result to the final PowerBi report. It would go through the front-end and back-end technologies and system architecture. The technologies used give an overview 4 different factors i.e. retrieval, orchestration, storing and visualisation. All products that are used for this demonstration are from Microsoft.

4.1 System Planning and Requirements

Since the project was mostly cloud based, there had to be constant change on requirements based on the data load. As it is an exploratory software, the data used is mostly available to the public eye and anyone can get hold of that information. The hardware and software requirements for this project are used by identifying what kind of services would be needed for data that is being processed. The available datasets finally sit in sql tables to be further used for visualisation.

The final report should fulfil certain expectations based on technical and depth and usage. Putting these 2 factors into consideration, the requirements will be divided into functional and non-functional. These give a glimpse on what expectations must be achieved before the project is finished and should make sure that every single unit has to be tested and validated. Once all these requirements are met, the check is done for the usage and UI experience just to see if the standards are met. Some of the functional and non-functional requirements were suggested by the project supervisor to

4.1.1 Functional Requirements

Functional requirements give an insight on the behaviour of the system. The can be divided into front end and back end capabilities to validate of all the requirements are being met.

- System shall stick its jurisdiction to the city of Leeds
- Data should be able to refresh at any given point of time when the ETL pipeline is scheduled
- System shall be able to select any year/month and day present in the report.
- The different graphs present in the report shall display all different values related to weather and pollution.
- Data Integrity must be maintained from start to finish based on security and starting to ending
- SQL server must be always available to ensure data can be refreshed at any given moment.

- Ensure that duplication is not done during the transformation process.
- Report should be able to cross reference and provide and analysis between change in climate from one point to another point due to pollution caused.
- Ensure the detailed reports can guide through at point of time to have a glance at what the weather was like at that moment in time.
- Report should also be able to hide tables and figures when the user wants to switch over higher level reports.

4.1.2 Non-Functional Requirements

Non-functional requirements give an insight on the usability and experience of it should ensure that boundaries are maintained when it comes to privacy.

- Users must find the PowerBi report easy to access
- The different filters in the report must be enabled on a page level rather than a single visual level to ensure that the users don't see different values on the same page
- Date filter must not be synched when moving to another page because pollution data is only 2021 and half of 2022 whereas the weather data is for 2010 to present.
- The callout values must be displayed with different colours and a legend should be present for each visual to make sure that users can differentiate between the different fields.

4.2 System Architecture

The inspiration behind this project is to make an application which uses cloud technologies in enabling data engineering driven applications. Most of the services that will be used in the report are based on Azure and show a workflow of an application with a data warehouse would interact. The storage account, ETL tool, Databricks and the database are hosted on Azure and PowerBi reports were developed using the desktop software but were later hosted on the PowerBi web-based application. The figure below will show an architecture on the necessary connections and the flow in which data is taken from starting to end point.

The architecture would be divided into 2 specific sections where one of them focusses on the what the user would see and the backend layer where all the transactions take place. Open weather is the platform that provides global weather indices in the form of APIs. The API data is later converted in the form dataframes and converted into csv to store in the blob storage account using Databricks. Blob Storage is used as a primary source for store and export the tables into the SQL database. By looking at figure 4.1, it gives an overview of the journey through which the data is processed and displayed at the end. On a backend level all the components communicate both ways, but the report and the APIs only communicate in a single way. The SQL database refreshes when required just to show the updated information.

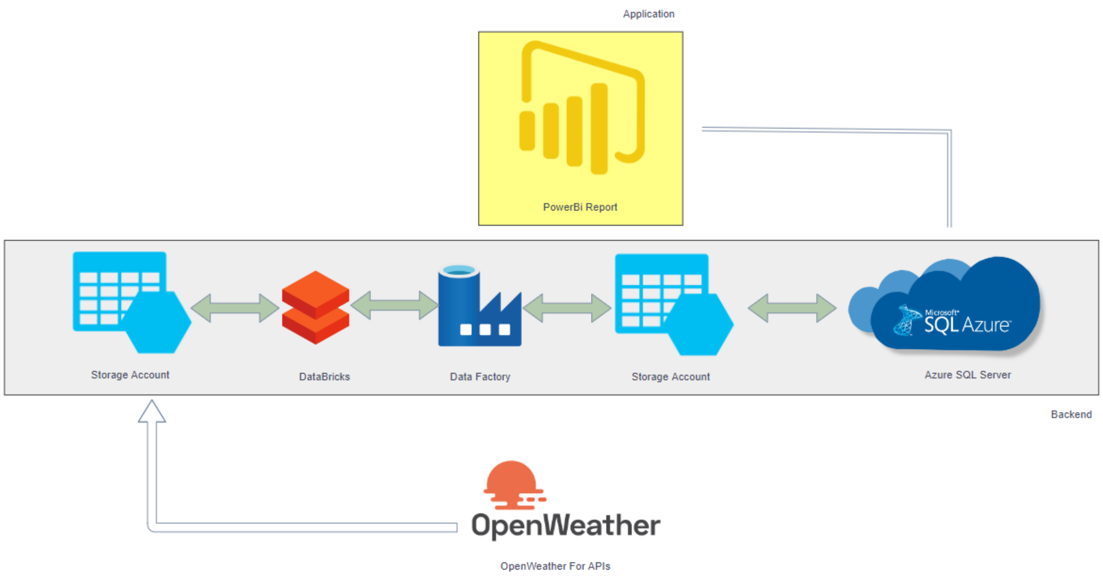


Figure 4.1 *System Architecture*

Data factory is the ETL tool that is being used to gather the input (source) and transform the data into cell values to be put into the database. Once the transformation is performed and the values are put into an SQL tabular format, the values can be used for visualisation and many other Business Intelligence activities.

4.3 Technologies

Historical Weather and Pollution data for one single city can show The technologies used give an overview 4 different factors i.e. retrieval, orchestration, storing and visualisation. The platforms used are all on the Microsoft environment and

4.3.1 Azure

Azure is the cloud service provider run by Microsoft which offers a vast number of services which enable clients or individuals to switch from on premises to a virtual infrastructure where the hardware is managed by Microsoft and clients must follow a pay as you go model. The benefits of using the platform are limitless as one must not be concerned about the maintenance of the underlying infrastructure. There are a list of services that were used for the execution of this project.

4.3.1.1 Azure Databricks

Databricks is a Spark based platform which is used for multiple applications like SQL, Machine Learning, Data Science and Engineering. Spark is an analytics engine which can run on multiple or single clusters just to ensure data parallelism and fault tolerance. The platform primarily runs on 3 different languages SQL, Python and Scala. It is currently preferred over standalone spark as interacting with other cloud technologies under same resource group becomes significantly easier. This can be used as a prominent feature for data warehousing transformation of source data and automatic cluster management can be done in one section. It can also be used for Machine Learning which help in running huge datasets. For this application, databricks is used to convert JSON files into a tabular format to be stored in the blob storage.

4.3.1.2 Azure Data Factory

Azure data factory is an integration service that is used to design ETL workflows. The reason behind using this tool is to automate movement and transform data at instance needed. More the number of times the pipeline is scheduled for refresh, costlier the process becomes. The datafactory must be connected to at least 2 platforms. That can range from databases, storage accounts to notebooks. Once these linked services are established a source (starting point) and a sink(destination) must be configured. Through notebooks, stored procedures or in-built transformation functions the destination receives the desired output. For this application, the data factory is used to schedule the notebook to perform a scheduled transform from one blob storage to the database. The datafactory works on triggers therefore once the triggers are published after every change, the pipeline must be triggered to run without any errors. New rows in the database get merged with the old ones after the pipelines are run. The data factory will be having a linked service between the blob storage and the database.

4.3.1.3 Azure SQL Server

A cloud-based database solution that runs on MSSQL. Features like scalability, reliability, backup, and robustness is taken care of in this solution. SQL server can be used to store data in the form of relational models as information retrieval would become easier. For this project two fact tables and 1 dim table will be used (Weather and Pollution Fact table and 1 Date Dimension table). It can help to store all the values rather just using CSV files to process because the database provides a better storing solution and information retrieval is much easier and robust. Once these values are stored in the database they are ready for visualisation.

4.3.1.4 Blob Storage

Blob storage is a storage solution for the cloud. Any client can access the objects in the storage using HTTP/HTTPS. Azure’s storage account would have a container in which it is dissected into different blobs. It almost acts like a third-party storage which has to be linked to the other services in the resource group. For this application, blob storage is used in 2 cases which is to accept input and from a local source and to store output files processed from databricks.

4.3.1.5 Key Vault

Key vault helps in storing separate keys for each and service in the Azure resource group. The keys can range from passwords, API keys to service credential tokens. This vault helps in user the name assigned to key directly rather than exposing the actual key name while entering in a databricks notebook. It promotes good cryptography practices. Keys can be assigned to a service level as well are create an authentication for a logic app, serverless function, storage queues, etc. Key vault in this case will be used to create a secret for the storage account to create a mount for the databricks service.

4.4.2 PowerBi

PowerBi is a visualisation tool that is generally used for Business Intelligence applications. It is a Software as a Service application when the report is uploaded. The desktop version offers a lot of features which helps in creating responsive graphs. It can connect to multiple environments like excel, csv, SQL server, XML, etc. It can be used to create new columns, DAX queries and data models. PowerBi ideally prefers a star schema where there would one fact table and multiple dimensions tables connected to that as data retrieval becomes faster. This project will PowerBi to generate reports relating to weather and the impact caused by pollutants.

4.5 Datasets used in the system

The datasets used in this system are external APIs which are taken from OpenWeather.org that provides information regarding weather at any given location, solar radiation, pollution, vegetation index and many more. For this project, the datasets used are weather and pollution. The dataset for weather is for 12 years and the pollution service has only stated since November 2020 therefore, the dataset is from December 2020- present. All the values are retrieved by registering to the portal and creating an API key for retrieval. There is a date table followed by a serial number which helps in creating relationships between 2 tables. The data is generated on an hourly basis but the date table only contains the date. The hour partition is done later in PowerBi.

4.5.2 LeedsWeather10yeardata.JSON

This JSON file provides weather data for years 2010-present. This data is fixed to the jurisdiction of Leeds (Lat - 53.8998N Long - 1.5491W). This file includes significant fields like pressure, max temperature, max temperature, dew point, weather description, visibility, wind and humidity. This JSON file is run through databricks to process into a CSV file that must be saved in the storage account.

4.5.3 Leeds_Historical_Air_Pollution_Data.JSON

The JSON file provides pollution data from 2022-present. This data provides fields like carbon monoxide, nitrogen dioxide, nitric oxide, ozone, sulphur dioxide, pm2_5 and pm10. The JSON file is run through databricks to process into a CSv file that must be saved into the storage account.

4.5.4 LeedsWeather10yeardata.CSV

This would be the processed CSV file of the weather dataset present in the storage account which would be used in the datafactory to be later stored in the database.

4.5.5 Leeds_Historical_Air_Pollution_Data.CSV

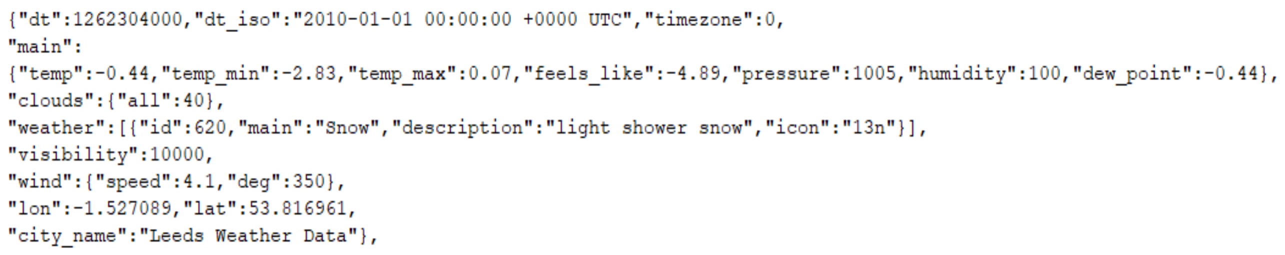
The processed pollution file in CSV format would be saved in the storage account and later used in the data factory to store all the values in the database.

Chapter 5
Software Implementation

This section will discuss the list of sprints taken to reach the final stage. Since, this is a cloud-based project there will be a list of screenshots that run through the whole development process.

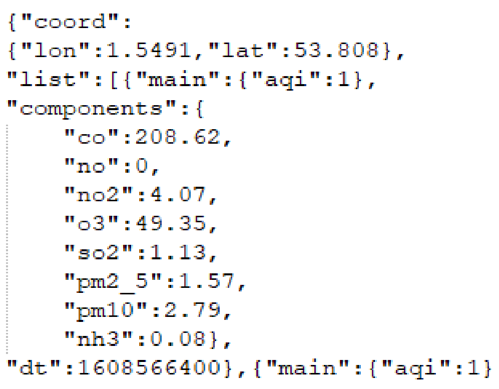
5.1 Data Source

The data used In this project is obtained from https://openweathermap.org/. The APIS used are Historical Bulk Weather API and Pollution API as shown in figure 4.2 and 4.3 respectively.



```
{"dt":1262304000,"dt_iso":"2010-01-01 00:00:00 +0000 UTC","timezone":0,
"main":
{"temp":-0.44,"temp_min":-2.83,"temp_max":0.07,"feels_like":-4.89,"pressure":1005,"humidity":100,"dew_point":-0.44},
"clouds":{"all":40},
"weather":[{"id":620,"main":"Snow","description":"light shower snow","icon":"13n"}],
"visibility":10000,
"wind":{"speed":4.1,"deg":350},
"lon":-1.527089,"lat":53.816961,
"city_name":"Leeds Weather Data"},
```

Figure 5.1 *Sample JSON object for Leeds Weather data*

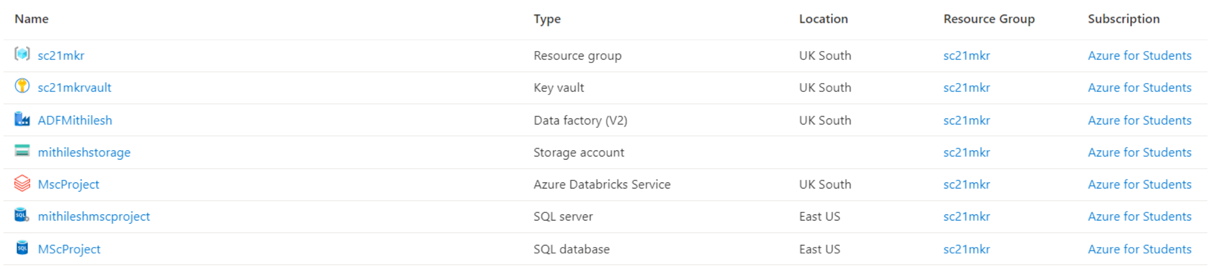


```
{"coord":
{"lon":1.5491,"lat":53.808},
"list":[{"main":{"aqi":1},
"components":{
    "co":208.62,
    "no":0,
    "no2":4.07,
    "o3":49.35,
    "so2":1.13,
    "pm2_5":1.57,
    "pm10":2.79,
    "nh3":0.08},
"dt":1608566400},{"main":{"aqi":1}
```

Figure 5.2 *Sample JSON object for Leeds Pollution data*

The pollution JSON has many nested elements regarding date, time zone, temperature, weather, location, visibility, and wind. The Leeds pollution data have nested list for the pollutants present and the rest are straight forward. To convert the data into a parquet or a csv format the JSON files have to be run through databricks. Databricks was chosen for this project as it is a cloud-based solution, and the rest of the infrastructure is all set up in the cloud as well.

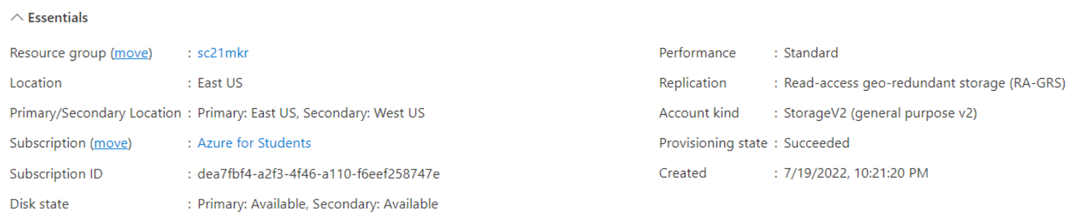
5.2 Storage Account



Name	Type	Location	Resource Group	Subscription
sc21mkr	Resource group	UK South	sc21mkr	Azure for Students
sc21mkrvault	Key vault	UK South	sc21mkr	Azure for Students
ADFMithilesh	Data factory (V2)	UK South	sc21mkr	Azure for Students
mithileshstorage	Storage account		sc21mkr	Azure for Students
MscProject	Azure Databricks Service	UK South	sc21mkr	Azure for Students
mithileshmscproject	SQL server	East US	sc21mkr	Azure for Students
MScProject	SQL database	East US	sc21mkr	Azure for Students

Figure 5.3 *List of services used under sc21mkr resource group*

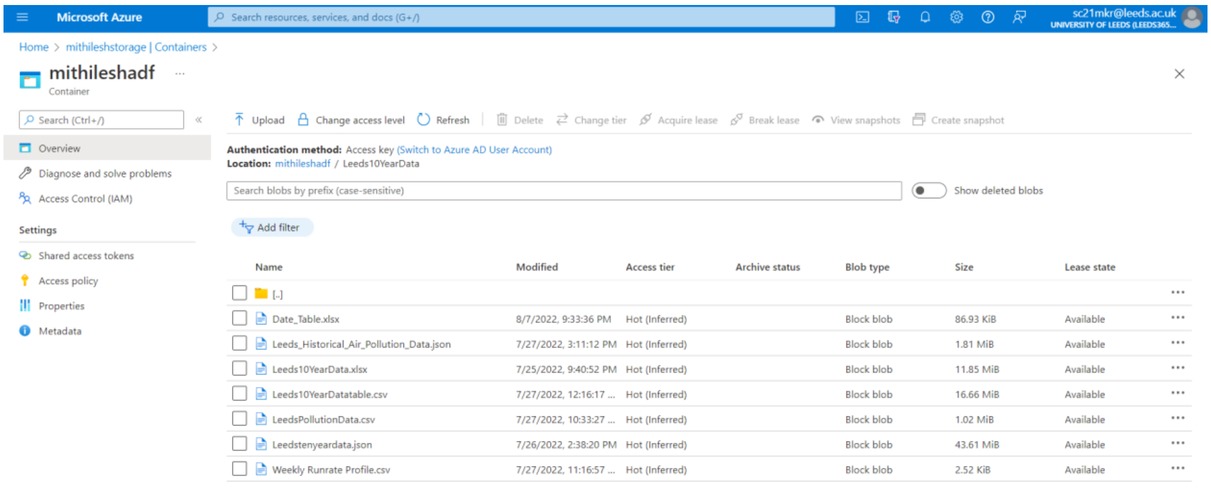
The list of resources used are displayed in Figure 4.4. The storage account is used as a secondary storage where all the files are dumped which were retrieved from the website. Figure 4.5 shows the configuration for the storage account created. In storage account there are many ways to store objects like containers, file shares, queues, and tables. Any unstructured data relating to the project is stored at the container (blob storage).



Essentials

Resource group (move)	: sc21mkr	Performance	: Standard
Location	: East US	Replication	: Read-access geo-redundant storage (RA-GRS)
Primary/Secondary Location	: Primary: East US, Secondary: West US	Account kind	: StorageV2 (general purpose v2)
Subscription (move)	: Azure for Students	Provisioning state	: Succeeded
Subscription ID	: dea7fbf4-a2f3-4f46-a110-f6eef258747e	Created	: 7/19/2022, 10:21:20 PM
Disk state	: Primary: Available, Secondary: Available		

Figure 5.4 *Storage Account Configuration*



Authentication method: Access key (Switch to Azure AD User Account)
Location: mithileshadf / Leeds10YearData

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
Date_Table.xlsx	8/7/2022, 9:33:36 PM	Hot (Inferred)		Block blob	86.93 KiB	Available
Leeds_Historical_Air_Pollution_Data.json	7/27/2022, 3:11:12 PM	Hot (Inferred)		Block blob	1.81 MiB	Available
Leeds10YearData.xlsx	7/25/2022, 9:40:52 PM	Hot (Inferred)		Block blob	11.85 MiB	Available
Leeds10YearDatatable.csv	7/27/2022, 12:16:17 ...	Hot (Inferred)		Block blob	16.66 MiB	Available
LeedsPollutionData.csv	7/27/2022, 10:33:27 ...	Hot (Inferred)		Block blob	1.02 MiB	Available
Leedstenyeardata.json	7/26/2022, 2:38:20 PM	Hot (Inferred)		Block blob	43.61 MiB	Available
Weekly Runrate Profile.csv	7/27/2022, 11:16:57 ...	Hot (Inferred)		Block blob	2.52 KiB	Available

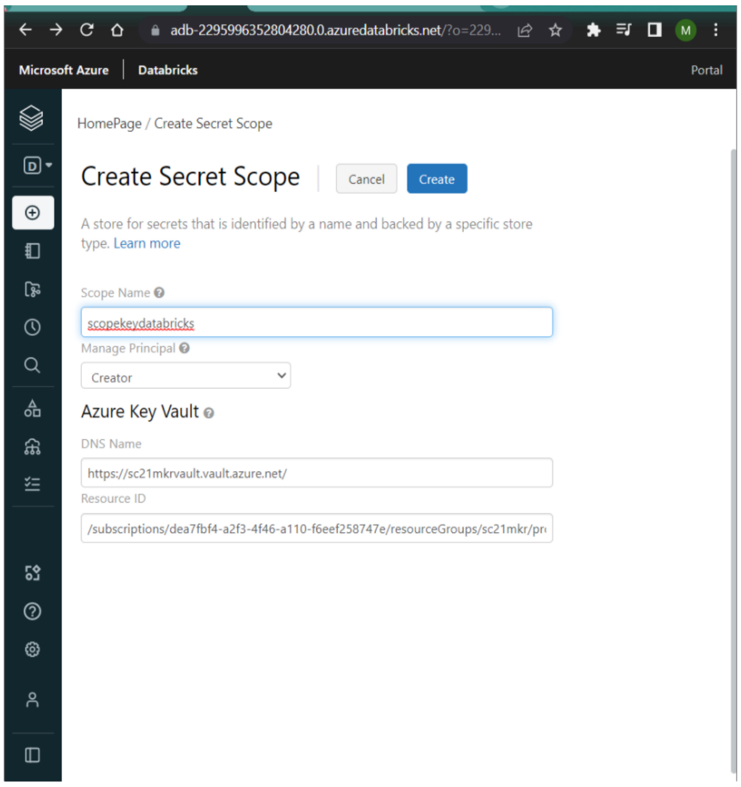
Figure 5.5 *Storage Account Configuration*

All the input and output files are stored in the container as shown in Figure 4.6. Once Leeds_Historical_Air_Pollution_Data.json and Leedstenyeardata.json are files are uploaded to the container, the next task will be to create a spark cluster and run the JSON file to csv format.

5.3 Databricks

Databricks runs on a spark platform where the code can either be run through Python or SQL or Scala. The spark cluster that is created is based on Python where the code is much more user friendly and has a wide range of libraries. To communicate between databricks and the storage account, a connection must be created. This connection helps in mounting the storage account to databricks that grants read and write permissions. Mounting Azure Data Lake Storage to Databricks has a list of steps that must be followed.

- Create an azure Active Directory Account
- Register web application to connect with databricks
- Store the key from web app to azure key vault
- Create scope in databricks to connect to key vault.
- Mount ADLS/ Blob storage using ClientID, tenant ID and key
- Add “storage blob data contributor” under Access Control (IAM) for storage account



HomePage / Create Secret Scope

Create Secret Scope

Cancel Create

A store for secrets that is identified by a name and backed by a specific store type. Learn more

Scope Name: scopekeydatabricks

Manage Principal: Creator

Azure Key Vault

DNS Name: https://sc21mkrvault.vault.azure.net/

Resource ID: /subscriptions/dea7fbf4-a2f3-4f46-a110-f6eef258747e/resourceGroups/sc21mkr/pr

Figure 5.6 *Create Scope in Databricks*

Once the scope is created as shown in figure 4.7 databricks and storage account have an established connection and can the storage account can be mounted for further editing of the unstructured data in the storage account.

Since data bricks runs on a spark platform to process datasets of huge volumes a cluster must be created for the code to run. This encourages parallel processing to run through datasets in a short amount of time.

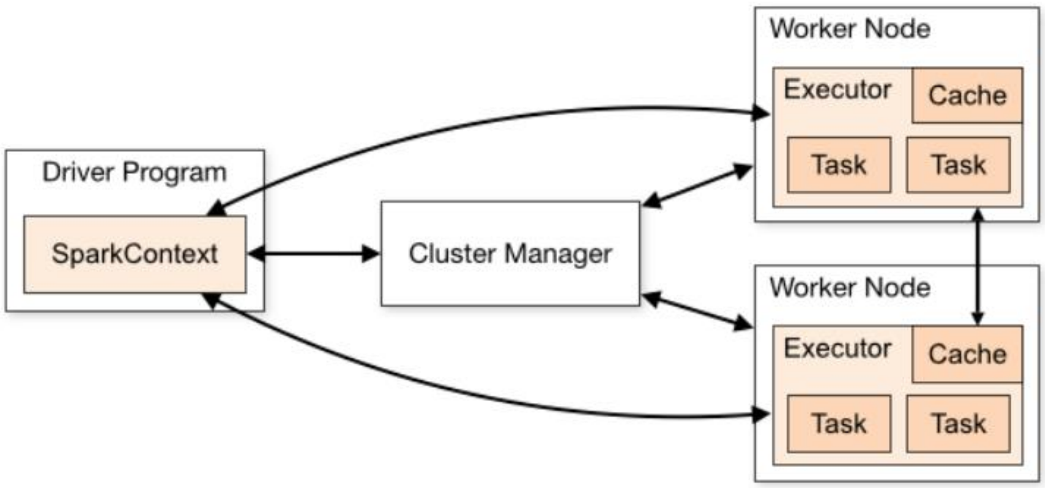
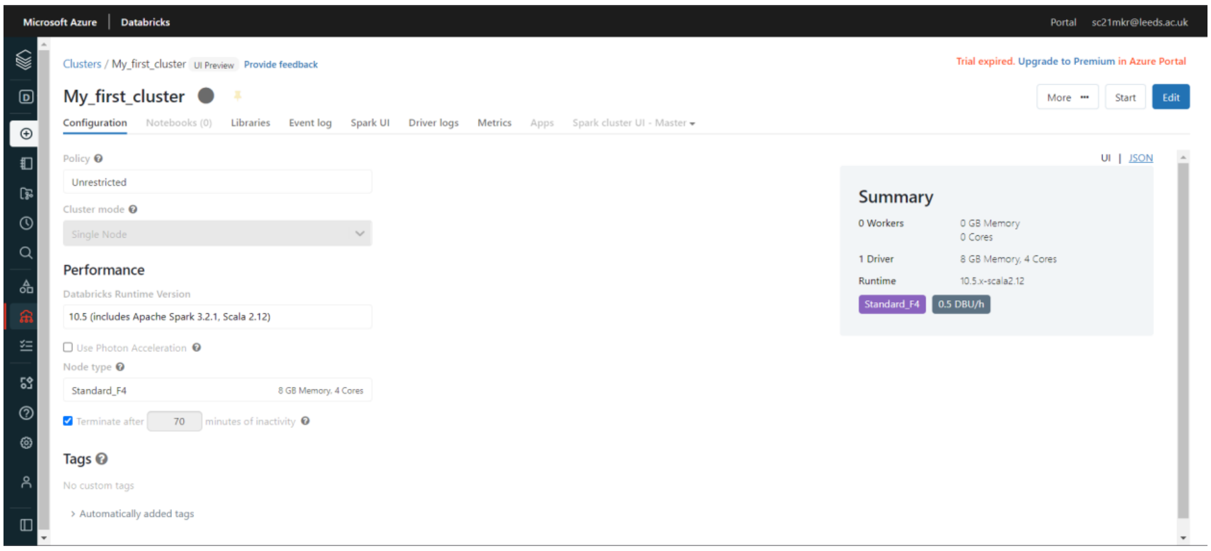


Figure 5.7 *Spark Architecture*

Spark works on a master slave architecture where the worker nodes work parallelly to various tasks assigned to them. Job scheduling is done by the cluster manager at a level with the application. Figure 4.9 shows the configuration of the cluster created which includes the drivers, performance, workers, and runtime. Th trial version last for 14 days only allows to choose the minimum configuration for the environment.



My_first_cluster

Policy: Unrestricted

Cluster mode: Single Node

Performance

Databricks Runtime Version: 10.5 (includes Apache Spark 3.2.1, Scala 2.12)

☐ Use Photon Acceleration

Node type: Standard_F4 8 GB Memory, 4 Cores

☑ Terminate after 70 minutes of inactivity

Tags

No custom tags

› Automatically added tags

Summary

0 Workers	0 GB Memory 0 Cores
1 Driver	8 GB Memory, 4 Cores
Runtime	10.5.x-scala2.12

Standard_F4 0.5 DBU/h

Figure 5.8 *Spark Cluster Configuration*

The next process is to extract the JSON and convert into a dataframe. Using the mount function in dbutils as shown in figure 4.10 the storage account can me connected to databricks.

```python
dbutils.fs.mount(
  source = "wasbs://mithileshadf@mithileshstorage.blob.core.windows.net/",
  mount_point = "/mnt/mithileshmount",
  extra_configs =
{"fs.azure.account.key.mithileshstorage.blob.core.windows.net":"FMrzI7vVsI1cXQjxnTLfQL2Z/J2mYvaU1wbiiJwtvsQGthUJaB7vNy5owGsmFShf2pRLgqRSw1Ck+AStQN71bQ=="}
)
```

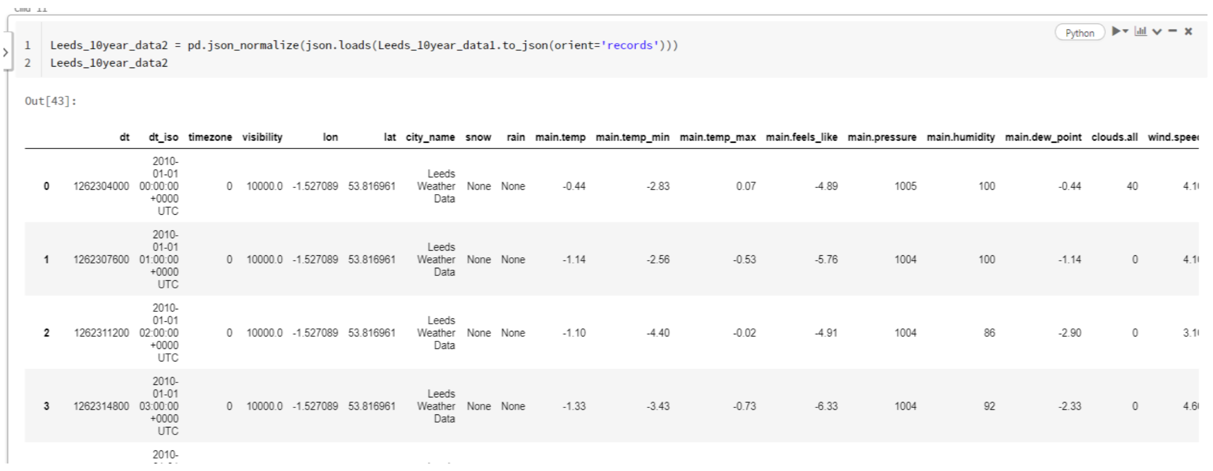
Figure 5.9 *Mount Function for Storage account*

After reading the JSON file, a data frame must be created for all these files to export to a CSV. Explode function and normalize function help in achieving this as shown in figure 4.11 and 4.12

```python
Leeds_10year_data1 = Leeds_10year_data.explode('weather')
```

```
Command took 0.25 seconds -- by sc21mkr@leeds.ac.uk at 7/27/2022, 12:11:42 PM on
My_first_cluster
```

Figure 5.10 *Explode function for the weather segment*



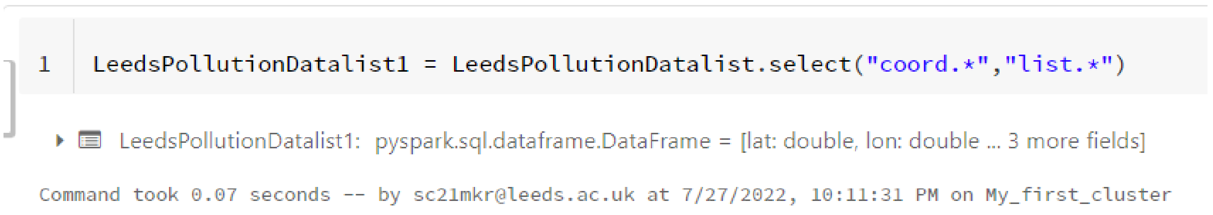
```
Leeds_10year_data2 = pd.json_normalize(json.loads(Leeds_10year_data1.to_json(orient='records')))
Leeds_10year_data2
```

Out[43]:

	dt	dt_iso	timezone	visibility	lon	lat	city_name	snow	rain	main.temp	main.temp_min	main.temp_max	main.feels_like	main.pressure	main.humidity	main.dew_point	clouds.all	wind.spee
0	1262304000	2010-01-01 00:00:00 +0000 UTC	0	10000.0	-1.527089	53.816961	Leeds Weather Data	None	None	-0.44	-2.83	0.07	-4.89	1005	100	-0.44	40	4.1
1	1262307600	2010-01-01 01:00:00 +0000 UTC	0	10000.0	-1.527089	53.816961	Leeds Weather Data	None	None	-1.14	-2.56	-0.53	-5.76	1004	100	-1.14	0	4.1
2	1262311200	2010-01-01 02:00:00 +0000 UTC	0	10000.0	-1.527089	53.816961	Leeds Weather Data	None	None	-1.10	-4.40	-0.02	-4.91	1004	86	-2.90	0	3.1
3	1262314800	2010-01-01 03:00:00 +0000 UTC	0	10000.0	-1.527089	53.816961	Leeds Weather Data	None	None	-1.33	-3.43	-0.73	-6.33	1004	92	-2.33	0	4.6

Figure 5.11 *Normalize function used to remove nested functions*

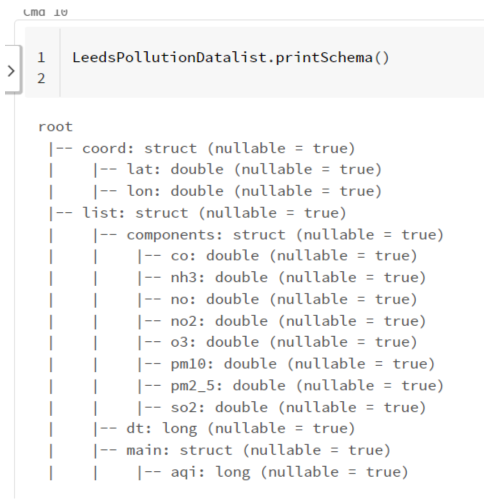
The pollution data is exported like the weather data. Some nested lists as shown in Figure 4.13 are a part of the pollution data, and they were removed by using select function for the pandas dataframe like the database query as shown in figure 4.12.



```
LeedsPollutionDatalist1 = LeedsPollutionDatalist.select("coord.*","list.*")
```

```
LeedsPollutionDatalist1:  pyspark.sql.dataframe.DataFrame = [lat: double, lon: double ... 3 more fields]
Command took 0.07 seconds -- by sc21mkr@leeds.ac.uk at 7/27/2022, 10:11:31 PM on My_first_cluster
```

Figure 5.12 *Select function to remove nested lists*



```
LeedsPollutionDatalist.printSchema()
```

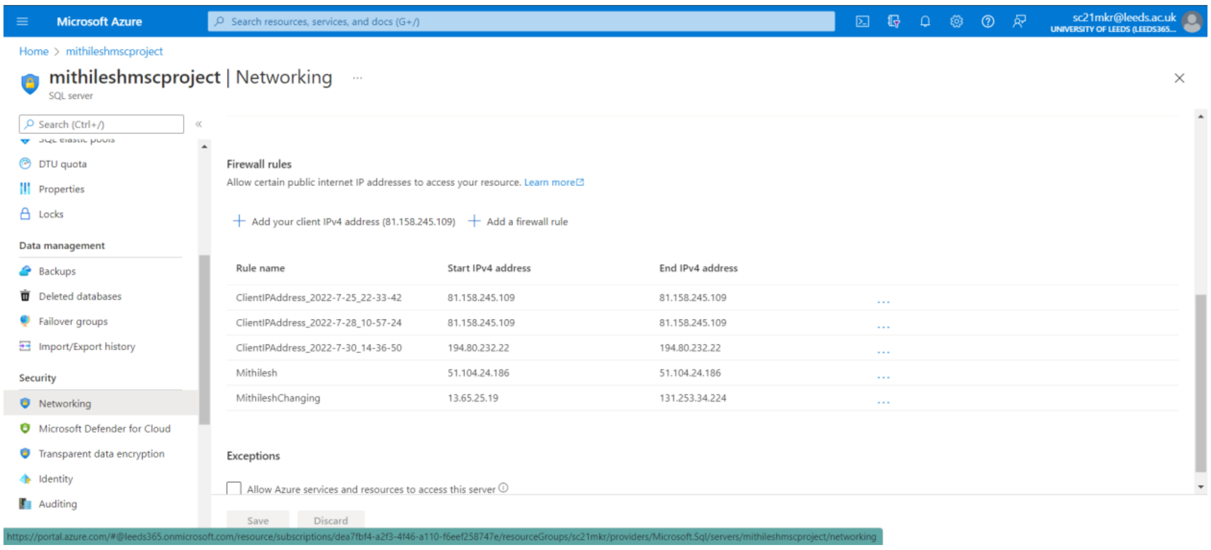
```
root
 |-- coord: struct (nullable = true)
 |    |-- lat: double (nullable = true)
 |    |-- lon: double (nullable = true)
 |-- list: struct (nullable = true)
 |    |-- components: struct (nullable = true)
 |    |    |-- co: double (nullable = true)
 |    |    |-- nh3: double (nullable = true)
 |    |    |-- no: double (nullable = true)
 |    |    |-- no2: double (nullable = true)
 |    |    |-- o3: double (nullable = true)
 |    |    |-- pm10: double (nullable = true)
 |    |    |-- pm2_5: double (nullable = true)
 |    |    |-- so2: double (nullable = true)
 |    |-- dt: long (nullable = true)
 |    |-- main: struct (nullable = true)
 |    |    |-- aqi: long (nullable = true)
```

Figure 5.13 *Select function to remove nested lists*

The two dataframes are exported to a CSV format as shown in figure 4.6 as output files from the dataframe. The next step would be to create a database with the same columns for all of the captures values in CSV to get stored.

5.4 Azure SQL server

Azure SQL server was chosen for this project as a local database would raise connectivity issues and wouldn't be available all the time. For the transaction to take place at regular intervals a reliable server must be chosen which suits the configuration and is available all the time. The security and backup is available in the Azure environment. The database has a size of 2GB, and the plan chosen was a basic plan which is for less demanding workloads that includes geo-redundant backup storage and has four database transaction units (DTUs). To access the server through local Sql server management studio environment, client IP address must be added as shown in Figure 4.13


Microsoft Azure
Search resources, services, and docs (G+/)
sc21mkr@leeds.ac.uk
UNIVERSITY OF LEEDS (LEEDS365...
Home > mithileshmscproject
mithileshmscproject | Networking
SQL server
Search (Ctrl+/)
SQL elastic pools
DTU quota
Properties
Locks
Data management
Backups
Deleted databases
Failover groups
Import/Export history
Security
Networking
Microsoft Defender for Cloud
Transparent data encryption
Identity
Auditing
Save
Discard
https://portal.azure.com/#@leeds365.onmicrosoft.com/resource/subscriptions/dea7fbf4-a2f3-4f46-a110-f6eef258747e/resourceGroups/sc21mkr/providers/Microsoft.Sql/servers/mithileshmscproject/networking

Firewall rules

Allow certain public internet IP addresses to access your resource. Learn more

+ Add your client IPv4 address (81.158.245.109) + Add a firewall rule

Rule name	Start IPv4 address	End IPv4 address
ClientIPAddress_2022-7-25_22-33-42	81.158.245.109	81.158.245.109
ClientIPAddress_2022-7-28_10-57-24	81.158.245.109	81.158.245.109
ClientIPAddress_2022-7-30_14-36-50	194.80.232.22	194.80.232.22
Mithilesh	51.104.24.186	51.104.24.186
MithileshChanging	13.65.25.19	131.253.34.224

Exceptions

☐ Allow Azure services and resources to access this server

Figure 5.14 *Client IP address for SQL server*

The database consists of 3 tables

- projectRAW.Date_Dim
- projectRAW.Leeds_10_year_Data
- projectRAW.Leeds_Pollution_2_Year_Data

projectRAW is the name of the schema used for all the tables. Leeds_10_year_Data and Leeds_Pollution_2_Year_Data are fact tables that contain the aggregation values for the necessary fields. Date_dim is the dimension table which helps as a join between the 2 date columns in the fact tables.

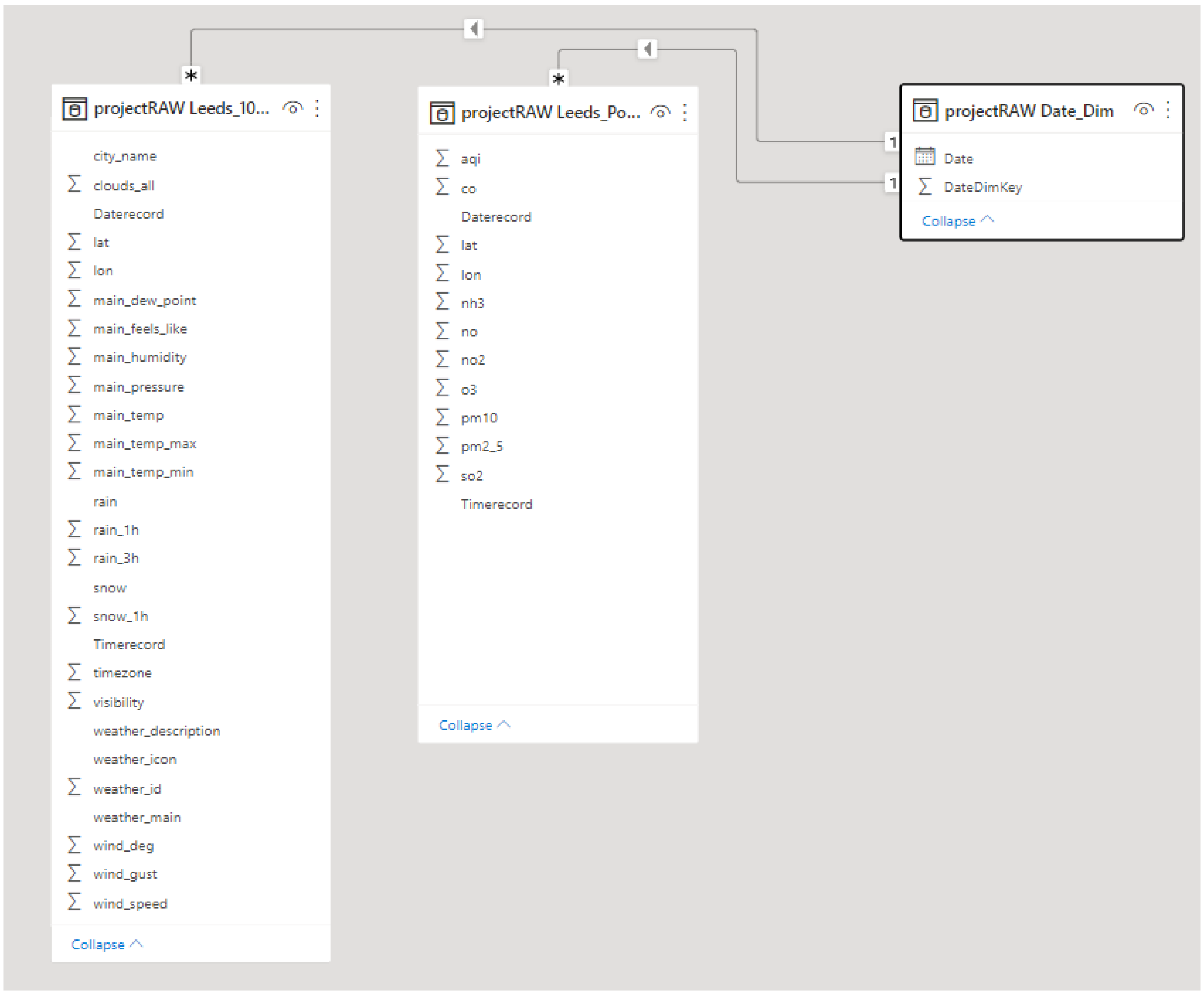
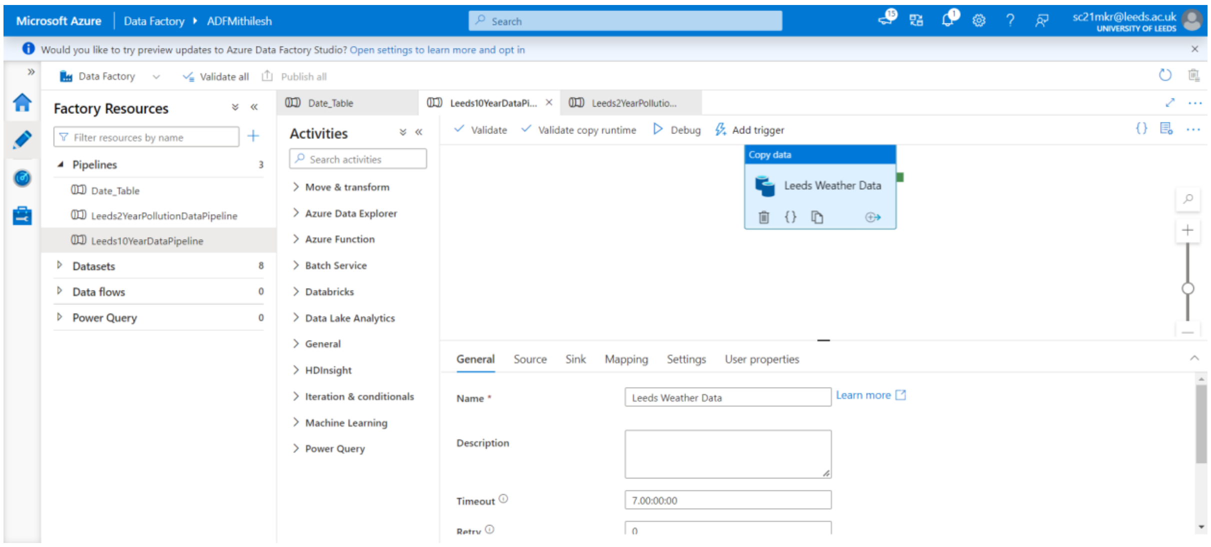


Figure 5.15 *Datamodel diagram*

To store the values into the database, the datafactory must perform an ETL load.

5.5 Azure Data Factory

Data Factory is an orchestrating which acts as a vital entity in the field of data warehousing. It’s much faster than its predecessor SSIS (SQL server Integration Service) which was on premises. Due to this tool being on the cloud it becomes significantly easier to connect to another source which can be any secondary storage to database to an on premises to database. For this project source would be CSV file present in Blob storage and the destination would be the tables at the SQL server.

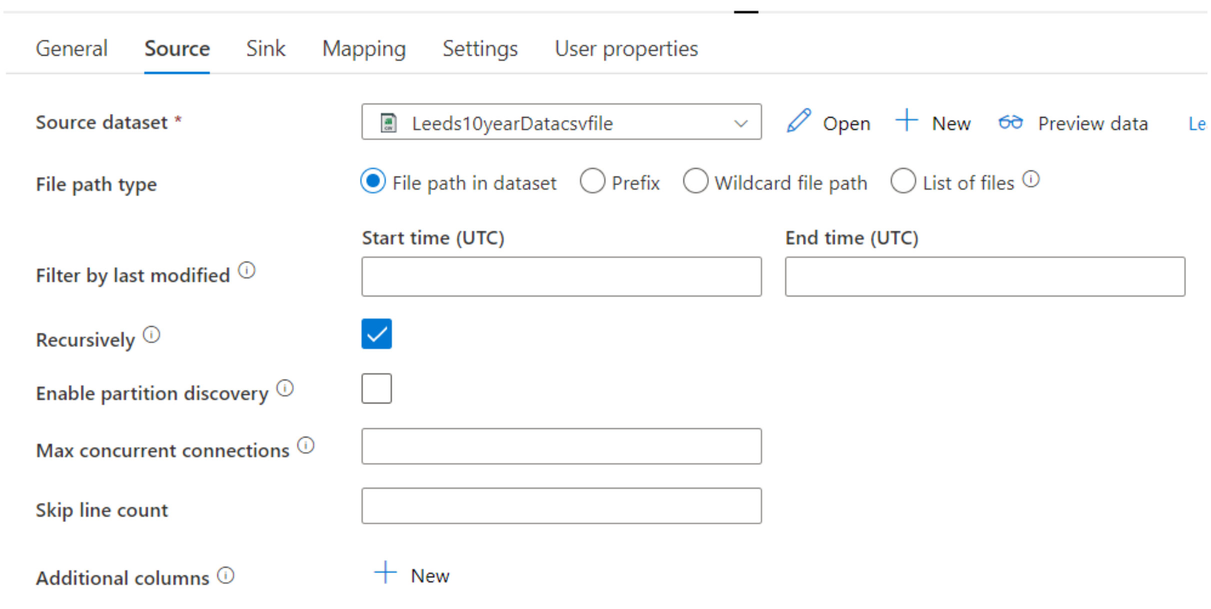


Would you like to try preview updates to Azure Data Factory Studio? Open settings to learn more and opt in

Name *	Leeds Weather Data
Description	
Timeout	7.00:00:00
Retry	0

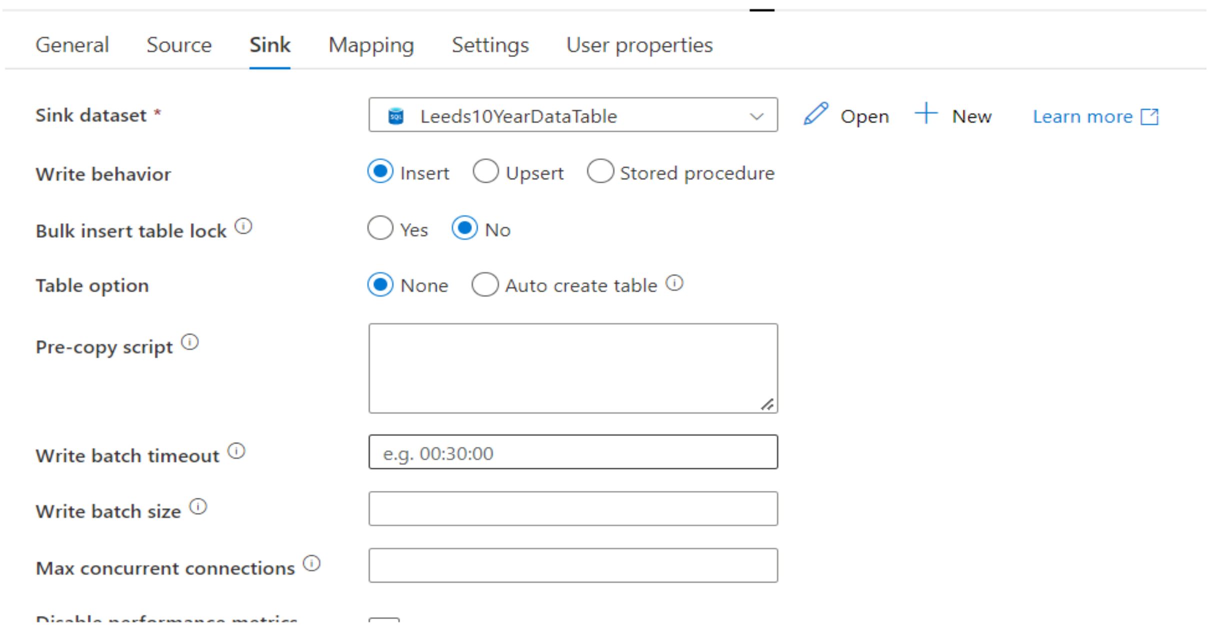
Figure 5.16 Leeds Weather Pipeline

Figure 4.15 gives an example of one of the pipelines created for the report, figure 4.17 gives the source of weather pipeline being displayed and figure 4.18 shows the destination of the pipeline. Figure 4.19 shows the JSON code for the mapping between the column headers between the CSV file and the database table. Data types have to be matched and made sure that the datafactory's IP address is added as a client in the SQL server. Since the datafactory's IP for a basic version keeps changing a range from 13.65.25.19 to 131.253.34.224 for IPV4 IP address was added as shown in figure 4.13. There should also be two connections with the blob storage and sql database. The connection is shown in figure 4.20



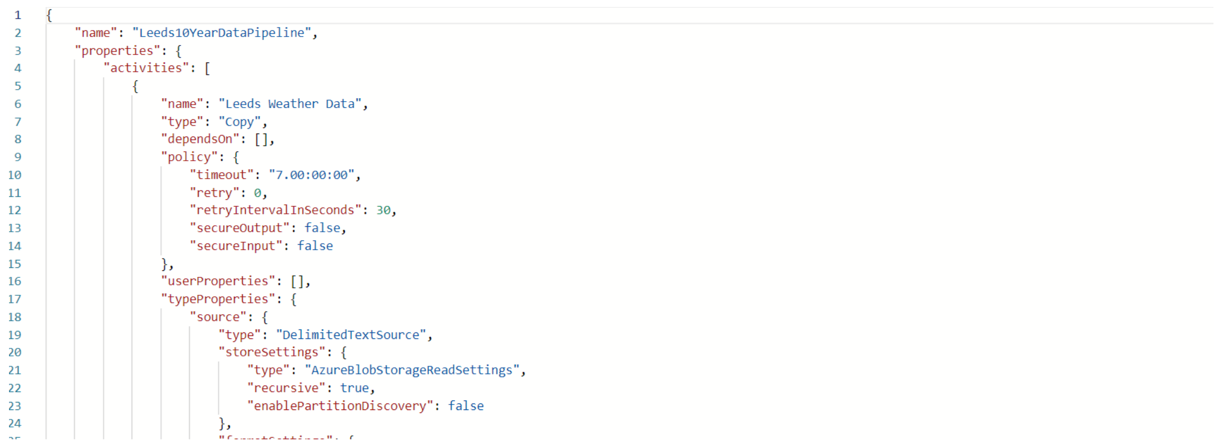
Source dataset *	Leeds10yearDatacsvfile
File path type	[✓ File path in dataset] Prefix Wildcard file path List of files
Filter by last modified	Start time (UTC): End time (UTC):
Recursively	✓
Enable partition discovery	☐
Max concurrent connections	
Skip line count	
Additional columns	New

Figure 5.17 Source of Pipeline



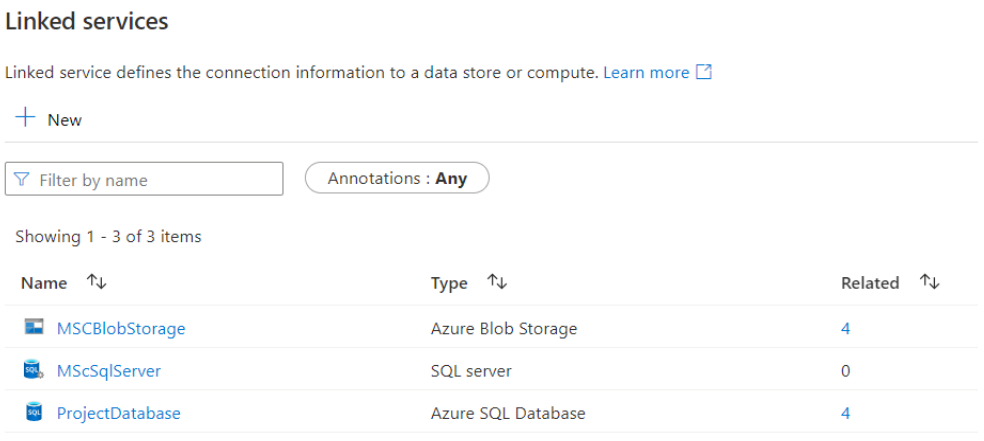
Sink dataset *	Leeds10YearDataTable
Write behavior	[✓ Insert] Upsert Stored procedure
Bulk insert table lock	Yes [✓ No]
Table option	[✓ None] Auto create table
Pre-copy script	
Write batch timeout	e.g. 00:30:00
Write batch size	
Max concurrent connections	

Figure 5.18 *Destination of Pipeline*



```
{
    "name": "Leeds10YearDataPipeline",
    "properties": {
        "activities": [
            {
                "name": "Leeds Weather Data",
                "type": "Copy",
                "dependsOn": [],
                "policy": {
                    "timeout": "7.00:00:00",
                    "retry": 0,
                    "retryIntervalInSeconds": 30,
                    "secureOutput": false,
                    "secureInput": false
                },
                "userProperties": [],
                "typeProperties": {
                    "source": {
                        "type": "DelimitedTextSource",
                        "storeSettings": {
                            "type": "AzureBlobStorageReadSettings",
                            "recursive": true,
                            "enablePartitionDiscovery": false
                        },
```

Figure 5.19 *Destination of Pipeline*



Linked services

Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 3 of 3 items

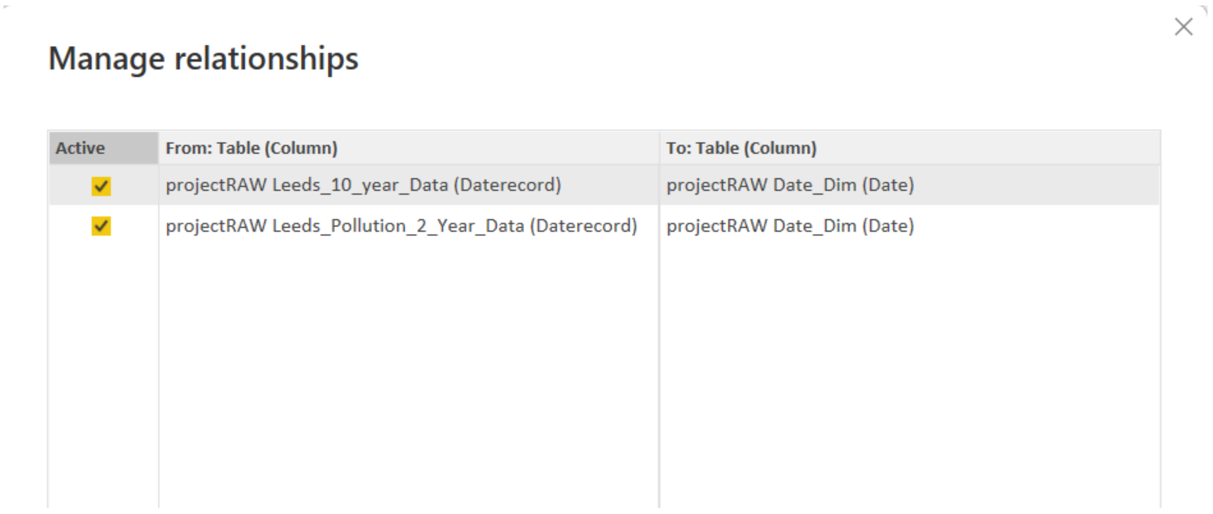
Name	Type	Related
MSCBlobStorage	Azure Blob Storage	4
MScSqlServer	SQL server	0
ProjectDatabase	Azure SQL Database	4

Figure 5.20 *Data Factory's Linked Services*

After these steps, the values will be loaded and have the values must be used for visualisation in PowerBi.

5.6 Power Bi

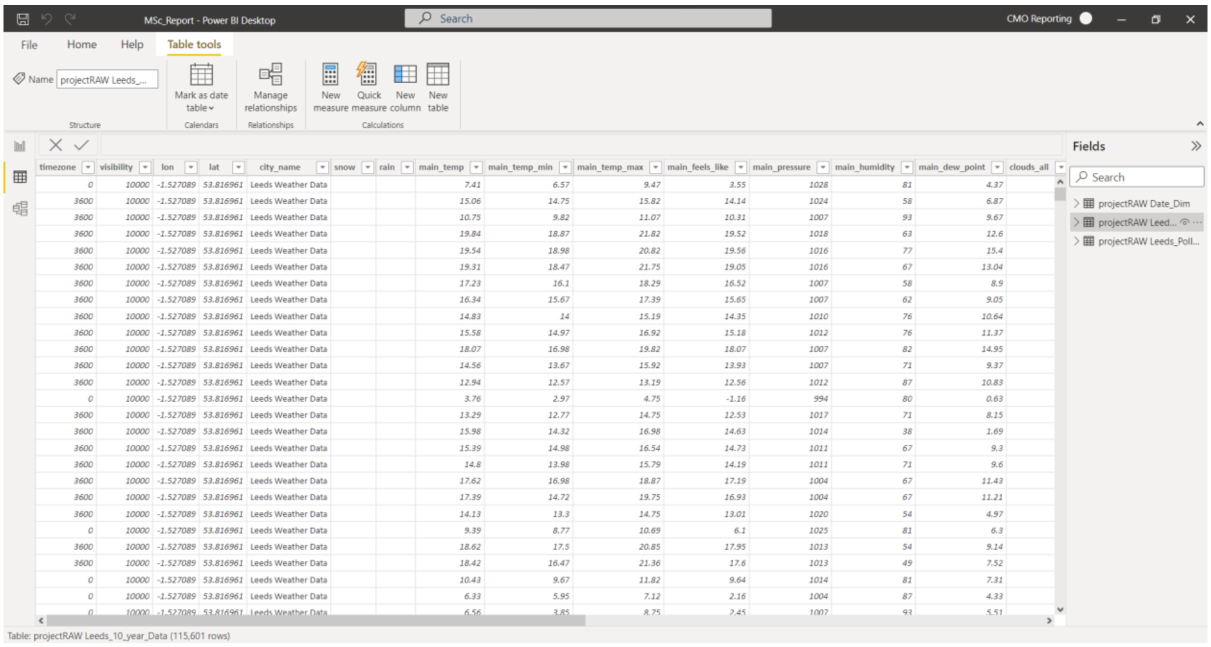
PowerBi is a visualisation tool which is desktop based and once published it goes on software as a Service (SaaS) platform. Once the values are added for the date table, weather table and the pollution dataset, a relationship must be added on the model shown above (Figure 4.14). Figure 4.21 shows the relationships being created. Fig 4.22 and 4.23 give a preview of the weather data and pollution data respectively.



Manage relationships

Active	From: Table (Column)	To: Table (Column)
✓	projectRAW Leeds_10_year_Data (Daterecord)	projectRAW Date_Dim (Date)
✓	projectRAW Leeds_Pollution_2_Year_Data (Daterecord)	projectRAW Date_Dim (Date)

Figure 5.21 *Relationships*



timezone	visibility	lon	lat	city_name	snow	rain	main_temp	main_temp_min	main_temp_max	main_feels_like	main_pressure	main_humidity	main_dew_point	clouds_all
0	10000	-1.527089	53.816961	Leeds Weather Data			7.41	6.57	9.47	3.55	1028	81	4.37	
3600	10000	-1.527089	53.816961	Leeds Weather Data			15.06	14.75	15.82	14.14	1024	58	6.87	
3600	10000	-1.527089	53.816961	Leeds Weather Data			10.75	9.82	11.07	10.31	1007	93	9.67	
3600	10000	-1.527089	53.816961	Leeds Weather Data			19.84	18.87	21.82	19.52	1018	63	12.6	
3600	10000	-1.527089	53.816961	Leeds Weather Data			19.54	18.98	20.82	19.56	1016	77	15.4	
3600	10000	-1.527089	53.816961	Leeds Weather Data			19.31	18.47	21.75	19.05	1016	67	13.04	
3600	10000	-1.527089	53.816961	Leeds Weather Data			17.23	16.1	18.29	16.52	1007	58	8.9	
3600	10000	-1.527089	53.816961	Leeds Weather Data			16.34	15.67	17.39	15.65	1007	62	9.05	
3600	10000	-1.527089	53.816961	Leeds Weather Data			14.83	14	15.19	14.35	1010	76	10.64	
3600	10000	-1.527089	53.816961	Leeds Weather Data			15.58	14.97	16.92	15.18	1012	76	11.37	
3600	10000	-1.527089	53.816961	Leeds Weather Data			18.07	16.98	19.82	18.07	1007	82	14.95	
3600	10000	-1.527089	53.816961	Leeds Weather Data			14.56	13.67	15.92	13.93	1007	71	9.37	
3600	10000	-1.527089	53.816961	Leeds Weather Data			12.94	12.57	13.19	12.56	1012	87	10.83	
0	10000	-1.527089	53.816961	Leeds Weather Data			3.76	2.97	4.75	-1.16	994	80	0.63	
3600	10000	-1.527089	53.816961	Leeds Weather Data			13.29	12.77	14.75	12.53	1017	71	8.15	
3600	10000	-1.527089	53.816961	Leeds Weather Data			15.98	14.32	16.98	14.63	1014	38	1.69	
3600	10000	-1.527089	53.816961	Leeds Weather Data			15.39	14.98	16.54	14.73	1011	67	9.3	
3600	10000	-1.527089	53.816961	Leeds Weather Data			14.8	13.98	15.79	14.19	1011	71	9.6	
3600	10000	-1.527089	53.816961	Leeds Weather Data			17.62	16.98	18.87	17.19	1004	67	11.43	
3600	10000	-1.527089	53.816961	Leeds Weather Data			17.39	14.72	19.75	16.93	1004	67	11.21	
3600	10000	-1.527089	53.816961	Leeds Weather Data			14.13	13.3	14.75	13.01	1020	54	4.97	
0	10000	-1.527089	53.816961	Leeds Weather Data			9.39	8.77	10.69	6.1	1025	81	6.3	
3600	10000	-1.527089	53.816961	Leeds Weather Data			18.62	17.5	20.85	17.95	1013	54	9.14	
3600	10000	-1.527089	53.816961	Leeds Weather Data			18.42	16.47	21.36	17.6	1013	49	7.52	
0	10000	-1.527089	53.816961	Leeds Weather Data			10.43	9.67	11.82	9.64	1014	81	7.31	
0	10000	-1.527089	53.816961	Leeds Weather Data			6.33	5.95	7.12	2.16	1004	87	4.33	
0	10000	-1.527089	53.816961	Leeds Weather Data			6.56	3.85	8.75	2.45	1007	93	5.51	

Table: projectRAW Leeds_10_year_Data (115,601 rows)

Figure 5.22 *Weather Table Preview*

There a list of commands executed at each fact table to split the date-time datatype into data and time separately

1. Navigation = Source{[Schema="projectRAW",Item="Leeds_10_year_Data"]}[Data]
2. Changed Type = Table.TransformColumnTypes(projectRAW_Leeds_10_year_Data,{{"Date_time", type datetime}})
3. Split column by delimiter =Table.SplitColumn(Table.TransformColumnTypes(#"Changed Type", {{"Date_time", type text}}, "en-IN"), "Date_time", Splitter.SplitTextByDelimiter(" ", QuoteStyle.None), {"Date_time.1", "Date_time.2"})
4. Change Type = = Table.TransformColumnTypes(#"Split Column by Delimiter”, {{"Date_time.1", type date}, {"Date_time.2", type time}})

Once these columns are split the joins would start to work and the graphs can be plotted.

5.7 Power Bi Graphs

Each of the visuals used in this dashboard have given a dynamic analytical explanation for all of the fields from the tables. Rain_1h visual gives an overview of how the weather (weather description) has been on that day. The visual used here is a decomposition tree. This can be used by applying the filter mentioned above and can be filtered by year, month and day. It also gives the data for a whole month or a whole year if needed when all the options are not selected.

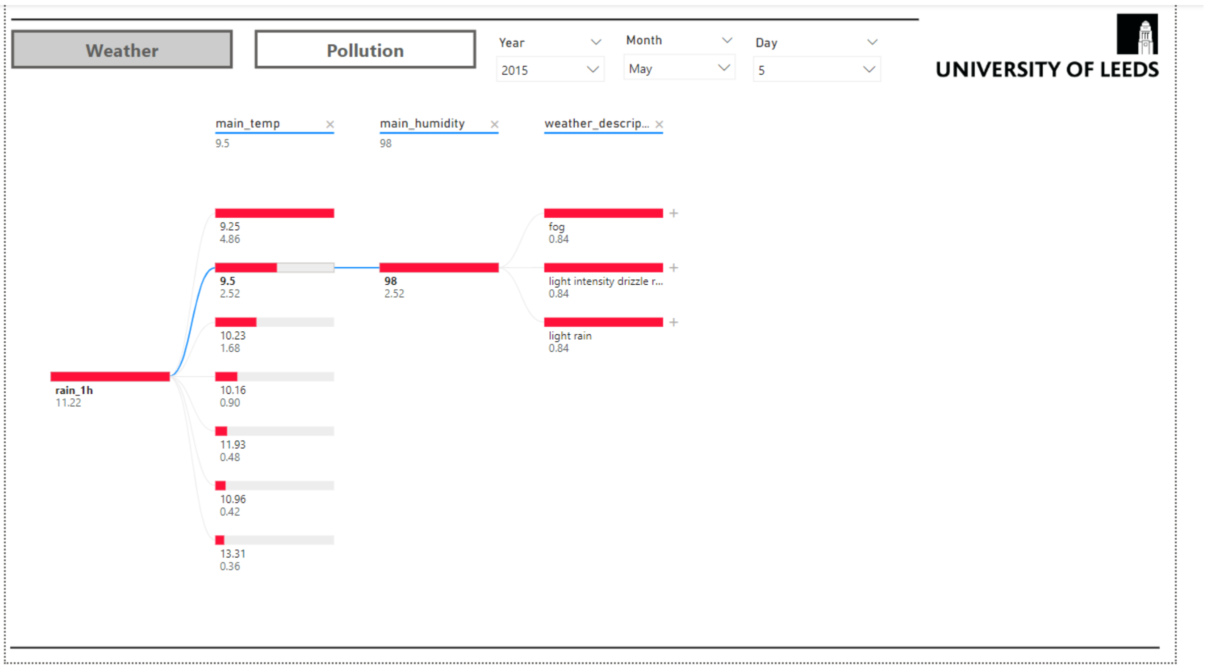


Figure 5.23 *Rain_1h visual*

In figure 4.23, the net pollutant levels are observed through time. The visual used here is a line graph with a secondary y-axis because the Carbon monoxide values are higher.

Qualitative name	Index	Pollutant concentration in μg/m^3			
		NO_2	PM_{10}	O_3	PM_{25} (optional)
Good	1	0-50	0-25	0-60	0-15
Fair	2	50-100	25-50	60-120	15-30
Moderate	3	100-200	50-90	120-180	30-55
Poor	4	200-400	90-180	180-240	55-110
Very Poor	5	>400	>180	>240	>110

Figure 5.24 *Pollutants Limits [11]*

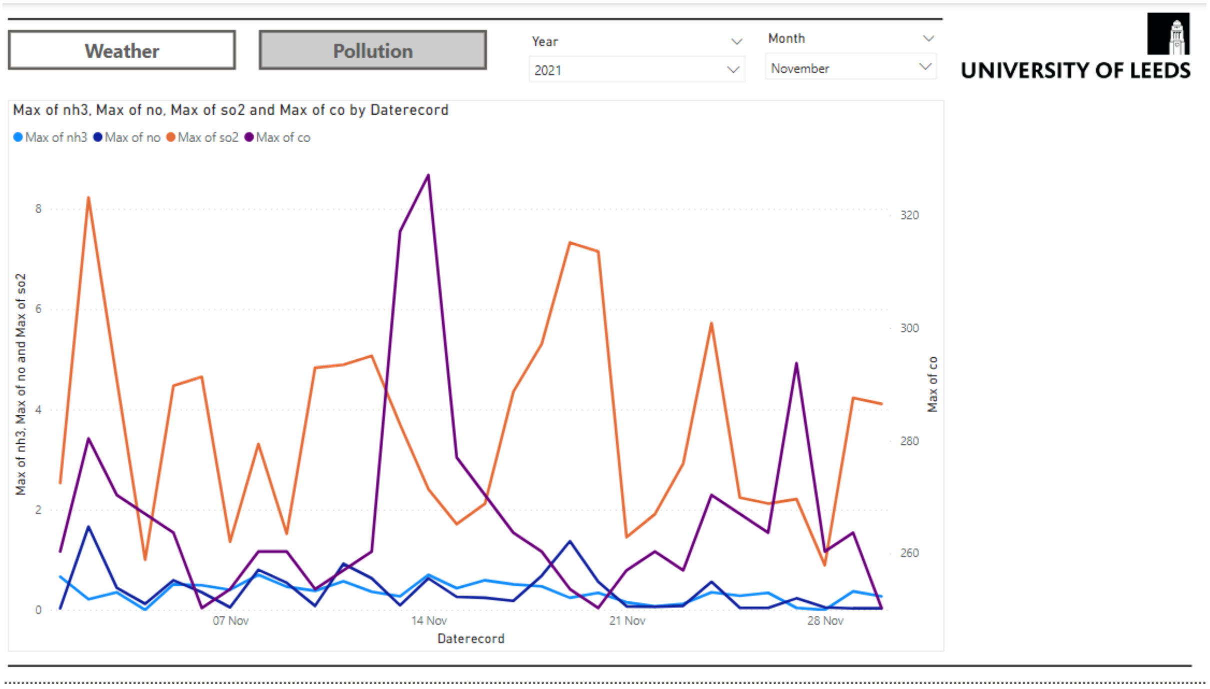
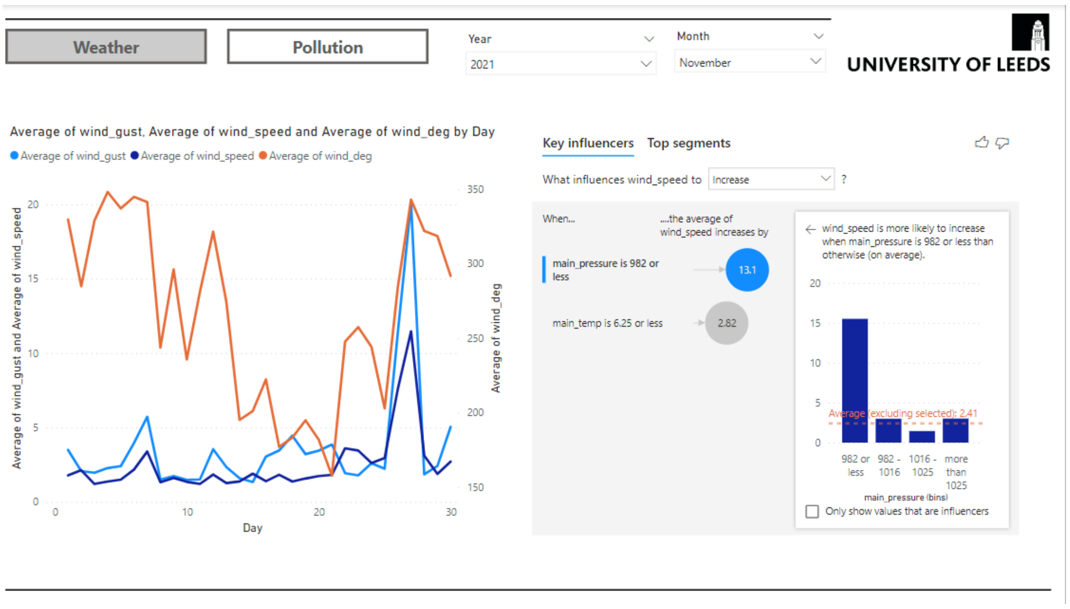

Weather
Pollution
Year
2021
Month
November
UNIVERSITY OF LEEDS
Max of nh3, Max of no, Max of so2 and Max of co by Daterecord
● Max of nh3 ● Max of no ● Max of so2 ● Max of co
Max of nh3, Max of no and Max of so2
0
2
4
6
8
Max of co
260
280
300
320
07 Nov
14 Nov
21 Nov
28 Nov
Daterecord

Figure 5.25 *Pollutants vs Time visual*

Figure 4.24 shows 2 graphs which is comprised of 2 visuals. The graph is a line graph with a secondary y-axis and the other visual is a key influencer visual. The key influencer visual helps to analyse based on explanation from different fields. In this visual, the analysis is for the wind speed and the explanation is given by temperature and pressure.


Weather
Pollution
Year
2021
Month
November
UNIVERSITY OF LEEDS
Average of wind_gust, Average of wind_speed and Average of wind_deg by Day
● Average of wind_gust ● Average of wind_speed ● Average of wind_deg
Average of wind_gust and Average of wind_speed
Average of wind_deg
Day
0
5
10
15
20
150
200
250
300
350
0
10
20
30
Key influencers
Top segments
main_pressure (bins)
982 or less
982 - 1016
1016 - 1025
more than 1025
Average (excluding selected): 2.41
20
15
10
5
0

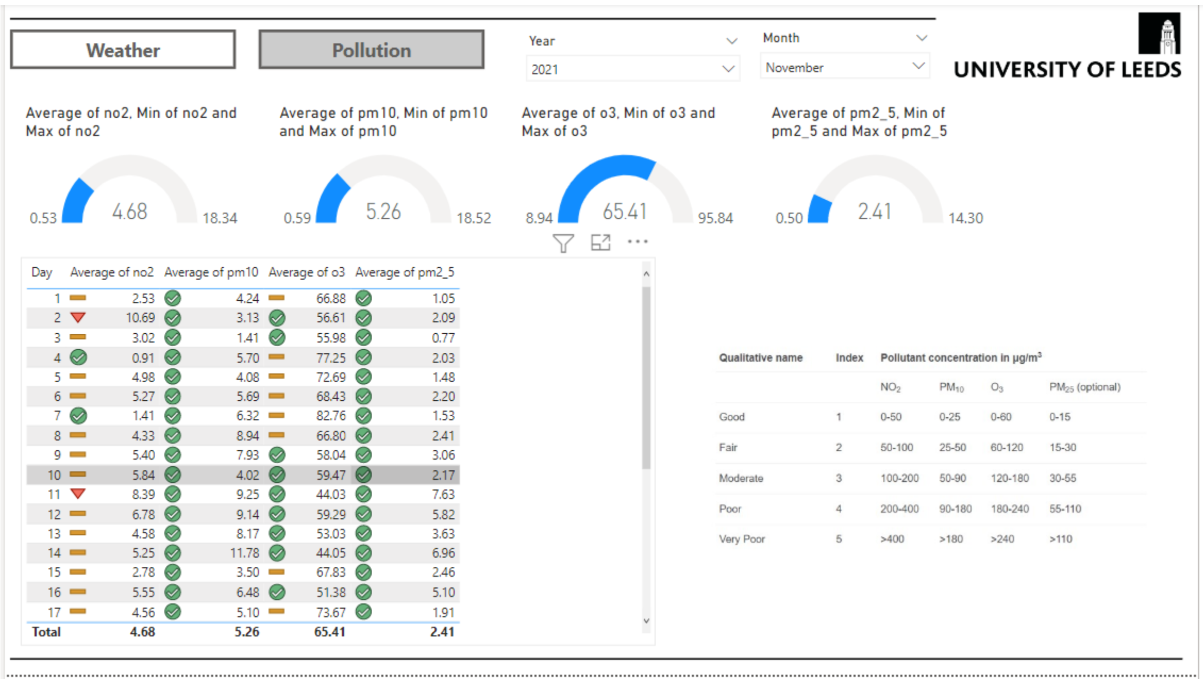
What influences wind_speed to Increase ?

When...	....the average of wind_speed increases by
main_pressure is 982 or less	13.1
main_temp is 6.25 or less	2.82

wind_speed is more likely to increase when main_pressure is 982 or less than otherwise (on average).

Only show values that are influencers

Figure 5.26 *Pollutants vs Time visual*



Average of no2, Min of no2 and Max of no2	Average of pm10, Min of pm10 and Max of pm10	Average of o3, Min of o3 and Max of o3	Average of pm2_5, Min of pm2_5 and Max of pm2_5
0.53 / 4.68 / 18.34	0.59 / 5.26 / 18.52	8.94 / 65.41 / 95.84	0.50 / 2.41 / 14.30

Day	Average of no2	Average of pm10	Average of o3	Average of pm2_5
1	▬ 2.53	✓ 4.24	▬ 66.88	✓ 1.05
2	▼ 10.69	✓ 3.13	✓ 56.61	✓ 2.09
3	▬ 3.02	✓ 1.41	✓ 55.98	✓ 0.77
4	✓ 0.91	✓ 5.70	▬ 77.25	✓ 2.03
5	▬ 4.98	✓ 4.08	▬ 72.69	✓ 1.48
6	▬ 5.27	✓ 5.69	▬ 68.43	✓ 2.20
7	✓ 1.41	✓ 6.32	▬ 82.76	✓ 1.53
8	▬ 4.33	✓ 8.94	▬ 66.80	✓ 2.41
9	▬ 5.40	✓ 7.93	✓ 58.04	✓ 3.06
10	▬ 5.84	✓ 4.02	✓ 59.47	✓ 2.17
11	▼ 8.39	✓ 9.25	✓ 44.03	✓ 7.63
12	▬ 6.78	✓ 9.14	✓ 59.29	✓ 5.82
13	▬ 4.58	✓ 8.17	✓ 53.03	✓ 3.63
14	▬ 5.25	✓ 11.78	✓ 44.05	✓ 6.96
15	▬ 2.78	✓ 3.50	▬ 67.83	✓ 2.46
16	▬ 5.55	✓ 6.48	✓ 51.38	✓ 5.10
17	▬ 4.56	✓ 5.10	▬ 73.67	✓ 1.91
Total	**4.68**	**5.26**	**65.41**	**2.41**

Qualitative name	Index	Pollutant concentration in μg/m³			
		NO_2	PM_{10}	O_3	PM_{25} (optional)
Good	1	0-50	0-25	0-60	0-15
Fair	2	50-100	25-50	60-120	15-30
Moderate	3	100-200	50-90	120-180	30-55
Poor	4	200-400	90-180	180-240	55-110
Very Poor	5	>400	>180	>240	>110

Figure 5.26 *Pollutant Performance visual based on limits from Figure 5.24*

Chapter 6
System Validation and Evaluation

This chapter provides checks and balances taken ensure the integrity of the whole system. It goes through the evaluation and validation. The reports were tested by professionals and some other users for non-functional understanding. The feedback was provided for the backend by professionals and the changes were made accordingly.

The system validation was conducted twice. By the first iteration, the backend was almost completed, and the reports were still pending as there was lot of knowledge on the APIs that had to be understood to produce better analysis based on the fields given.

The approach taken to provide the feedback that would benefit for an exploratory software based on cloud would be to perform runs of the same process multiple times just to see if any data is being duplicated or noticed to be any leak is observed in the pipeline.

6.1 System user testing - First iteration

Figure 4.1 explains the system architecture and the flow at which data is carried from one resource to another. The first iteration was made to ensure whether the dataflow from the databricks to the SQL database is running fine or some anomalies are to be noticed. This was achieved through running the notebook at the pipelines multiple times and overwriting the storage account with a new file and truncating the tables in the database and triggering the pipeline again to see the if the data received matches the previous time the pipeline ran. Some errors were noticed in the first iteration. The PowerBi report was working able to load the dataset without any required transformation. After the 2 datasets were loaded, joining them was the next error. To overcome that error a separate dimension table for date had to be created so, both the tables can use the aggregation values and can analyse weather results based on pollution. The backend had the following steps that had to be followed for system user testing:

1. See if there are any duplicates in the whole databricks notebook when the JSON file is converted to a dataframe.
2. Once the dataframe is converted, download the file from storage account to the local and check if the count is the same and there is in the dataframe.
3. After conversion, validate the column in the datafactory to see if there are mapping with the same name with the matching datatype.
4. After the pipeline is published, run the pipeline and check the empty database table to see all the values have arrived and the count is matching.

The result of the first iteration is summarised below:

1. The backend was working fine but the front end lacked clarity on the visuals.
2. The data is moving successfully even after the applied transformation, there was no leak or duplication of information.
3. The dataset used doesn’t violate any privacy policy.
4. Visualisation was done separately rather than analysing both factors together.

6.2 System user testing - Second Iteration

After the feedback from the first iteration there were a lot more changes made in the front end and the date dimension table was added to the database. The changes made are as follows:

1. Adding Date Dimension table for joining the two datasets
2. Creating visuals that help in not only projecting but also analysing the necessary fields.
3. One single filter in a page should be working for both weather visuals and pollution visuals
4. All of the page level filters should be in sync with the other filters in other pages.

After these changes were made the feedback received from the user’s side as are as follows:

1. Make the necessary UI changes to differentiate between both the graphs by correcting colours and making sure that when it’s a weather graph it is represented and when it’s a pollution graph it is represented accordingly.
2. Key influencers visual must need more fields to analyse for deriving better results.

Chapter 7
Conclusion and Future Work

This chapter aims in the highlighting the achievements, whether the aim of the project is being met and moreover discusses the future scope that this report can be used for and made better.

The purpose of this project was to explore the field of weather and forecasting through API technology as it is a very popular source of communication, and it can have many use cases. Data Engineering and BI development are technologies that have been around for a long time, but the technology stack has tremendously changed and made it more user friendly, cloud based and dynamic. Due to the rise of Azure and cloud service providers, students and developers can get hold of resources work on trial subscriptions and pay-as-you-go models. These resources really encouraged to explore many of the resources and security compliance standards while working on a cloud. Communication between resources becomes much easier through secrets and certificates that can obtained through the key vault.

7.1 Achievements

The initial tasks for the projects have all been completed.

- The datasets for the bulk APIs have been downloaded and stored in the local.
- Went through relevant research material for data warehousing as well front-end web app to show a weather forecast just to understand the new technologies present in the market and what benefits my system might have
- SQL server with the database tables is created for the values to be stored in.
- Non-functional requirements were addressed to match the match the initial standards set in the beginning.
- Validation was done through systems user testing where all the front end and the backend were treated like 2 separate entities to provide separate feedback.

7.2 Limitations and Future Work

There were more APIs that could yield better results like the solar radiation APIs which can derive results for the rise in temperature. Unfortunately, a heavy price has to be paid to access the particular API. During COVID there was a government shutdown and a lot of the daily use products that release carbon emissions were not functioning. A lot of the industries have not been in full attendance of workforce during that time. Due to this reason most of

2020 and some of 2021 projected some figures that were hard to believe if they were forecasted 3 years ago. This may result in future values to provide not fully accurate results because of the lack of carbon footprint and pollutants during COVID. One of the limitations was that Leeds is the city chosen for this project, but a lot of the places are noticed to have where population is denser like Asian countries. There might be more factors why a lot less change is noticed in a city like Leeds. This is when the future scope comes into picture.

There are some suggestions that can be used for future work:

1. Expand the reach from Leeds to very famous cities across the world like New York, London, Delhi, etc.
2. Lot more manmade issues are taking part in global warming. To access this, there must be more field values where this should be looked upon like population, vehicle emissions.
3. Altitude is one factor which was not put into consideration for this project because it was just fixed to one point.
4. Using live values including the daily forecast APIs. These can be used for multiple for correction when the actual values appear.
5. Encourage the open weather service providers to provide other new APIs that help in deriving more results for the reports.
6. The report can extend its reach by creating a mobile application so, viewing some the information on phones is much more quicker and user friendly that compared to a PC.

7.3 Challenges

One of the main challenges faced was for creating the spark cluster. The configuration used was basic as it was under the student subscription which was initially a big problem to resolve. The rate at which the data has to be refreshed is another problem that has to be put into consideration. There is no pipeline that is triggered through a logic app. In the future a logic app would be necessary for triggering these pipelines if the new data has to be refreshed.

List Of References

[1] An Overview of Data Warehousing and OLAP Technology Surajit Chaudhuri Umeshwar Dayal Microsoft Research, Redmond, Hewlett-Packard Labs, Palo Alto

[2] Microsoft Azure : Cloud Platform for Application Service Development Varsha Desai V.P.Institute of Management Studies & Research, Sangli, Kavita Oza Shivaji University, Kolhapur, Poornima Naik CHHATRAPATI SHAHU INSTITUTE OF BUSINESS EDUCATION AND RESEARCH, Priyanka Shinde Government College of Engineering, October 2018

[3] Development Of A REST API For Obtaining Site-Specific Historical And Near-Future Weather Data In EPW Format Hu Du1 , Phillip Jones1 , Eva Lucas Segarra2 , Carlos Fernández Bandera2 1Welsh School of Architecture, Cardiff University, Cardiff, United Kingdom 2School of Architecture, University of Navarra, Pamplona, Spain DuH4@cardiff.ac.uk

[4] Weather Data Forecast and Analytics Vishal D. Wavhale, Vivek Kumar, Vidit Raj Choudhari, Saurav Bira August 2020

[5] https://spark.apache.org/docs/latest/cluster-overview.html

[6] https://docs.microsoft.com/en-us/azure/data-factory/load-azure-data-lake-storage-gen2

[7] https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-databricks-delta-lake?tabs=data-factory

[8] https://docs.microsoft.com/en-us/azure/databricks/security/secrets/secret-scopes

[9] https://openweathermap.org/api/solar-radiation

[10] https://openweathermap.org/history

[11] https://openweathermap.org/api/air-pollution

[12] https://www.clouddefense.ai/blog/system-development-life-cycle

[13] Data Warehousing Fundamentals by Paulraj Ronniah

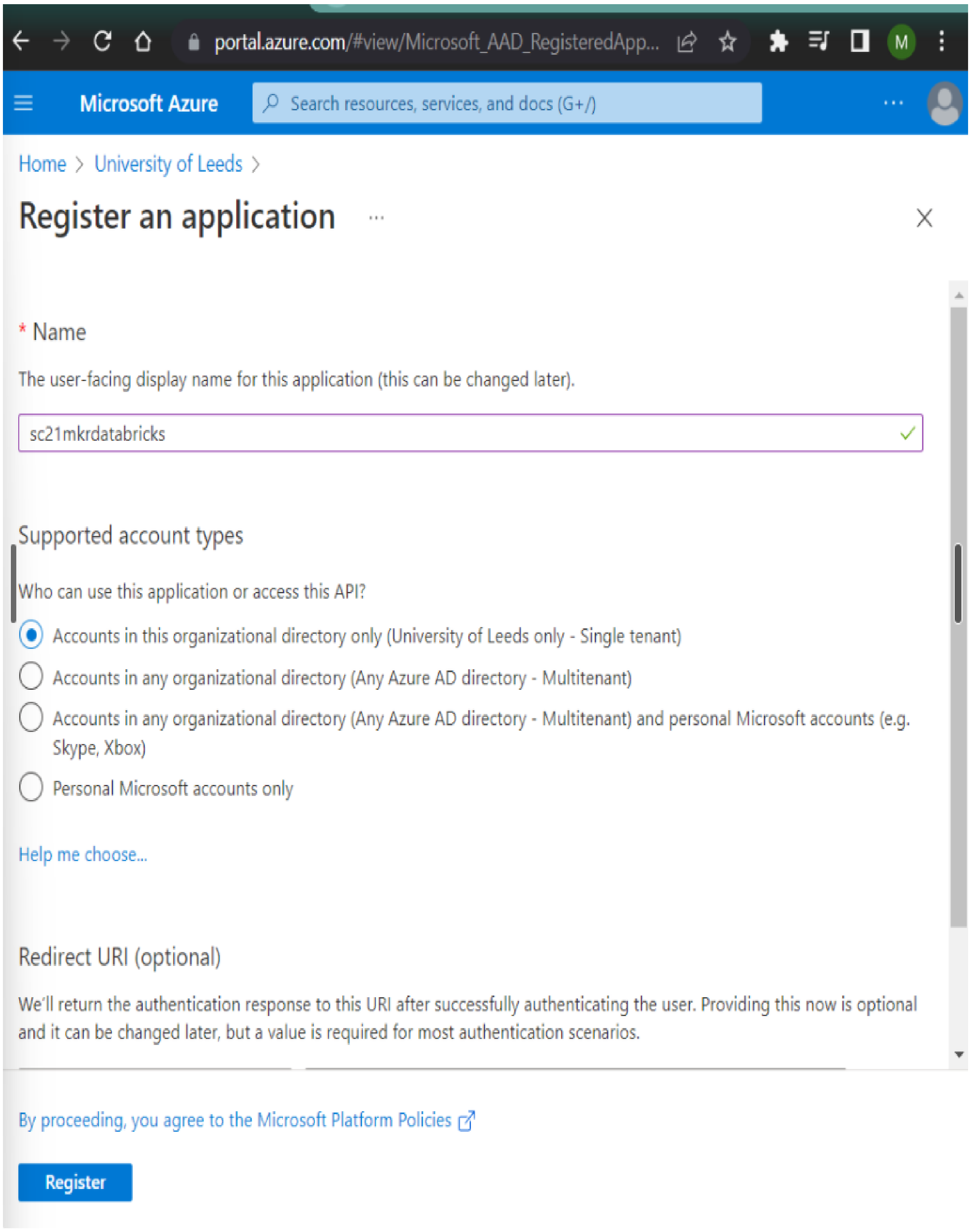
[14] Sustainable energy management of solar greenhouses using open weather data on MACQU platform Li Li , Jieyu Li , Haihua Wang , Ts. Georgieva, K. P. Ferentinos, K. G. Arvanitis , N. A. Sigrimis August 2018

[15] Agricultural Management and Climatic Change Are the Major Drivers of Biodiversity Change in the UK March 2016

Appendix A
External Materials

A.1 The steps of actions taken for the mounting of storage account to databricks.

1. Registering in Active directory


portal.azure.com/#view/Microsoft_AAD_RegisteredApp...
Microsoft Azure
Search resources, services, and docs (G+/)
Home > University of Leeds >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

sc21mkrdatabricks ✓

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (University of Leeds only - Single tenant)]
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

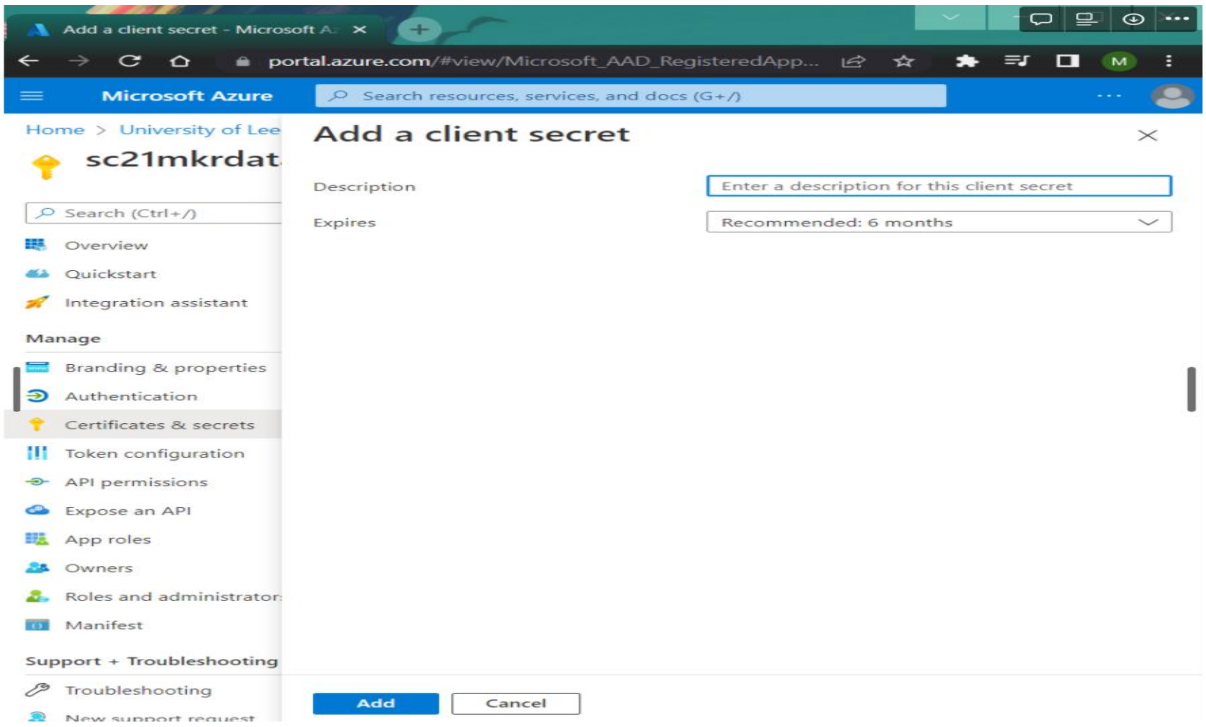
Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

By proceeding, you agree to the Microsoft Platform Policies

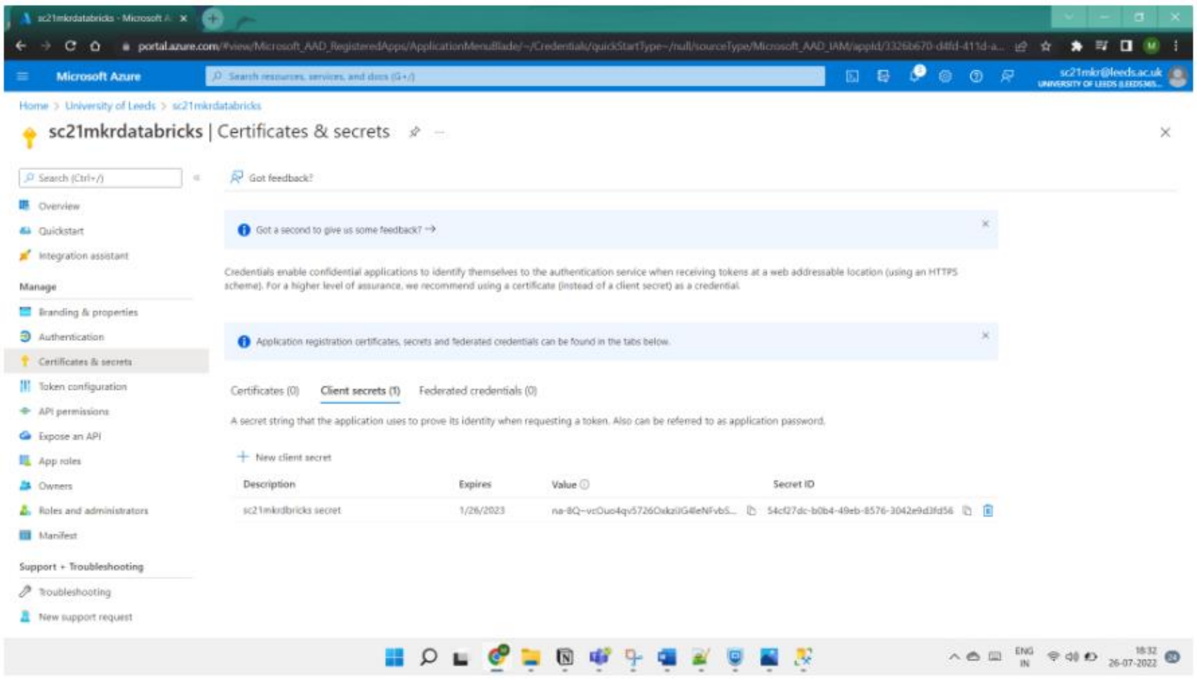
2. Adding client ID for databricks as secret.



Add a client secret

Description: Enter a description for this client secret

Expires: Recommended: 6 months



sc21mkrdatabricks | Certificates & secrets

Got a second to give us some feedback? →

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

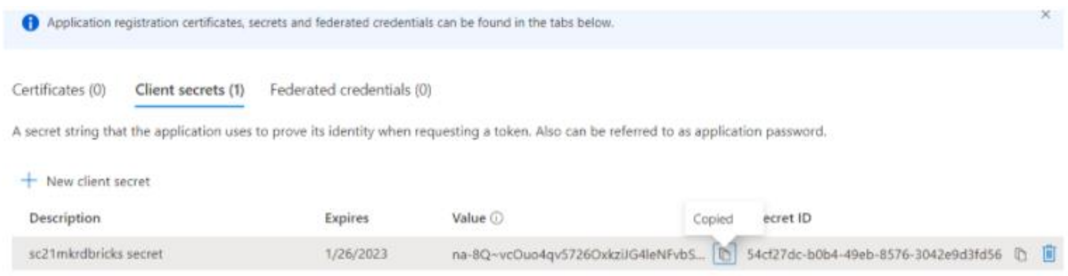
Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) | Client secrets (1) | Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
sc21mkrdbricks secret	1/26/2023	na-8Q~vcOuo4qv5726OxkziJG4leNFvbS...	54cf27dc-b0b4-49eb-8576-3042e9d3fd56



Application registration certificates, secrets and federated credentials can be found in the tabs below.

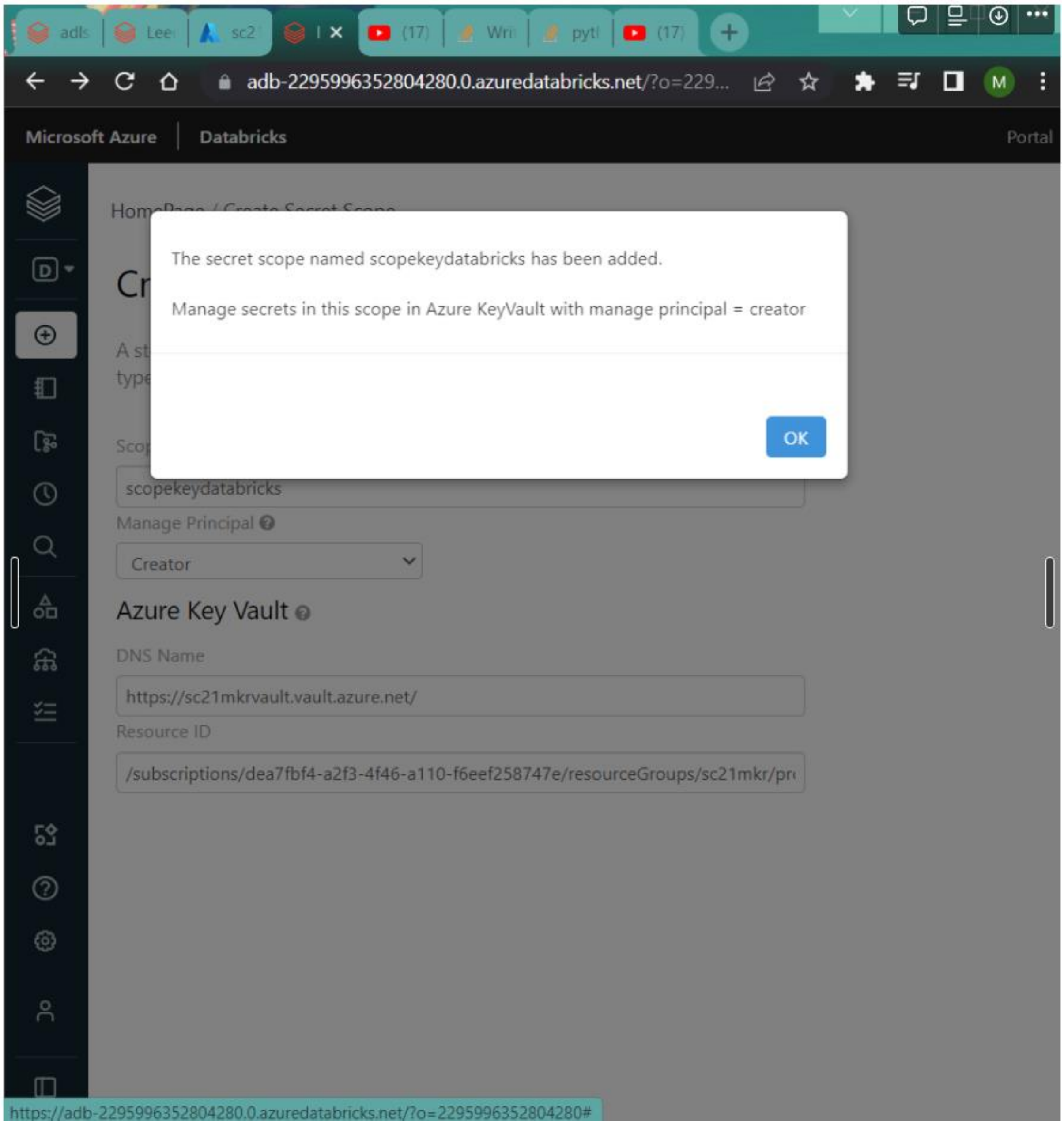
Certificates (0) | Client secrets (1) | Federated credentials (0)

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

+ New client secret

Description	Expires	Value	Secret ID
sc21mkrdbricks secret	1/26/2023	na-8Q~vcOuo4qv5726OxkziJG4leNFvbS...	54cf27dc-b0b4-49eb-8576-3042e9d3fd56

3. Creating scope in spark cluster to ensure the mounted storage account can be used.



adb-2295996352804280.0.azuredatabricks.net/?o=229...

Microsoft Azure | Databricks

The secret scope named scopekeydatabricks has been added.

Manage secrets in this scope in Azure KeyVault with manage principal = creator

Scope Name: scopekeydatabricks

Manage Principal: Creator

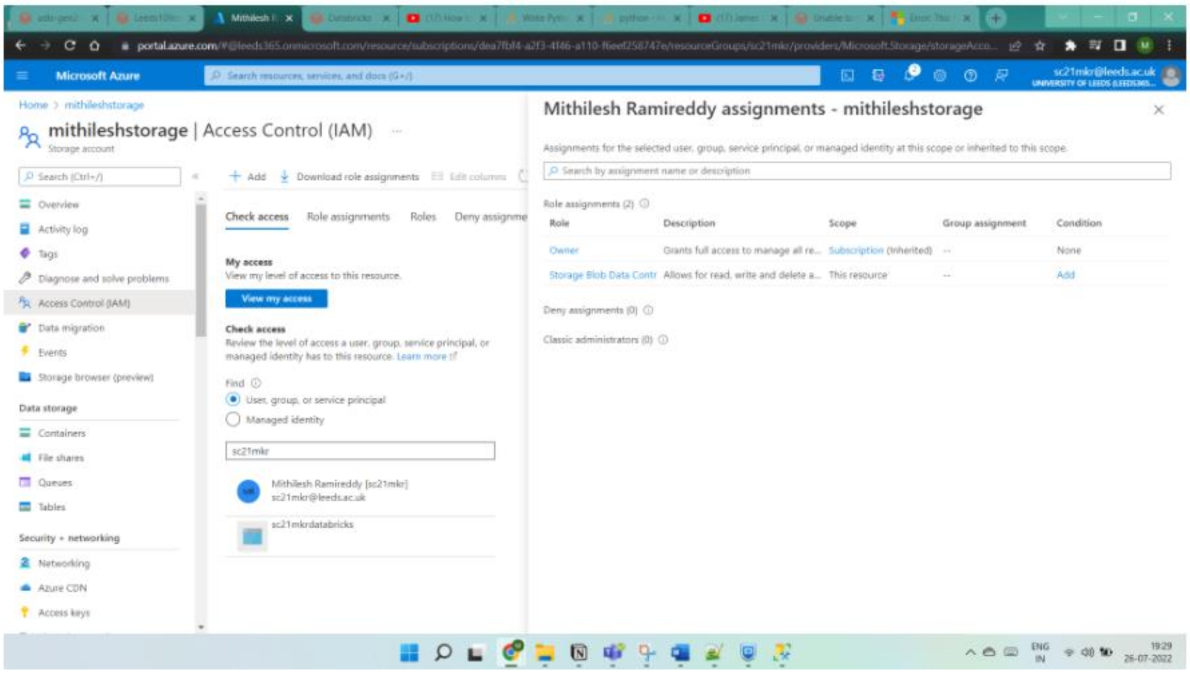
Azure Key Vault

DNS Name: https://sc21mkrvault.vault.azure.net/

Resource ID: /subscriptions/dea7fbf4-a2f3-4f46-a110-f6eef258747e/resourceGroups/sc21mkr/pr

https://adb-2295996352804280.0.azuredatabricks.net/?o=2295996352804280#

4. Add new Access control as Storage Blob contributor


mithileshstorage | Access Control (IAM)
Storage account

Check access

My access
View my level of access to this resource.

Check access
Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more

Find
(●) User, group, or service principal
( ) Managed identity

sc21mkr

Mithilesh Ramireddy [sc21mkr]
sc21mkr@leeds.ac.uk

sc21mkrdatabricks

Mithilesh Ramireddy assignments - mithileshstorage

Assignments for the selected user, group, service principal, or managed identity at this scope or inherited to this scope.

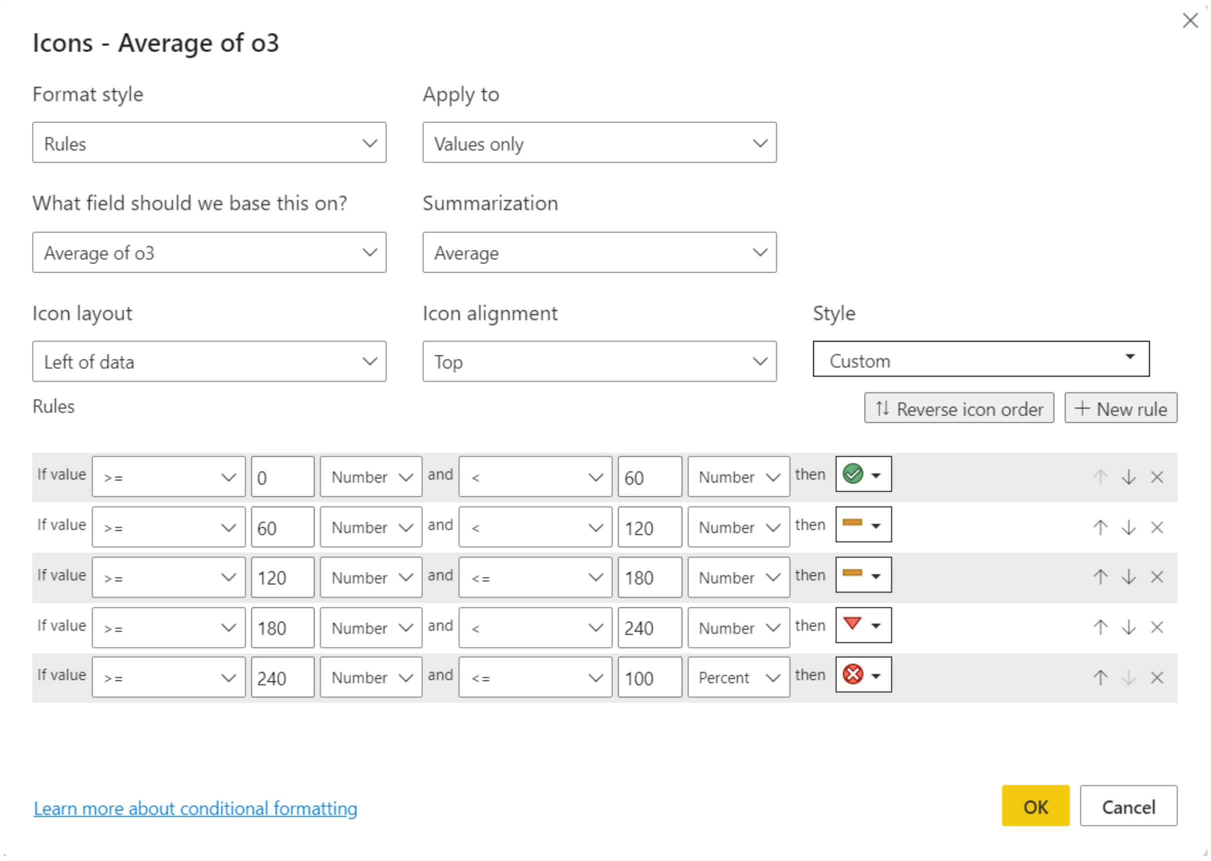
Role assignments (2)

Role	Description	Scope	Group assignment	Condition
Owner	Grants full access to manage all re...	Subscription (Inherited)	--	None
Storage Blob Data Contr	Allows for read, write and delete a...	This resource	--	Add

Deny assignments (0)

Classic administrators (0)

A.2 Conditional formatting for PowerBi visual


Icons - Average of o3

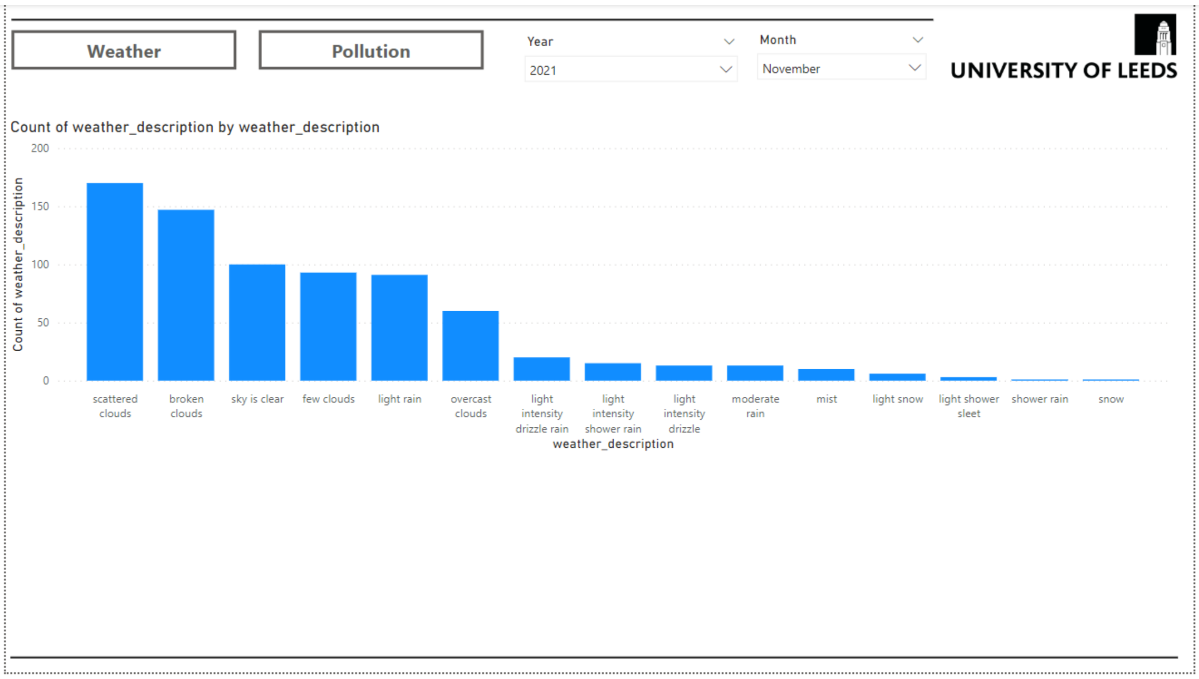
Format style: Rules
Apply to: Values only
What field should we base this on?: Average of o3
Summarization: Average
Icon layout: Left of data
Icon alignment: Top
Style: Custom

Rules

If value	>=	0	Number	and	<	60	Number	then	✓
If value	>=	60	Number	and	<	120	Number	then	▬
If value	>=	120	Number	and	<=	180	Number	then	▬
If value	>=	180	Number	and	<	240	Number	then	▼
If value	>=	240	Number	and	<=	100	Percent	then	⊗

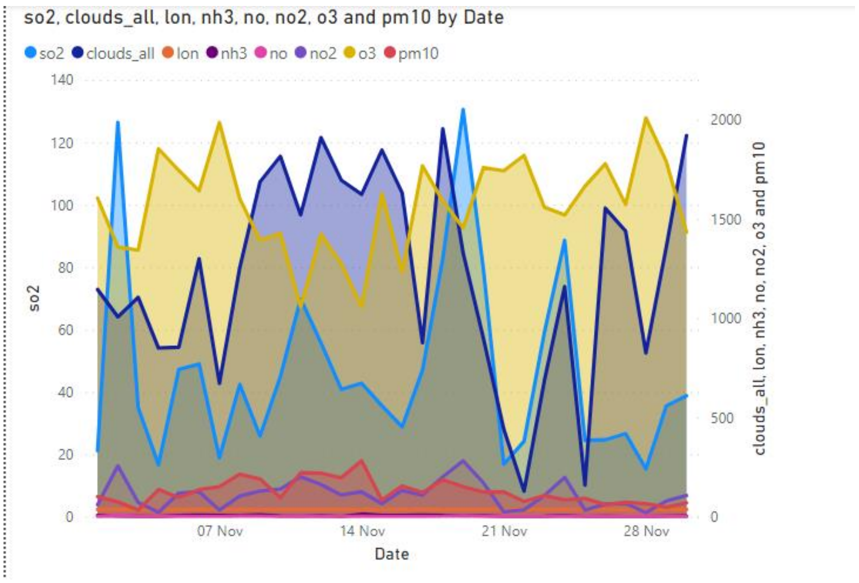
Learn more about conditional formatting

A.3 Hourly weather count



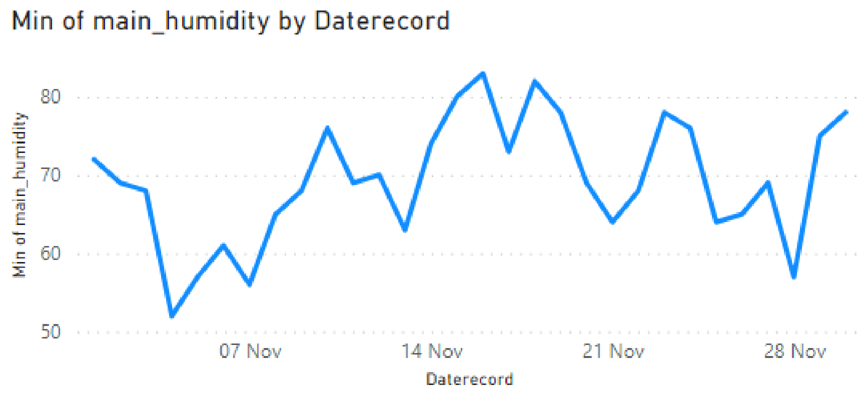
The above graph is a description of the number of times it was reported rain or not based on description only for the month of November. Using the filter, the month and year can be adjusted.

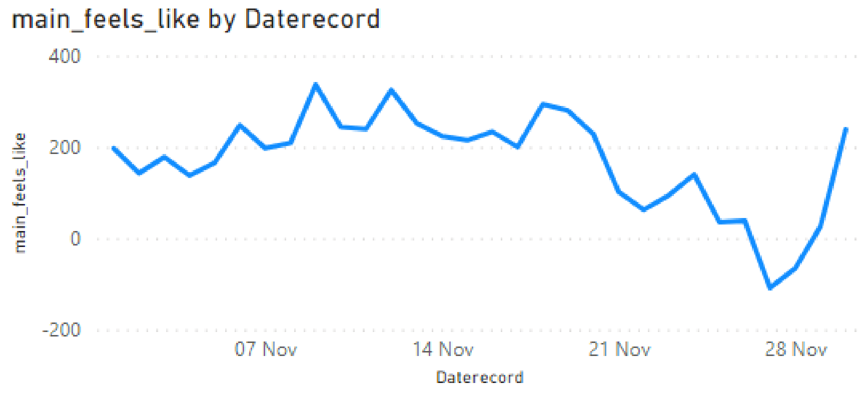
A.4 Pollutants vs clouds



This graph was an attempt to co relate between the pollutants and how the clouds would look like based on emissions. There was not a lot of understanding attained from this visual.

A.5 Humidity vs Feels like Temperature





This visual was an attempt to fins an inverse relationship between the humidity and temperature. By this graph there can a conclusion drawn that during the winter the humidity doesn't help in feeling warmer whereas in the summer the humidity increase is a total increase in overall temperature.