COMP90024 Assignment 2: Team 45

Ojaswi Dheer 1447227 Petr Andreev 1375858
Rafsan Al Mamun 1407776 William Chen 1400081

May 21, 2024

GitHub Repository Link: https://github.com/Rafsan7238/COMP90024_Assignment-2
YouTube Video Link: https://youtu.be/fhftcsxGKWA

Table of Contents

1 Introduction

In the modern world, the flow of data is rapid and voluminous, presenting both challenges and opportunities for businesses and data analysts. Organizations often struggle to keep pace with the constant influx of information, which can obscure insights into customer needs and market trends. However, that same data deluge provides a fertile ground for data analysts to uncover patterns and insights that were previously unattainable. By leveraging technologies such as cloud computing, businesses can transform this challenge into a strategic advantage [1].

This project aims to showcase the potential of cloud computing for data analytics. By utilizing a commonplace technology stack for cloud infrastructure, we will demonstrate how these tools can be used to analyze and interpret complex data sets. The focus of our analysis will be on the livelihoods and well-being of Australians, examined through the lenses of physical and mental health.

The project will specifically explore two key areas:

1. **Lung-Related Disorders and Air Quality:** Lung-related disorders such as asthma, chronic obstructive pulmonary disease (COPD), and lung cancer significantly impact public health in Australia [2]. We will correlate air quality to health records to identify trends and potential risk factors that contribute to lung-related disorders. Such information could inform policymakers and healthcare providers when considering effective interventions and public health improvements.

2. **Weather and Mood in Social Media:** Weather has been known to influence human emotions and behavior. In the digital age, social media platforms offer a rich source of data for analyzing public sentiment. By using media sentiment as an indicator of population mental health, and correlating this to weather, we gauge the psychological effects of weather.

All in all, the project will demonstrate the efficacy of cloud computing in performing data analyses that can inform decision-making.

2 Technology Stack

The technology stack used in this project is Fission [3] for data ingestion and ElasticSearch [4] for data storage and retrieval. These are orchestrated using Kubernetes [5] within the Melbourne Research Cloud (MRC) [6]. The MRC itself is based off the OpenStack [7] environment. Finally, to visualise our analytics, we use Jupyter Notebook [8]. All components to this stack are open-source.

2.1 MRC and OpenStack

OpenStack is a cloud computing platform that provides infrastructure as a service by aiding in the management of virtual machines. It does this by simplifying the management of cloud resources such as compute, storage, and networking. The MRC is based off OpenStack and is a privately hosted resource for University of Melbourne (and affiliate) research.

2.2 Kubernetes

Each component of our technology stack exists in multiple containers that interact with one another. To simplify the orchestration, deployment and scaling of these container, we utilise Kubernetes. Kubernetes suits the use case of our project as it will simplify development by abstracting complex container management into a simpler framework. For example, it controls the automatic recovery of container in the case of failure and integrates with other technologies to enable automatic horizontal scaling. This saves us time by not having to manage network and communication between containers.

2.3 Fission

Fission is a framework that streamlines the development of serverless functions on Kubernetes. It allows the creation of short-lived functions that are mapped to HTTP requests (or other event triggers). Furthermore, developers can deploy functions instantly with single commands, and Fission also works with Kubernetes to manage individual pods that invoke these functions.

2.4 ElasticSearch

ElasticSearch is an analytics engine that provides horizontal scalability and strong search capabilities. This synergises well with cluster cloud computing infrastructures like Kubernetes, as it can be used to deploy containers for ElasticSearch, which makes actioning ElasticSearch horizontal scaling a simpler process. In this project, it will also serve as a form of data storage.

2.5 Jupyter Notebook

Jupyter Notebook is an open-source web application that streamlines the creation of visualizations. It is a versatile tool that can simultaneously run Python code and display graphical output. This makes Jupyter Notebooks suitable for the rapid development of data analytics displays given RESTful API data access.

3 Scenarios

The aim of this project is to use data analytics to capture certain aspects of life in Australia with the help of the cloud stack described in the previous section. For our analysis, we selected two particular scenarios:

1. What is the relationship between air quality and lung-related disorders, e.g. asthma, COPD?
2. What impact does the weather have on people's mental health?

3.1 Air Quality and Lung-related Disorders in Victoria

In this analysis, we explore the correlation between air quality and lung-related disorders in Victoria. The chosen scenario investigates whether there is a significant relationship between air pollution levels and the prevalence of lung diseases and their related mortality rates. The lung diseases of interest are asthma, COPD, and lung cancer. The context is within the space of Australian lifestyle and environmental conditions.

Australia, known for its diverse landscapes and thriving urban centers, also suffers from air quality related challenges. This is due to industrial activities, transportation emissions, and urbanization. With large populations residing in metropolitan areas, exposure to air pollution has become a growing concern for public health. According to recent reports from the Australian Institute of Health and Welfare (AIHW), respiratory diseases causes substantial burden on the nation's health, with conditions such as asthma and COPD affecting a significant portion of the population [9]. In Australia, it's linked to more than 3,200 deaths a year [9] at an estimated cost of A$6.2 billion [10].

Hence, this scenario is motivated by the need to understand the relationship between air pollution and respiratory health outcomes, given the potential implications for public policy and health interventions.

3.1.1 Motivations

To underscore the importance of this investigation, consider the following statistics and evidence:

- **Disease Burden:** Respiratory diseases accounted for over 10% of the total disease burden in Australia in 2020, with asthma being one of the most prevalent conditions [9].
- **Association with Air Pollution:** Numerous epidemiological studies have demonstrated associations between air pollution exposure and adverse respiratory health outcomes. For instance, exposure to fine particulate matter (PM2.5) and nitrogen dioxide (NO2) has been linked to increased incidence of asthma and exacerbation of COPD [11].
- **Mortality Rates:** Long-term exposure to air pollutants has been associated with higher mortality rates from respiratory diseases, highlighting the profound impact of air quality on public health [12]. During the 2019-20 Black Summer fires, the smoke alone was linked to 400 deaths and 4,500 hospitalisations and emergency department visits [13].

PM2.5, defined as fine particulate matter with aerodynamic diameter ≤ 2.5 micrometers, has been linked to various respiratory and cardiovascular diseases [14]. For instance, a meta-analysis by Hoek et al. [15] reported a significant increase in respiratory mortality with elevated levels of PM2.5. Moreover, research by Thurston et al. [16] identified PM2.5 as a key risk factor for the development of COPD and other chronic lung diseases. Given the compelling evidence linking PM2.5 exposure to lung diseases, deeper analysis in this scenario will focus on this particular parameter.

3.1.2 Scenario Objectives

The primary objective of this investigation is to determine if there exists a statistically significant correlation between air quality indicators and the prevalence/mortality-rates of lung-related disorders in Victoria. By addressing this objective, the analysis aims to achieve the following:

- **Quantify the Impact:** Evaluate the extent to which air quality contributes to the burden of lung-related diseases in different regions.
- **Raise Public Awareness:** Increase public awareness about the link between air quality and respiratory health, promoting preventive measures and individual actions to reduce exposure to pollutants.
- **Focus on PM2.5:** Since PM2.5 has the strongest evidence linking it to lung diseases, this scenario will focus on this parameter when conducting deeper analysis.

3.1.3 Methodology and Analytical Approach

To achieve the objectives outlined above, the analysis will employ the following methodology:

1. **Data Collection:** Gather comprehensive datasets on air quality parameters from national air monitoring stations for Victoria gathered from the Environment Protection Authority Victoria (EPA) [17] and epidemiological data on lung-related disorders from the Spatial Urban Data Observatory (SUDO) [18].
2. **Statistical Analysis:** Perform descriptive, exploratory, and correlation analysis, followed by regression modeling, and spatial analysis techniques.
3. **Data Visualization:** Utilize data visualization tools to visualize the analysis done.

To reiterate, our first scenario identifies and quantifies the associations between air quality and lung-related disorders and deaths in Victoria. It aims to inform evidence-based policies and interventions that promote cleaner air and better respiratory health outcomes for the population.

3.2 Weather and People's Mood in Australia

Weather has long been acknowledged as a significant influence on human behavior and emotions. Seasonal affective disorder (SAD) is a well-documented example of how changes in weather can impact mood [19]. From anecdotal evidence to scientific studies [20], research in this area has shown that various weather conditions, such as temperature, humidity, and sunlight, can have both direct and indirect effects on mental health and emotional states.

3.2.1 Motivations

- **Limited Research:** Early studies found associations between weather conditions and emotions, but were limited by small sample sizes. Subsequent studies in the most recent decade have found small to negligible associations, while others have documented associations that vary across individuals [21]. This limited and contradicting existing study motivates additional research into this area.
- **Regularly Updating Analytics:** Weather data is readily available on an up-to-date basis through various meteorological sources. Mental health can be extrapolated from sentiments in social media posts, which are also available in an up-to-date manner through social media websites like Mastodon. By combining the two together, there is an opportunity to produce a regularly updating analysis on the relationship between weather and mental health.

3.2.2 Scenario Objectives

The primary objective of this analysis is to investigate the potential correlation between weather conditions and mental health in Australia. By achieving this objective, we aim to providing valuable insights into how weather impacts the lives and well-being of Australians. We will also introduce a system to perform this correlation analysis on an hourly basis. By doing so, we contribute the power of cloud based data collection and analysis techniques to an area of research that, traditionally, has had limited available data.

3.2.3 Methodology and Analytical Approach

To achieve this scenario's objectives, the analysis will employ the following methodology:

1. **Historic Data Collection:** We will collect historical weather data from the Australian Bureau of Meteorology (BoM) between the years 2021 and 2022. The dataset to correlate with will be Twitter data from those same years, which includes the sentiments of tweets.

2. **Statistical Analysis:** The analysis to be done will involve correlations and line plots to showcase the relationship between the weather and social media sentiments.

3. **Hourly Analysis:** Finally, we will utilise the abilities of cloud based solutions to regularly grow a dataset of weather and sentiments from Mastodon. Weather data will be collected from the BoM website, and the social media records will be taken from Mastodon, both via their APIs. The result will be a comparative line graph that extends once every hour.

To reiterate, our second scenario identifies and quantifies the associations between weather values and social media sentiment in Australia. By extrapolating media sentiment to mental health, this scenario aims to contribute cloud based data collection analyses to seasonal depression research, a traditionally data-limited field.

4 Datasets

4.1 Air-Quality Data

The "EPA Air Watch All Sites Air Quality Hourly Averages - Yearly" dataset provides comprehensive air quality measurements obtained from EPA's network of air monitoring stations across Victoria [17]. This dataset presents 1-hourly average air quality parameters recorded at various monitoring locations throughout the year.

The data encapsulated within this annual report encompasses a detailed overview of air quality metrics across all stations, reflecting the diverse environmental conditions monitored by the EPA [17]. At 76,5202 entries long, it serves as a powerful, albeit large, resource for analyzing and understanding air quality trends and variations throughout Victoria.

The dataset includes many recorded pollutants, including potentially interesting parameters: PM2.5, PM10, Carbon Monoxide (CO), Ozone (O3) and Sulphur Dioxide (SO2). Key among these is PM2.5, as mentioned in Section 3.1.1. The dataset has data spread across 8 columns as follows:

- **Date:** The date of the observation.
- **Time:** The time of the observation.
- **Location Name:** Name of the monitoring station where the measurement was recorded.
- **Latitude:** The y-coordinate of the monitoring station.
- **Longitude:** The x-coordinate of the monitoring station.
- **Parameter Name:** Name of the observed air-quality parameter.
- **Value:** Value of the observed parameter.
- **Parameter Description:** Description of the observed parameter.

4.2 Lung Disease Data

The datasets related to lung diseases across Australia have been all gathered from SUDO [18]. A total of 5 datasets were collected with information related to cancer mortality, asthma, and COPD prevalence.

4.2.1 Cancer Mortality Data

The dataset titled "AIHW - Cancer Incidence and Mortality Across Regions (CIMAR) - Males Mortality (GCCSA) 2009-2013" [18] presents detailed male cancer mortality statistics for Australia.

The dataset titled "AIHW - Cancer Incidence and Mortality Across Regions (CIMAR) - Females Mortality (GCCSA) 2009-2013" [18] offers comprehensive insights into female cancer mortality statistics across Australia.

The dataset "AIHW - Cancer Incidence and Mortality Across Regions (CIMAR) - Persons Mortality (GCCSA) 2009-2013" [18] gives higher level insights into cancer mortality statistics across Australia regardless of gender.

These three datasets contain data regarding many kinds of cancer, but we are focused on one in particular, lung cancer.

There are some key insights common to all the 3 aforementioned datasets:

- **Data Scope:** The datasets provide insights into the number of cancer-related deaths from the years 2009 to 2013.
- **Data Completeness:** Values assigned as "n.p." in the original data have been excluded from this dataset.

These datasets were filtered to only consist of lung cancer-related mortality across Australia, and now contains 8 rows of data each spread across 7 columns as follows:

- **GCCSA Code:** The Greater Capital City Statistical Areas code.
- **GCCSA Name:** The name of the GCCSA.
- **Lung Cancer Total Mortality:** Deaths related to lung cancer.
- **Lung Cancer Population:** Number of people diagnosed with lung cancer.
- **Lung Cancer Rate per 100K:** Number of people with lung cancer per 100,000 people.
- **All Cancer Total Mortality:** Deaths related to all types of cancer.
- **All Cancer Population:** Number of people diagnosed with at least one type of cancer.

4.2.2 Asthma and COPD Prevalence Data

The dataset "Type of Long-Term Health Condition by Selected Person Characteristics" contains insights into the prevalence of several long-term diseases among Australians [22]. The census has been compiled by the Australian Bureau of Statistics (ABS). Long-term health conditions, as defined in this dataset, are diagnosed by a healthcare professional and persist for six months or longer. These conditions may recur periodically, require ongoing medication for management, or be in a state of remission. The dataset includes a range of specific health conditions that fall under the long-term category, such as arthritis, asthma, dementia (including Alzheimer's), and many other specified long-term health conditions. For this project, we are interested in the values of asthma and COPD.

This dataset was filtered to only consist of data related to asthma and COPD. The filtered dataset had a total of 8 rows of data spread across multiple columns as follows:

- **GCCSA Code:** The Greater Capital City Statistical Areas code.
- **GCCSA Name:** The name of the GCCSA.

- **Australian Asthma/COPD:** Number of Australians diagnosed with asthma or COPD.
- **Foreigner Asthma/COPD:** Number of foreign nationals living in Australia diagnosed with asthma or COPD.
- **Employed Asthma/COPD:** Number of employed people diagnosed with asthma or COPD.
- **Unemployed Asthma/COPD:** Number of unemployed people diagnosed with asthma or COPD.
- **Assistance Needed Asthma/COPD:** Number of people diagnosed with asthma or COPD who require regular assistance by a carer.
- **Weekly Income Asthma/COPD:** Number of people diagnosed with asthma or COPD based on their weekly income: nil, A$1-299, A$300-649, A$650-999, A$1000-1749, A$1750-2999, A$3000+.
- **Total Asthma/COPD:** Total number of people diagnosed with asthma or COPD.

4.3 Weather Data

Both historical and real-time weather data was accessed from the BoM website [23].

4.3.1 Historic Rainfall Data

Rainfall includes rain, drizzle, hail, and snow, that fall from clouds and reach the ground. In Australia, rainfall data is primarily collected using rain gauges, which measure precipitation in millimeters. For historic data, observations of daily rainfall are typically made at 9 am local time, recording the total precipitation for the preceding 24 hours [24]. This data was collected in pre-aggregated monthly format.

- **Data Collection:** The historic rainfall data was collected from the following eight stations, each located in a different Australian state.
  - Melbourne Botanical Gardens (Station number: 86232) - Victoria
  - Adelaide (West Terrace / Ngayirdapira) (Station number: 23000) - South Australia
  - Brisbane (Station number: 40913) - Queensland
  - Canberra Parliament House (Station number: 70246) - Australian Capital Territory
  - Darwin Botanic Gardens (Station number: 14163) - Northern Territory
  - Perth Metro (Station number: 9225) - Western Australia
  - Sydney Botanic Gardens (Station number: 66006) - New South Wales
  - Hobart (Ellerslie Road) (Station number: 94029) - Tasmania

4.3.2 Historic Temperature Data

Air temperature is a critical parameter in understanding climate dynamics, affecting various aspects of life, including agriculture, health, and infrastructure. This report uses monthly mean maximum temperature data collected from eight different cities across Australia.

- **Data Collection:** The mean maximum temperature represents the average highest temperature recorded over a month. The data was collected from the following eight stations, each located in a different Australian state:
  - Melbourne (Olympic Park) (Station number: 86338) - Victoria
  - Adelaide (West Terrace / Ngayirdapira) (Station number: 23000) - South Australia
  - Brisbane (Station number: 40913) - Queensland

- Canberra Airport (Station number: 70351) - Australian Capital Territory
- Darwin Airport (Station number: 14015) - Northern Territory
- Perth Metro (Station number: 9225) - Western Australia
- Sydney (Observatory Hill) (Station number: 66214) - New South Wales
- Hobart (Ellerslie Road) (Station number: 94029) - Tasmania

Note that some of these stations are slightly different to those for rainfall data.

- **Data Quality:** Air temperature is measured using various instruments in modern automatic weather stations, which generally use electronic sensors [25]. The daily maximum temperature is added to the dataset once each day at 9 am local time, representing the highest temperature reached in the 24 hours preceding the time of entry into the dataset. As with the rainfall data, this data is retrieved from the Bureau's website in a pre-aggregated monthly form [26].

4.3.3 Real-time Weather Data

The real-time weather data has been collected from the Bureau of Meteorology by the use of API (see Section 5.1.2 for more details). We collected data from 95 weather stations around Victoria. Once every 15 minutes, the latest weather record is fetched from each of these stations, and those records are added to the growing dataset.

4.4 Sentiment Data

4.4.1 Historic Twitter Data

We aim to compare weather conditions with public sentiment to explore any potential correlations between the two. To achieve this, our primary source of historic sentiment data is Twitter. This dataset consisted of tweets from Australia, and included a pre-generated sentiment value.

Given the large size of the Twitter dataset (≈100GB), it was processed on the University of Melbourne's High-Performance Computing (HPC) system, Spartan [27]. Using Spartan, each tweet's pre-generated sentiment scores where extracted, and all tweets where aggregated by day. The results were the total tweet counts and total sentiment per day. These entries were then further aggregated by month by ElasticSearch.

4.4.2 Real-time Social Media Data

We obtained real-time social media data from Mastodon, a social networking platform, and utilized the Australian server to align with our area of study [28]. The data extraction process is detailed in Section 5.1.2. For the analysis of these toots, we employed a pre-trained natural language processing model from "vaderSentiment" [29], which generates the sentiment value of each Mastodon message.

5 System Architecture

We have chosen to develop an event-driven architecture for our system (Figure 1). Fission serverless functions and ElasticSearch data storage/search are the key components of the system which are containerised inside a Kubernetes cluster. There are two main categories of Fission functions in this system: the data insertion and data retrieval functions. Moreover, there are two subtypes of data insertion functions: static inserters and harvesters. Static inserters simply insert a given set of data, while the harvester functions pull the information from BoM and Mastodon APIs and persist the data inside ElasticSearch database. The Fission data retrieval functions are designed as RESTful API endpoints that retrieve data from ElasticSearch and are triggered when users refresh the data in the analytics dashboards.

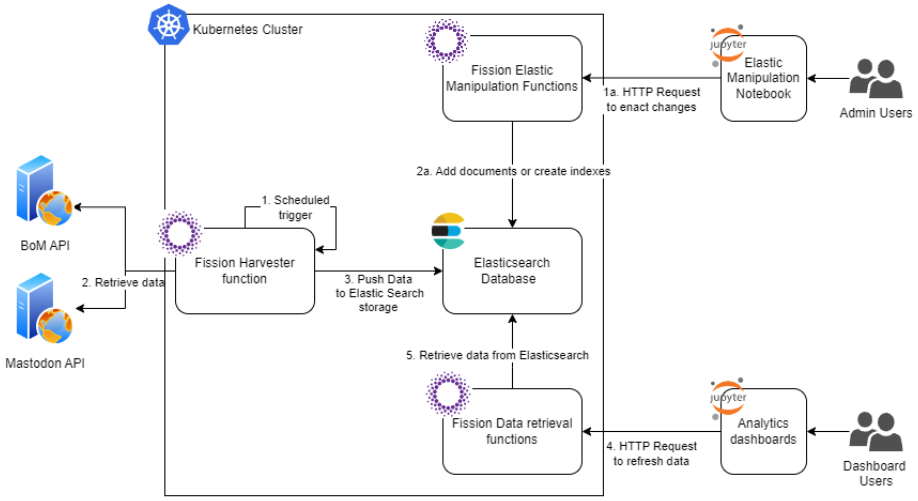


Figure 1: High-Level System Architecture

The general process flow in the system is as follows:

1. Fission harvester functions are activated on an automatic scheduled trigger.
2. The harvester functions retrieve the data from BoM and Mastodon APIs.
3. The data is persisted as documents inside the ElasticSearch database.
4. When users want to refresh the data in the dashboard inside Jupyter notebook, a GET HTTP request will be sent to the Fission data retrieval functions. Request will contain the details of the index in the ElasticSearch that needs to be queried.
5. Relevant data is retrieved from ElasticSearch and returned to the analytics dashboard as a JSON payload inside HTTP response.

Additionally, there is a process reserved for admin purposes that allow the following:

1. Admin user uses the Elastic Manipulation Notebook to request the creation of an index or insertion of documents into an index. This makes a POST HTTP request, including the to-be-inserted documents in the request body if required.
2. The Fission elastic manipulation functions action the request, creating an index if it exists, or inserting the provided documents.

5.1 Data Ingestion

5.1.1 Static File Ingestion

Static files are uploaded to Elastic through a Fission function with the endpoint "elastic/index/documents" where "index" is replaced with the index name in which to insert. Duplicate data is prevented by only allowing static file ingestion if an index is empty. For easy management, the elastic management notebook was created to hit these endpoints. The following are the steps that take place in this process (Figure 2):

1. First, the notebook reads the static file.
2. The notebook makes a POST request, including the file data in the body.
3. The Fission router passes this request to the data insertion function.
4. The function bulk inserts the data into the index.

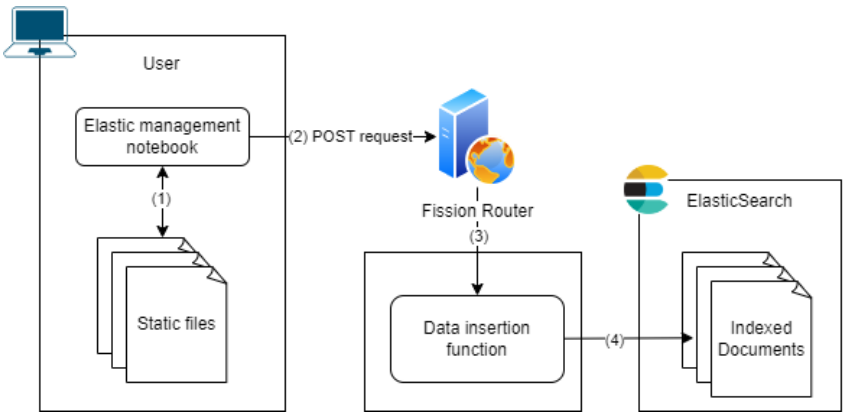


Figure 2: Static Data Insertion

5.1.2 Data Harvesting

BoM Harvester:
Fission function "backend/harvesters/BOM/addobservations.py" is scheduled to run every 15 minutes to retrieve updated weather information from BoM API. It is important to note that while this function is run every 15 minutes, many (if not all) of the stations only write new records every half hour. To safeguard against duplicates, the station name and observation recording time are used as a joint id. The function completes the following steps (Figure 3):

1. First the function queries the "stations" index in ElasticSearch to get the list of station names and corresponding URLs that need to be hit in order to get the weather information. The "stations" index contains the same information as the "data/bom_stations.csv" file.

2. A set of threads is then created to parallelize querying of the BoM URL endpoints.

3. Each thread is allocated a specific URL that it retrieves data from using HTTP GET request.

4. Once the thread receives the data from the endpoint it persists that weather observation information in "bom_observations" index in ElasticSearch. Each record is saved using its location and observation datetime as a joint id to prevent duplicate entries.

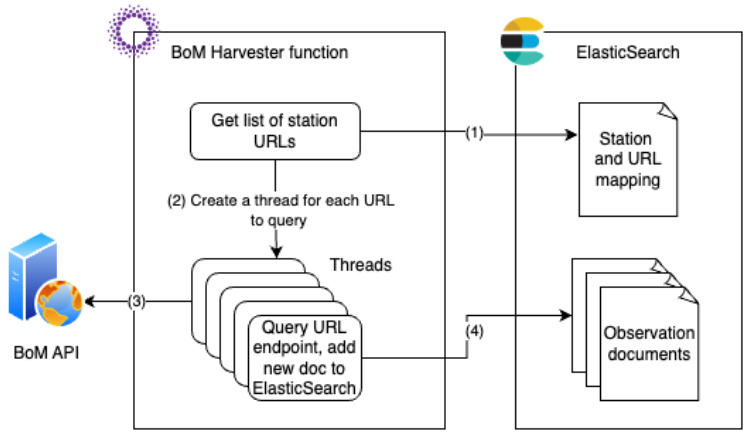


Figure 3: BoM Harvester Fission Function

Mastodon Harvester:
Fission function "backend/harvesters/Mastodon/mharvester.py" is scheduled to run every 5 minutes to retrieve latest posts from the Mastodon server. Duplicates are avoided by numerous means, the strongest of which is the usage of the unique Mastodon message id as the elastic id as well. This prevents duplicated data. The function completes the following steps (Figure 4):

1. First the function queries "mastodon_observations" index in ElasticSearch to get the ID of the last inserted record.

2. Last record ID is used when querying the Mastodon server API, as it allows us to retrieve a set amount of records (up to 2000) which were made after the post with last record ID has been published.

3. Function gets the data from the Mastodon API.

4. For all the retrieved posts, a check is applied that verifies that the post was created within last 5 minutes and was made after the last ingested post. This ensures that even if only a few posts have been made since the last time we have queried the API, the function will only upload fresh and not duplicated records into the ElasticSearch index. This is an additional safeguard on top of ElasticSearch's _id column. If the check fails this means that there is no new fresh data and function stops looping through returned records and moves on to the next step.

5. The new records are enriched by adding a sentiment analysis score to them using "vaderSentiment" Python module.

6. Enriched records are then stored in "mastodon_observations" index in ElasticSearch.

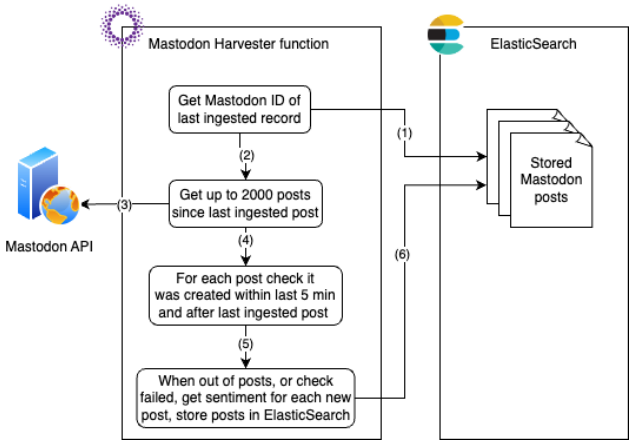


Figure 4: Mastodon Harvester Fission Function

As an aside, the Mastodon harvester is setup in a way which allows it to backfill missing data by retrieving all Mastodon messages within the last five days. However, this is a backup in the event the regular harvester fails, and it is a *very* long running job. As such, we will not go into detail about it here. If the harvester works as described above, the backfill functionality should never need to be used.

5.2 Data Management

ElasticSearch spreads data across shards. More shards means queries are more parallelized, but incurs overhead costs. Shards also enable partial data usage in the event some shards' data is lost. For this project, most data sources are too small to consider using multiple shards, and the data is easily re-inserted via the elastic management notebook (Section 5.1.1) so partial data is not very valuable.

The exceptions to this are the the hourly average air quality data, which is given three shards, and the regularly updating indexes. The BoM and Mastodon indexes are given three shards as the data within is expected to grow very large in the hypothetical long-term existence of the project cluster.

Replicas duplicate data, protecting against data loss in the event of a lost ElasticSearch node. In the project's cluster, there are two ElasticSearch nodes. Hence, each data index is created with one extra replica. In the event that data storage needs to grow, more machines can be requisitioned into the Elastic cluster, i.e., horizontally scaled to store more data and maintain access speeds.

5.3 Data Access

Data access is exposed via a RESTful API, where the endpoints are in the form "/data/resource category/resource". Each resource category maps to a different Fission function and each function returns different data depending on which resource was requested. These resources are as processed as possible, completing all computationally costly actions such as searching, grouping, and merging, before returning the data as a response. This leaves only cosmetic code for the frontend notebooks to achieve. When a resource is requested, the following takes place (Figure 5):

1. First, the notebook makes a get request to the cloud backend.
2. The Fission router invokes the relevant function based on the requested resource category.
3. Function gets the data from ElasticSearch. This can include multiple requests to different indexes.
4. The data is returned in the form of columns and rows.
5. The notebook converts this to a dataframe to feed into the analytics displays. The notebooks only contain code that makes cosmetic changes, such as the renaming of columns for graphical output purposes and the actual visualizations themselves.

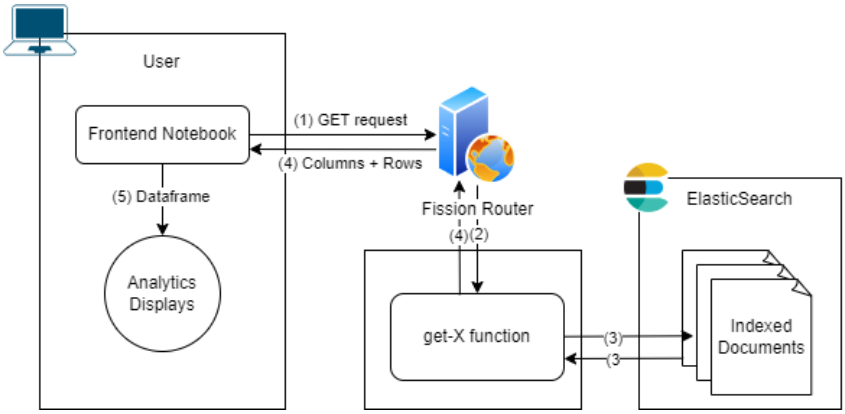


Figure 5: Fetching Data

5.4 Front-End

The frontend of our data analytics project uses Jupyter Notebooks to provide interactive data analysis, allowing users to execute code cells and view results dynamically.

For most of the analytics, this will simply result in the same graphs being produced. However, for the graph that uses the regularly growing weather and Mastodon sentiment datasets, the output graph will be extended once every hour. By using an interactive Jupyter notebook, accessing this updated information is as simple as a click of a button.

We organized our analysis into two notebooks for our two scenarios. Separating scenarios into different notebooks is a recommended practice to maintain clarity and organization within the analysis process [30]. Moreover, leveraging the show/collapse cell feature of Jupyter Notebooks can enable users to focus on specific sections of our analyses without information overload.

5.5 Error Handling

In an event driven architecture (EDA) system such as ours it is crucial to have robust error handling procedures to ensure reliability of the operation of the system.

Idempotent Operations:
One of the key ways to guarantee robustness of the EDA system is to ensure that all operations are idempotent. This allows the same operation to be repeated multiple times without changing the outcome. We have applied this principle when developing all the Fission functions that complete write operations. For example, when harvesting data from Mastodon we perform freshness and

sequence checks to ensure that we are not uploading duplicate records into the ElasticSearch index. Even if the function is run multiple times at small intervals, it will still not result in duplicates in the system.

Similarly, we have ensured impotency when creating indexes and uploading static files to Elastic-Search. Various checks are in place in "backend/index_creation" and "backend/ingestion" scripts that guarantee that same index cannot be created twice and no duplicate data can be inserted into it.

Exception Handling:
We have implemented timeout and retry mechanisms to enable exception handling. For all the HTTP requests that are made between different components of the system or to external systems, we define 'try/except' statements to ensure that system gracefully handles exceptions.

Crash Resistance:
The cluster has resistance against crashes in regards to nodes being broken. Kubernetes already self-heals by regenerating nodes that have gone down, using the templates the cluster was created with. The Fission aspect of our project goes untouched in a machine fault case, since the functions are stateless and do not contain or store any state. ElasticSearch is protected against faults because all our indexes have a replica (see Section 5.2), so both nodes where the two copies of a document exist must simultaneously be faulty for data to be lost. See Section 7.1.2 for detailed fault tolerance mechanisms.

6 Data Analysis and Results

6.1 Air Quality vs Lung Disease

This scenario aims to investigate the relationship between air quality parameters and the prevalence of lung-related disease in Victoria. For this, we begin with exploratory data analysis (EDA), followed by spatial, temporal, and correlation analysis to get an overall understanding of any underlying trends that is present in the extracted dataset.

6.1.1 EDA of Air Quality Dataset

Summary Statistics:

Parameter	count	mean	std	min	25%	50%	75%	max
CO	43966	0.167876	0.128603	-0.589	0.095	0.144	0.2060	1.912
O3	66467	16.296012	9.242939	-0.971	9.937	16.684	22.4985	66.943
PM10	56597	16.437867	11.658281	-19.612	9.208	13.903	20.7580	416.870
PM2.5	81045	5.775832	5.362307	-42.107	2.466	4.900	8.0410	96.918
SO2	49353	0.434215	1.719208	-5.458	-0.070	0.280	0.6780	115.233

Table 1: Summary Statistics by Air-Quality Parameter

Based on the summary statistics, we can draw the following insights:

1. **Presence of Negative Values:** The summary statistics reveal the presence of negative measurement values, indicating measurement errors or anomalies. This is consistent with the findings of an 2018 audit which includes scrutiny on EPA's data quality [31].

2. **Extreme Values:** Some parameters exhibit extreme maximum values (e.g., PM10, PM2.5, SO2), indicating the presence of outliers or highly polluted conditions in certain instances. Alternatively, this could also be an artefact of EPA measurement errors.

As stated earlier in Section 3.1.1, most research implicate PM2.5 to be mainly responsible for lung-related disorders. Hence, for the subsequent analyses we will solely focus on this single air quality parameter.

Distribution of Data Across Monitoring Locations:
This visualizes the distribution of data across monitoring locations using a pie chart (Figure 6).

This analysis provides an overview of the distribution of observations among different monitoring stations, offering insights into the spatial coverage of the dataset.

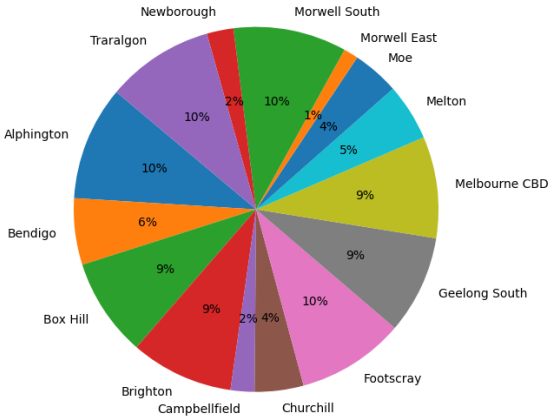


Figure 6: Distribution of PM2.5 Data Across Major Cities

There are several suburbs such as Morwell East, Newborough, and Campbellfield which have disproportionately low amounts of data samples compared to others. This is important to note as conclusions involving such suburbs could be flawed due to unbalanced data.

Comparative Analysis Across Locations:
Next, we perform a comparative analysis (Figure 7) to compare PM2.5 concentration across different monitoring locations. Through box plots grouped by location, we explore spatial variations in air quality and investigate potential differences in pollution levels among the monitoring stations.

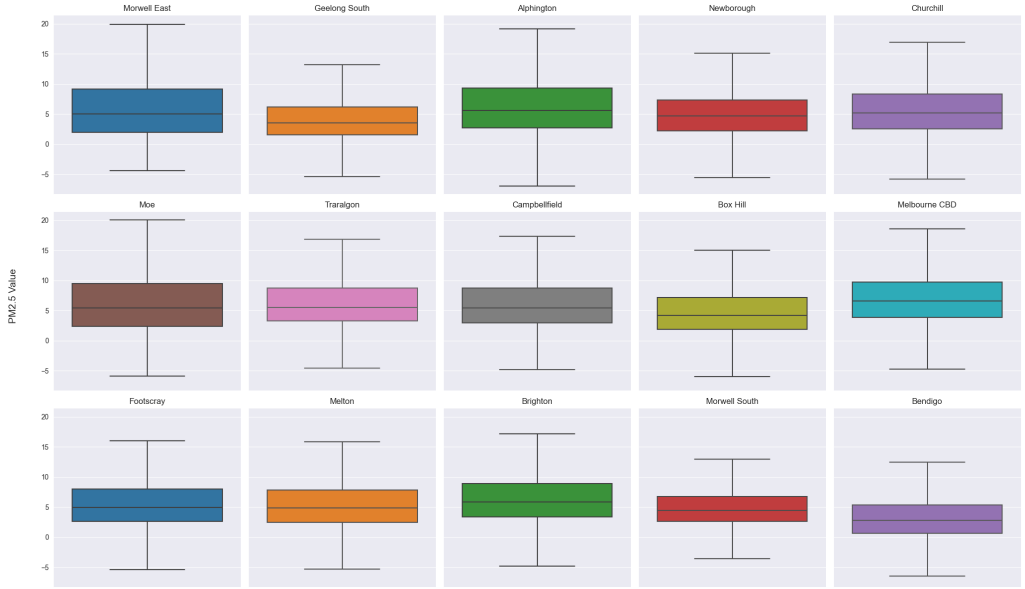


Figure 7: Comparative Analysis of PM2.5 Across Locations

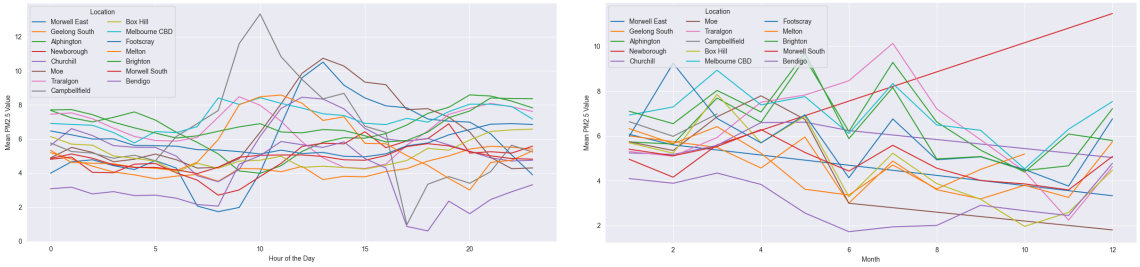
Although the boxplots might all look similar at first glance, scrutinising them reveals some hidden insights:

1. **Variability:** There is considerable variability in PM2.5 levels across different locations, as indicated by the range of values and standard deviations. For instance, standard deviations range from approximately 3.70 µg/m^3 to 6.36 µg/m^3, suggesting differing levels of air pollution or local factors affecting PM2.5 concentrations.

2. **Urban vs. Regional:** Urban areas like Melbourne CBD and Box Hill tend to have higher mean PM2.5 levels compared to regional areas like Bendigo and Traralgon. For example,

Melbourne CBD exhibits a mean PM2.5 level of approximately 7.27 µg/m^3, whereas Bendigo has a mean of about 3.32 µg/m^3. This discrepancy suggests differing sources of pollution, including traffic congestion, industrial activities, and population density, influencing urban air quality.

Temporal Trend Analysis:
Seasonal variations, diurnal patterns, and trends over time may reveal underlying factors influencing air quality at different locations. For that we have plotted the hourly (Figure 8a) and monthly (Figure 8b) trends of PM2.5 across all locations.



(a) Hourly Trend (b) Monthly Trend

Figure 8: Trends of PM2.5 Across All Locations

As expected, the hourly trend of PM2.5 gives some unique perspectives about this air-quality parameter:

1. **Impact of Human Activity:** Diurnal spikes in PM2.5 levels are observed during the day, indicating the influence of peak human activity hours. This may be attributed to increased traffic congestion, industrial operations, and energy consumption (e.g., from heating, cooking, and air conditioning), which release particulate matter into the air.

2. **Regional and Local Variations:** Spatial and temporal variability in PM2.5 concentrations is observed, influenced by regional and local factors such as geographical location, urbanization, and proximity to emission sources. These factors contribute to the observed hourly trends in PM2.5 values across different locations. For example, some suburb's hourly PM2.5 values lack the peak during working hours, while some suburbs are heavily affected. A user more specialised in environmental science may be better equipped to deduce possible reasons for this observation.

Furthermore, the monthly patterns help in identifying long-term trends in air quality dynamics:

1. **Wood Burning and Heating:** Cooler temperatures during autumn and winter may lead to increased use of wood heaters and fireplaces for residential heating purposes, as well as electric energy demand resulting in more output required from coal fired electricity plants. Comparatively, the warmer months of October to December have less pollution output. Wood and coal combustion emits particulate matter and other pollutants, which can contribute to higher PM2.5 concentrations, particularly in urban and rural areas.

2. **Always Increasing Trend of Newborough:** Coal-fired power plants like Yallourn can be significant sources of particulate matter (including PM2.5) emissions, and there has been rising safety concerns surrounding this plant over the last few years [32].

Spatial Analysis:
Finally, we conduct a spatial analysis of the PM2.5 data to visually analyse the spread of PM2.5 particles across Victoria (Figure 9). This will help us pinpoint locations of interest where there might be a higher chance of lung-related disorders.

The analysis of PM2.5 levels reveals that Greater Melbourne has an average concentration of 6.24 µg/m^3, slightly higher than the surrounding areas of regional Victoria, which have an average concentration of 5.29 µg/m^3. Most notably, central Melbourne has the highest concentration at 7.3 µg/m^3. Another extreme is Bendigo (off the map) that has 3.3 µg/m^3.

Here are some insights into why this could be the case:

1. **Urbanization and Population Density:** Suburbs closer to Melbourne City tend to have higher population density and more industrial activities compared to other regions of Victoria. These factors contribute to higher levels of PM2.5 emissions.

2. **Geographical Factors:** Geography and weather patterns can also influence PM2.5 levels. Coastal areas might experience lower levels due to cleaner air coming in from the ocean. For example, Geelong has rather low PM2.5 values. The lowest PM2.5 value is in Bendigo, possibly due to its remoteness relative to the other locations.

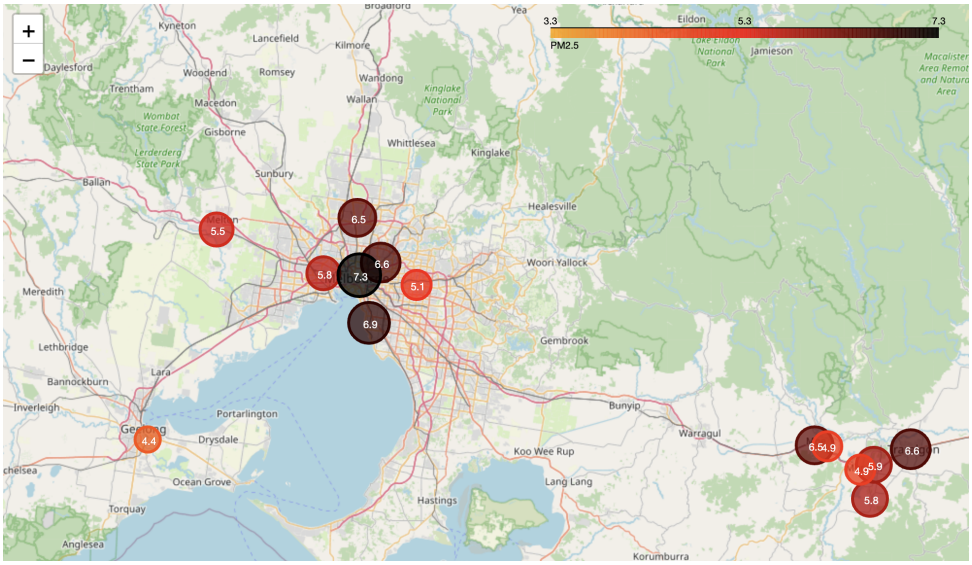


Figure 9: Comparative Analysis of PM2.5 Across Locations

6.1.2 EDA of Lung Disease Dataset

Summary Statistics:
Same as before, we first examined the summary statistics (Table 2) from various angles to understand the distribution and central tendencies of the data on a region-by-region basis.

	Total Asthma	Total COPD	Lung Cancer Total Mortality
Count	15	15	15
Mean	137608.06	29331.93	2690.86
Standard Deviation	127839.51	25478.82	2505.53
Minimum	4429	942	137
25% Percentile	34008	7573.5	585
50% Percentile	121331	25546	2506
75% Percentile	218554	52892	4017
Maximum	386347	72543	7272

Table 2: Summary Statistics of Asthma, COPD, and Lung Cancer Mortality by Region

The key information from this table relevant to the scenario is the high variation. All three metrics have standard deviations almost equal to their averages. This indicates that there are some factors that are different per region which affect lung disease. This scenario will attempt to show that air pollution is one of them.

Distribution of Data Across Locations:
Next, we plot pie charts (Figure 10) to see the distribution of asthma and COPD prevalence and lung cancer mortality across major regions to understand the spatial spread of the problems.

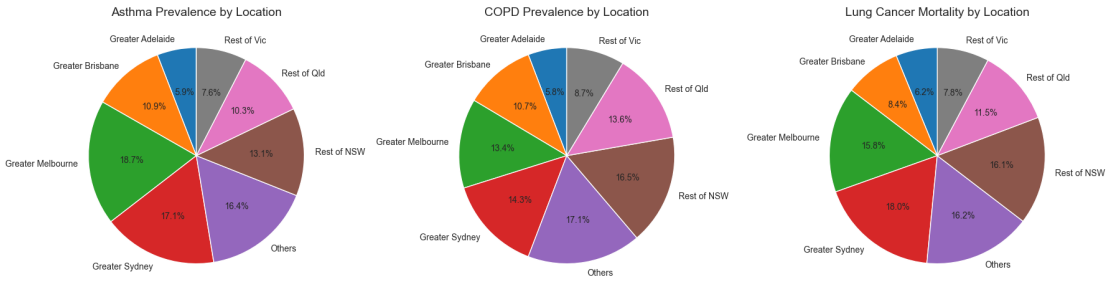


Figure 10: Distribution of Lung-Related Disorders Across Major Regions

From the pie-charts we can understand the following:

1. **Higher Pollution Levels in Urban Centers:** Major cities like Greater Sydney (17.1% asthma, 14.3% COPD) and Greater Melbourne (18.7% asthma, 13.4% COPD) experience elevated levels of air pollution due to industrial activities and vehicular emissions. Poor air quality can exacerbate respiratory conditions, leading to higher prevalence rates compared to regional areas with cleaner air.

2. **Impact of Geographic Features:** Coastal cities such as Greater Brisbane (10.9% asthma, 10.7% COPD) may benefit from milder climates and lower levels of pollution, contributing to relatively lower prevalence rates compared to inland cities.

6.1.3 Correlation Analysis of PM2.5 vs Lung Disorder

Now that all exploratory data analysis is done, we will focus on the correlation between air quality and lung-related disorders among the population in Victoria. From the correlation analysis (Table 3) conducted between PM2.5 levels and lung diseases, and the comparison of lung disease categories across region (Figure 11), the following conclusions can be drawn:

1. **Positive Correlation with Lung Cancer, All Cancer, and COPD Mortality:** Greater Melbourne has higher PM2.5 levels and comes with higher mortality rates relating to lung disease for all disease types.

2. **Higher Risk for Foreigner Populations:** Additionally, the data indicates that foreigner populations are particularly susceptible. Asthma rates in a higher PM2.5 region (Greater Melbourne) is disproportionately higher for foreigners than for foreigners in lower PM2.5 regions.

3. **Regional Livability:** Given the established link between elevated PM2.5 levels and various lung diseases, it is evident that residing in areas with lower PM2.5 levels may reduce the risk of these health issues. Therefore, from a public health perspective, residing in regions outside of Greater Melbourne, with comparatively better air quality, may be beneficial for reducing the risk of lung diseases associated with elevated PM2.5 levels.

Region	PM2.5	Cancer Mortality		Total		Australian		Foreign	
		All	Lung	COPD	Asthma	Asthma	COPD	Asthma	COPD
GMEL	6.24	6377	35887	58925	386347	292521	37523	91496	20287
RVIC	5.29	3138	17475	38195	156014	140299	31287	13405	5734

Table 3: Correlation Analysis of PM2.5 and Lung-Related Disorders

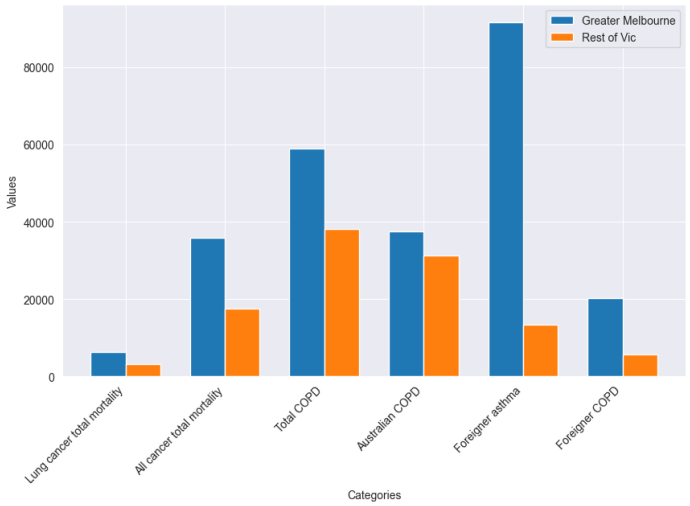


Figure 11: Comparison of Lung Disease Categories by Region

6.1.4 Additional Analyses

As an extension of this scenario, and as an exercise in easily extending the scope of analysis possible via a cloud based solution, we created two additional mini-scenarios. These scenarios link lung disease to gender and income.

Lung Cancer Prevalence by Gender and Location:
The grouped bar chart (Figure 12) compares the prevalence rates of lung cancer between males and females in different geographical areas, providing insights into potential gender disparities in lung cancer incidence.

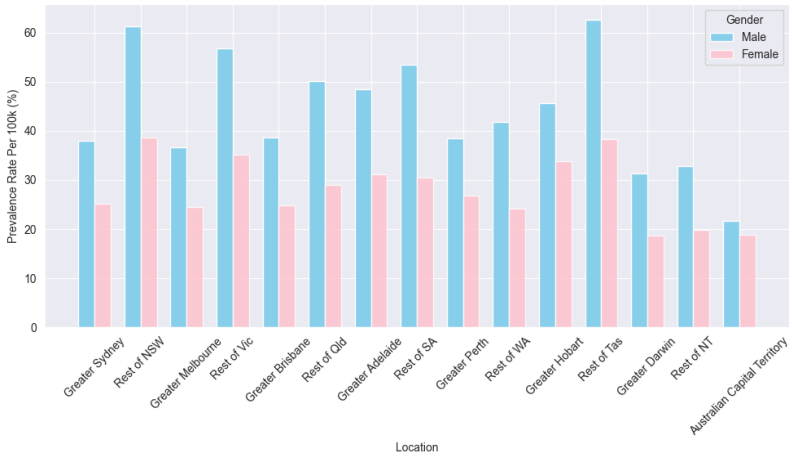


Figure 12: Lung Cancer Prevalence by Gender and Location

From the bar chart, we can notice that across most regions, males consistently demonstrate higher lung cancer prevalence rates than females. Notable discrepancies are observed in locations like 'Rest of Vic', where the male prevalence rate reaches 56.77 per 100,000 population compared to 35.15 per 100,000 population for females.

Prevalence of Asthma and COPD by Income Level:
Using boxplots (Figure 13) to visualize the prevalence of asthma and COPD across different weekly income levels, we can show the distribution of prevalence rates within each income category.

From these graphs, we can see that there is a clear relationship between income and lung disease. For COPD, it is especially obvious that low-medium income earners are at a much higher risk of COPD. Higher income earners are likely able to afford healthcare to alleviate COPD, while lower income earners may simply go undiagnosed. Similar conclusions may be drawn about asthma,

albeit that asthma is less treatable and more common than COPD. Hence, it does not drop away as much with income as COPD does, and retains a higher variation per income group.

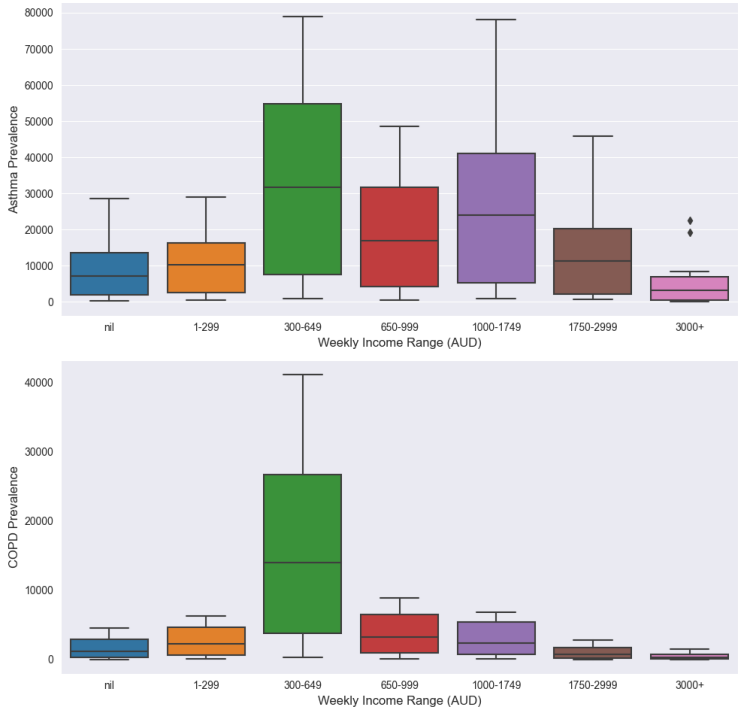


Figure 13: Prevalence of Asthma and COPD by Income Level

6.2 Weather Patterns and Mental Health

This scenario aims to find a relationship between weather and mental health. The analysis is done on both historical weather data and Twitter sentiments, and real-time weather data and Mastodon sentiments.

6.2.1 Historical Analysis

We begin by analyzing Twitter sentiments grouped monthly for the years 2021 and 2022. For this we plotted a line graph (Figure 14) to visualise the trend of monthly weather patterns and people's sentiments extracted from their tweets. Sentiment have been scaled (divided by 100) so that the graph has a reasonable y-scale.

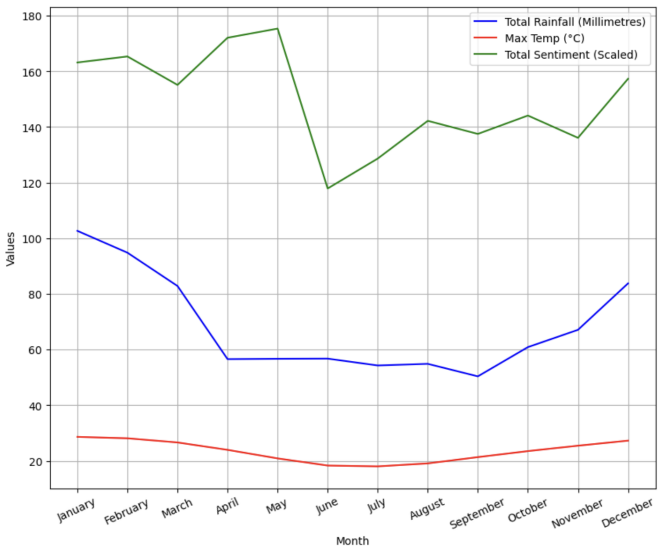


Figure 14: Monthly Climate Data and Sentiment Scores

From the graph, we can see that the sentiment values, monthly rainfall, and max temperature all appear to loosely follow the same trajectory. These values are all higher during the summer months, dropping lower as winter rolls around. Notably, there is a sharp drop in sentiment between the months May and June. A possible speculation is that this is due to the incoming winter bringing down the mood on social media. Alternatively, since this graph was only made using one year's data, a large negative event such as a COVID variant being discovered or a political event could also cause a sentiment drop. Any strongly negative and widely discussed topic could cause such a sentiment drop in social media.

6.2.2 Correlation Analysis

To complete analysis of historical data, we now conduct a correlation analysis between the weather data and sentiments. The results are as follows:

1. A **strong positive** correlation (0.87) exists between total rainfall and mean maximum temperature, suggesting that higher temperatures are often accompanied by increased rainfall.

2. A **moderate positive** correlation (0.44) between total sentiment and total rainfall indicates that more rainfall is somewhat associated with more positive sentiments.

3. A **positive** correlation (0.58) between total sentiment and mean maximum temperature implies that higher temperatures are linked to more positive sentiments.

Factor	Correlation Coefficient
Total Rainfall and Avg. Max Temp	0.87
Total Sentiment and Total Rainfall	0.44
Total Sentiment and Avg. Max Temp	0.58

Table 4: Correlation Analysis between Weather Data and Sentiments

These insights reveal a noticeable positive correlation between warmer temperatures and higher sentiment scores. Warmer weather likely encourages outdoor activities, social interactions, and general well-being, leading to more positive tweets. This is a sign of better mental health. Interestingly, correlation exists between higher rainfall and sentiment. However, it is more likely that of the three factors being considered, warm weather is the causal factor. Warm weather brings better moods and more rain, as opposed to there being a direct connection between rainfall and positive moods.

6.2.3 Real-Time Analysis

To extend this scenario, and to utilise the strengths of the cloud system we have, we conducted an analysis between the real-time weather and sentiment data (Figure 15).

This graph demonstrates some interesting behaviour. First of all, there is a sharp spike in sentiment between the 18th and the 19th of May. Perhaps there was some viral discussion at the time. Secondly, and most interestingly, there appears to be a correlation between sentiment and rainfall. Previously, in Section 6.2.1, we concluded that warm weather is the cause of rising sentiment. This real-time graph contradicts that, and poses a new hypothesis. Perhaps rain simply forces people indoors, where they post more. If the general mood is typically positive, then total sentiment will rise.

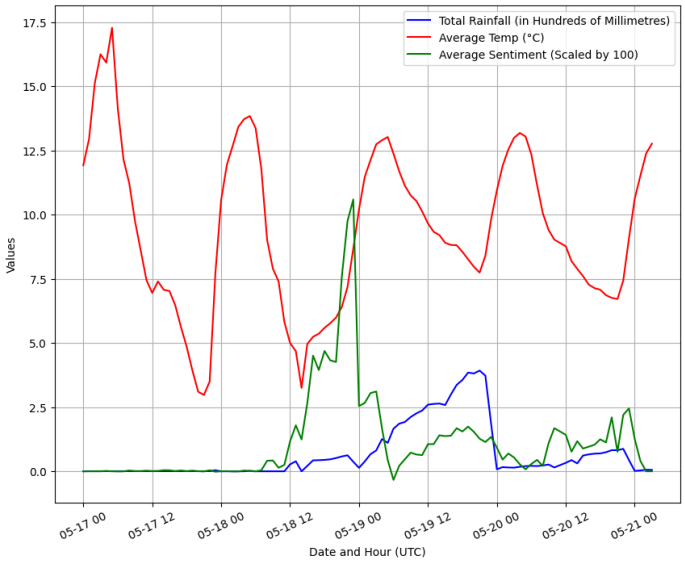


Figure 15: Real Time Climate Data and Mastodon Sentiment Scores

To test this hypothesis, we can simply create a new graph (Figure 16) that records the message counts. This involves adding a new function to the backend, and exposing it through the RESTful API.

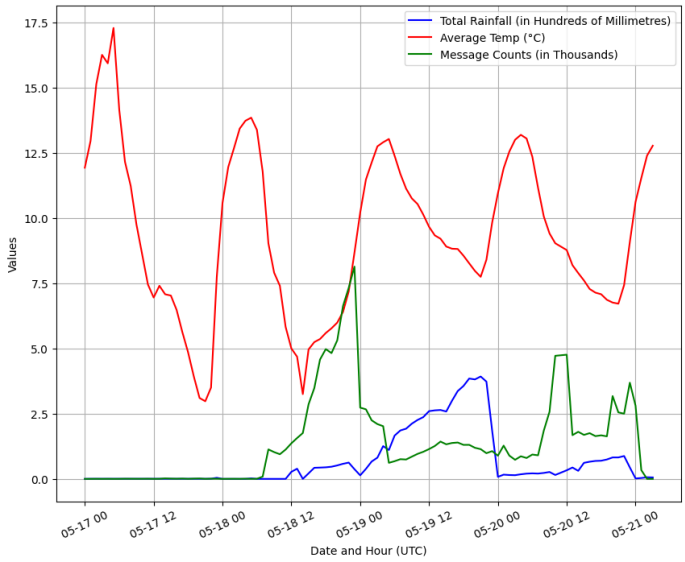


Figure 16: Real Time Climate Data and Mastodon Message Counts

With this graph, we see that the hypothesis falls apart, as the message count appears to be largely unaffected by rainfall.

6.2.4 Analysis Reflections

From the historic trends, we observed that there was correlation between all three of rainfall, temperature, and social media sentiment. At first, it felt logical that warm weather was the causal factor for both rainfall and media sentiment. However, with closer inspection of the incrementally growing dataset, it could be observed that rainfall during every hour affected sentiment. We then tested that hypothesis by finding the trend of total message counts, disproving that hypothesis.

The only final conclusion to be made is that the relationship between weather and social media sentiment is very complicated, and has many contributing factors beyond those recorded in this project. However, the important take away is the ability of a RESTful API to expand when backed by a well designed system. As demonstrated above in the real-time analysis (Section 6.2.3),

extensions to this project's capabilities based on new analytics demands from users is quick and easy. This is because our system is well designed with a modular backend that can be easily extended to accommodate a growing RESTful API scope.

7 Reflection

7.1 Technology

7.1.1 Tech Stack: Pros and Cons

During this project, we were thankful of certain aspects of the technologies used, and struggled with some other aspects. In this section we cover the notable pros and cons of these technologies.

MRC:
The Melbourne Research Cloud was a convenient resource for setting up a secure cloud infrastructure. Being based off Openstack, the MRC has the same benefits of convenience regarding cluster resource procurement and orchestration. Setting up the project using a prescribed recipe was very simple because of this. MRC also comes with security benefits. With just a few console lines, the project's cluster was protected behind a security node which only allows connections through SSH. This security was also the main con of this system, since it made automatic CI/CD at a level above unit testing non-trivial.

Kubernetes:
Interaction-wise, we as developers interacted little with Kubernetes asides from reading diagnostic values. This is a valuable aspect of Kubernetes, and it had abstracted away many complicated container orchestrations that we would otherwise had to manually do. This includes:

1. **Container Regeneration:** It can automatically manage the deployment of applications, allowing for smooth updates and the ability to revert to a previous version if something goes wrong. For example, if a Fission function pod or ElasticSearch container fails, Kubernetes is able to self-heal and recreate those containers.

2. **Integration:** Kubernetes is a well known and common framework. Many technologies, such as those used in this project, are easily integrated with Kubernetes. At the same time, since Kubernetes is a means of orchestration, these technologies can be summoned and removed as needed.

3. **Horizontal Scaling:** It allows for automatic scaling of applications up and down based on demand, ensuring efficient use of resources. While our project does not currently utilise this, in the hypothetical long-run, some components of our project would benefit from horizontal scaling. This is possible because those components integrate with Kubernetes. For example, Fission with a New Deploy execution methodology would communicate with Kubernetes to summon and remove Fission execution pods as needed.

When we did interact with Kubernetes, it was to read the logs of certain pods. A particularly notable incident was the debugging of failed package builds. Being able to read the Fission build logs and that of the Python environment was critical in diagnosing the issue. The ease at which a few Kubernetes CLI commands can deliver logs messages from anywhere in the cluster was a massively contributing factor towards gaining sufficient development speed.

Fission:
Fission was outstandingly useful in simplifying the actions possible for users of our project. In theory, the use of Fission could have been ignored in favor of directly accessing the ElasticSearch component in our cloud cluster. However, this would over-complicate the understanding and connections (code-wise) required by a user/client to request the data they need. By using Fission, this data can instead by exposed via a RESTful API. Furthermore, a data management API was created to manage ElasticSearch indexes and documents (Figure 2), which sped up the development cycle of modifying and creating of indexes, as well as the insertion of data.

Other key aspects that made Fission a great option for our use case are:

1. **Kubernetes Native:** The Fission framework is designed to run on Kubernetes clusters

making it easy to setup Fission on our dedicated MRC instance.

2. **Event-Driven Architecture:** It allows us to develop data processing functionality that is triggered through events. Specifically, we utilised its ability to use HTTP requests as triggers to implement RESTful APIs, and timer based triggers to regularly ingest from external data sources like Mastodon.

3. **Horizontal Scaling:** As briefly mentioned in Kubernetes, Fission has the capability to automatically scale horizontally given an influx of requests. While we do not utilise this feature, the fact that this is readily available means that this project could be easily extended to service many clients at once.

ElasticSearch:
ElasticSearch was convenient for document storage, but its potential was not fully utilised in this project. ElasticSearch really shines with truly massive amounts of data to search through, where it can utilise multiple shards to search in a parallel fashion. In time, the regularly growing data will start to incur query delays as the data sets grow large enough, allowing sharding to provide benefits. For now though, this project benefits little from sharding.

Being a document based storage system, ElasticSearch does not provide support for cross-index joins. To avoid this issue, indexes should be designed to accommodate fully enriched documents. In hindsight, several of our indexes could have been conjoined together by employing enrichment during insertion. This would avoid the need to action joins upon the fetching of certain data resources. Right now, these joins are handled within the Fission functions where needed. For example, when fetching the averaged weather per month, the Fission function needs to join the data itself, since ElasticSearch is unable to do so. This is because all the weather data is split into multiple indexes, one per location.

One benefit we did gain from ElasticSearch was the ability to have duplicated data in replicas, meaning having each document copied in at least two machines. This allows the project's system to be more fault tolerant in the face of ElasticSearch node loss. All our indexes had one replica, meaning that even if an ElasticSearch node was lost, all the data remains available.

Another benefit of ElasticSearch is is that it comes with a powerful GUI called Kibana [33]. This was very useful for developing, diagnosing, and testing data management.

Jupyter Notebook:
Jupyter Notebook was particularly helpful for visualizing analytic scenarios. By utilising Jupyter with powerful data analytics graph packages such as Matplotlib, Seaborn and Folium, we were able to easily create graphics that provide easily understandable insights. Some of these insights are also interactable, such as the maps generated by Folium. Making these graphical outputs readily accessible would have been more difficult without the use of Jupyter Notebooks.

Additionally, since Jupyter Notebooks comes with a well-designed UI, it enables the rapid development and running of Python code. This expedited data exploration, shortening development time.

7.1.2 Performance and Functionality

Initial Data Load:
For most of the files, the initial insertion into ElasticSearch was fast and simple due to their small sizes. However, with the large hourly air quality file (765202 rows), the ElasticSearch client frequently timed out. This necessitated a timeout increase for the ElasticSearch client from the default 10 seconds to 30 seconds, as well as batched insertions.

Query Speed:
The largest index that is queried is the hourly air quality index. Querying this index from the corresponding frontend notebook takes between 30 and 45 seconds. However, logging the invoked Fission function shows that it takes around 8 seconds to actually query the index. Given that our system currently uses the pool manager to constantly keep function pods warm, there should be minimal invocation delay. It remains unclear as to where this delay is coming from.

Scaling:
Currently, the project does not automatically scale. The Fission environments are created with a pool size of 3, meaning that three pods are ready to invoke Fission functions at any time. If all three are in use, a running pod needs to finish before the next request can be serviced, which for now is sufficient due to low user counts. A change can easily be made to service the Fission functions using the New Deployment executor type instead of Pool Manager. This will spin up new pods to service Fission functions as needed, albeit at the cost of a longer delay for cold starts.

ElasticSearch also does not automatically scale. To horizontally scale ElasticSearch, an administrator must increase the shard count for an index, which involves re-indexing the contained data. The recommended size to achieve per shard is around 200M documents, or between 10 to 50 GB. Our current data stored is nowhere near the recommended size for a single shard, let alone the excessive three shards that have been given to the indexes which are currently growing. Until these sizes are achieved, there is little reason to manipulate the shard counts.

Fault Tolerance:
We have put extensive effort to minimise manual configurations in our system and adhere to infrastructure as code principles as much as possible to ensure that in event of a critical failure the system can be spun up again. We have defined "backend/Fission_startup.sh" and "backend/Fission_wipe.sh" scripts that will create and tear down all of the functions provisioned on the Fission client respectively. Although this solution is sufficient in allowing us to update and refresh our Fission setup with a one line command, the usage of spec files would have been better. For example, spec files will delete unused Fission objects whereas our current system requires a developer to remember to add it to the Fission_wipe file.

Similarly to Fission, ElasticSearch could be another point of failure for our system. In the event of a failure regarding ElasticSearch, our data is safely backed up due to all indexes having a replica. Kubernetes then works with ElasticSearch to self-heal the fault, all without interference from a developer.

Finally, we acknowledge that our system heavily relies on external systems like MRC and inherently depends on the SLAs that it adheres to. Hence, any faults there will also impact our system. For example, when our Bastion node spontaneously went down, the project cluster became entirely inaccessible.

7.2 Software Design

We utilised an informal pseudo-extreme programming methodology to develop this project. All formal meetings/documentation was sacrificed for a faster development cycle. We deemed this necessary to accomplish a minimally viable product that achieves this project's intention within the allotted time. We discovered that this approach, and indeed any Agile approach, would result in difficulties given inexperience and incomplete consideration about some of the technologies used.

Tech Debt:
Even within the small time frame of this project, tech debt was rapidly accrued resulting in considerable time cost to work around inefficient software design and flaws. These were largely due to insufficient consideration to the strengths and weaknesses of the frameworks used, especially ElasticSearch.

Inexperience with ElasticSearch and the underestimation of the importance of enriching documents at index-time resulted in poorly designed indexes. This is best observed with the weather indexes where there are separate indexes per weather data origin when they could have been placed into the same index together. The documents could simply have been enriched with the origin label during insertion time. Doing so would have avoided the need to join this data within a Fission function, as ElasticSearch does not support cross-index joins.

Another consequence of inexperience with ElasticSearch relates to the accidental usage of dynamic mappings in indexes corroborated with the incorrect usage of indexing APIs. This led to significant development time wasted trying to figure out why data was appearing in our indexes in the wrong format. This was due to dynamic mappings being enabled, and incorrectly formatted data being sent through via the misuse of indexing APIs.

Asides from ElasticSearch, inexperience with Fission resulted in similar mistakes. For example, our inexperience led to the underestimation of the value of spec files. If we had made spec files to manage our Fission setup, we could have avoided confusion around what is currently in our project cluster or not.

Given this experience, the team now leaves this project with a much better understanding of these possible mistakes, and can avoid these issues in future projects.

Review Process:
Despite inexperience costing the team significant development time, it could have been worse without our peer review process. As part of our development process, each new feature was vetted by another team member prior to merging to the main branch in GitHub. This prevented several mistakes from further delaying the project.

We were also aware that each team member had sensitive data such as the Kube config and ssh keys stored locally. These files were not only added to the .gitignore file, but GitGuardian [34] was also used to ensure sensitive files were not included in pull requests.

7.3 Software Testing

To streamline software development and testing process we have created an automated testing pipeline (Figure 17). This pipeline, powered by GitHub Actions, goes through the following steps:

1. Pipeline is triggered when user makes a pull request from a feature branch into the main branch.

2. Once the pull request is made, a GitHub action job is executed that runs a set of unit tests in "test/Fission_tests.py" using pytest module.

3. Github Actions job triggers GitGuardian checks.

4. Another user peer reviews and approves the pull request.

5. Once all checks are passed the pipeline is merged into the main branch.

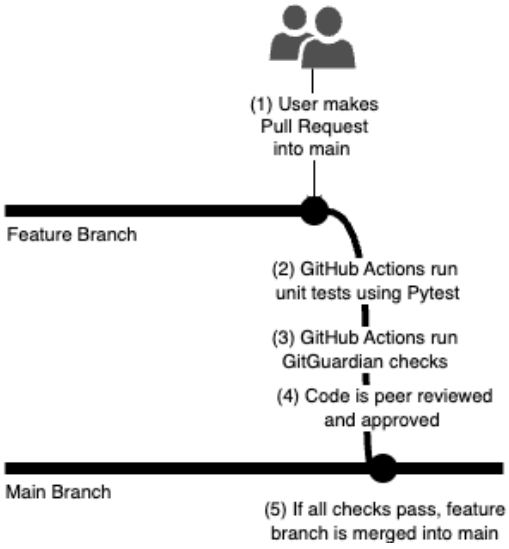


Figure 17: CI/CD Software Testing Process

The primary scope of our unit testing are the backend Fission functions and supporting Python methods that they use. We have used 'pytest-mock' to simulate responses from 'requests' and 'elasticsearch8' modules.

Additionally, we have relied on UAT testing for the front-end (Jupyter notebooks) component. We have reviewed that analytical dashboards correctly run and satisfy the functional requirements.

7.4 Team Coordination

Our team comprised of four members, each bringing unique skills and expertise to the project, contributing to the successful delivery of our analytics system.

7.4.1 Individual Contributions

1. **Ojaswi Dheer: Data Gathering and Front-End**
   Collected the static datasets from across SUDO, BoM and EPA, and preprocessed them to ensure data quality and relevance. Developed the frontend for "Weather Patterns and Mental Health", performing various statistical analysis on the different datasets and generating insights from them.

2. **Petr Andreev: Fission Management and Automated Testing**
   Implemented serverless Fission functions that efficiently processed incoming data and triggered necessary actions in real-time, ensuring scalability and responsiveness of the system. Designed and implemented a comprehensive suite of automated tests. Set up a CI/CD pipeline to run tests automatically upon code commits, ensuring immediate feedback on code quality and preventing the introduction of bugs.

3. **Rafsan Al Mamun: Cloud Stack Setup and Front-End**
   Configured the cloud environment based on the instructions from the tutors. Performed data cleaning and preprocessing before conducting analysis on the air quality of Victoria and its connection to lung-related disorders among the population. Also oversaw the overall progress of the project to ensure it adhered to the project goals and provided specifications.

4. **William Chen: ElasticSearch and Data Access**
   Designed the packaging methodology to update the Fission objects in MRC through a single command line interface command. Utilised that packaging methodology to develop Fission functions which enabled the rapid creation and re-creation of ElasticSearch indexes and data insertion through a basic API. Created the RESTful API that accesses the resources and modified the frontend notebooks to utilise that API. Expanded upon the real-time harvesters to also backfill old data. Regularly refactored and reworked backend code to fix and prevent system wide malfunctions that could not be detected by unit tests alone.

7.4.2 Team Dynamics

Strengths:

- **Collaboration:** Our team demonstrated strong collaborative efforts, with clear communication and regular updates through online messaging platforms. Weekly meetings ensured alignment and timely progress checks.

- **Complementary Skills:** Each member's specialized skills complemented the others, creating a well-rounded team capable of tackling various aspects of the project efficiently.

- **Problem-Solving:** When faced with technical challenges, such as optimizing ElasticSearch performance or integrating Fission functions, the team came together to brainstorm and implement effective solutions.

Challenges and Resolutions:

- **Coordination Issues:** Initially, there were some coordination challenges, particularly in aligning the data processing tasks with the frontend development timelines. These issues were mitigated by implementing more detailed project planning and scheduling regular sync-up meetings.

- **Integration Hurdles:** Integrating different system components developed by various team members posed challenges. We addressed these by conducting frequent testing and peer-review sessions.

- **Project Timeline and Conflicting Priorities:** One of the major challenges we faced was the limited time provided for the project, coupled with conflicting priorities from other

commitments. The tight deadline required us to work overtime and occasionally crunch to ensure timely delivery of the system. To manage them, we prioritized tasks based on their criticality and impact on the project. We also implemented flexible working hours that allowed team members to contribute at times that best suited their schedules.

- **Team Member Dropout:** Another significant challenge was the unexpected dropout of a team member. This sudden change meant that the remaining team members had to take on additional responsibilities to cover the work left incomplete. The team quickly adapted by redistributing the tasks among the remaining members. Each person took on additional responsibilities, with a focus on critical tasks to ensure project continuity. We re-prioritized our workload and increased our collaborative efforts to cover the gap.

7.4.3 In-Team Harmony

Our in-team coordination strategies evolved throughout the project. Initially, we faced some issues with overlapping tasks and miscommunication. To improve, we adopted the following measures:

- **Defined Roles and Responsibilities:** Clearly defining each member's role and responsibilities helped reduce overlaps and confusion.
- **Regular Updates and Feedback:** We scheduled weekly review meetings to provide updates, receive feedback, and adjust our approach as needed.
- **Collaborative Tools:** Utilizing version control systems (Git for code collaboration) ensured smooth workflow and version control, and allowed others to perform code reviews before merging them into the main system.

8 Conclusion

In conclusion, this project has demonstrated the power of cloud computing in conducting sophisticated data analytics to gain insights into the health and well-being of Australians. By leveraging a robust technology stack and analyzing diverse datasets, we have uncovered valuable information about lung-related disorders, air quality, weather impacts on mood, and social media sentiment. These insights have practical implications for policymakers, healthcare providers, and businesses seeking to improve public health outcomes and enhance the quality of life for Australians.

In doing so, we demonstrate how cloud computing can be utilised to store and gather data, and to provide data analytics through a simplified API. To showcase its strongest aspects, we also demonstrated how that process can be done in a regularly running fashion, using timers to incrementally grow a dataset and present it through an analytics graph that updates on the hour.

This project has also been a humbling experience, in which we have learnt many lessons through the mistakes we have made. Through reflection on the successes and regrets in the process of this project, we have gained skills and experiences that will aid us in future cloud related works.

References

[1] M. Chen, S. Mao, and Y. Liu, "Big data: related technologies, challenges and future prospects," *Springer*, vol. 46, no. 5, pp. 255–258, 2014.

[2] N. Hime, G. Marks, and C. Cowie, "Review of the health impacts of emissions from the bushfires in australia," *Respirology*, vol. 23, no. 1, pp. 13–22, 2018.

[3] Fission.io, "Fission." https://github.com/fission/fission. GitHub.

[4] Elastic, "Elasticsearch: The official distributed search & analytics engine." https://www.elastic.co/elasticsearch.

[5] Kubernetes.io, "Kubernetes: Production-grade container orchestration." https://kubernetes.io/. Kubernetes.

[6] "Melbourne research cloud." https://docs.cloud.unimelb.edu.au/.

[7] OpenStack, "Openstack - open source cloud computing infrastructure." https://www.openstack.org/.

[8] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (F. Loizides and B. Schmidt, eds.), pp. 87 – 90, IOS Press, 2016.

[9] Australian Institute of Health and Welfare, "Australian burden of disease study: Impact and causes of illness and death in australia 2018," 2021. Canberra: AIHW.

[10] I. C. Hanigan, R. A. Broome, T. B. Chaston, M. Cope, M. Dennekamp, J. S. Heyworth, K. Heathcote, J. A. Horsley, B. Jalaludin, E. Jegasothy, F. H. Johnston, L. D. Knibbs, G. Pereira, S. Vardoulakis, S. Vander Hoorn, and G. G. Morgan, "Avoidable mortality attributable to anthropogenic fine particulate matter (pm2.5) in australia," *International Journal of Environmental Research and Public Health*, vol. 18, no. 1, 2021.

[11] R. D. Brook, B. Franklin, W. Cascio, Y. Hong, G. Howard, M. Lipsett, R. Luepker, M. Mittleman, J. Samet, S. C. Smith, and et al., "Air pollution and cardiovascular disease: a statement for healthcare professionals from the expert panel on population and prevention science of the american heart association.," *Circulation*, vol. 109, p. 2655–2671, Jun 2004.

[12] R. T. Burnett, C. A. Pope, M. Ezzati, C. Olives, S. S. Lim, S. Mehta, H. H. Shin, G. Singh, B. Hubbell, M. Brauer, and et al., "An integrated risk function for estimating the global burden of disease attributable to ambient fine particulate matter exposure," *Environmental Health Perspectives*, vol. 122, p. 397–403, Apr 2014.

[13] N. Borchers Arriagada, A. J. Palmer, D. M. Bowman, G. G. Morgan, B. B. Jalaludin, and F. H. Johnston, "Unprecedented smoke-related health burden associated with the 2019–20 bushfires in eastern australia," *Medical Journal of Australia*, vol. 213, p. 282–283, Mar 2020.

[14] K. R. Smith and et al., "Global burden of disease of household air pollution for 2010," *Environmental Health Perspectives*, vol. 122, no. 12, pp. 131–137, 2016.

[15] G. Hoek and et al., "Long-term air pollution exposure and cardio-respiratory mortality: A review," *Environmental Health*, vol. 12, no. 1, p. 43, 2013.

[16] G. D. Thurston and et al., "Ambient particulate matter air pollution exposure and mortality in the nih-aarp diet and health cohort," *Environmental Health Perspectives*, vol. 125, no. 8, p. 087001, 2017.

[17] Environment Protection Authority Victoria, "Epa air watch all sites air quality hourly averages - yearly." https://discover.data.vic.gov.au/dataset/epa-air-watch-all-sites-air-quality-hourly-averages-yearly/, Jan 2013.

[18] R. Sinnott, "Spatial Urban Data Observatory." https://sudo.eresearch.unimelb.edu.au/.

[19] N. E. Rosenthal, D. A. Sack, J. C. Gillin, A. J. Lewy, F. K. Goodwin, Y. Davenport, P. S. Mueller, D. A. Newsome, and T. A. Wehr, "Seasonal affective disorder: a description of the syndrome and preliminary findings with light therapy," *Archives of general psychiatry*, vol. 41, no. 1, pp. 72–80, 1984.

[20] P. Baylis, N. Obradovich, Y. Kryvasheyeu, H. Chen, L. Coviello, E. Moro, M. Cebrian, and J. H. Fowler, "Weather impacts expressed sentiment," *PLOS ONE*, vol. 13, no. 4, p. e0195750, 2018.

[21] P. Baylis, N. Obradovich, Y. Kryvasheyeu, H. Chen, L. Coviello, E. Moro, M. Cebrian, and J. Fowler, "Weather impacts expressed sentiment," *PLOS ONE*, vol. 13, 08 2017.

[22] Australian Bureau of Statistics, "Type of long-term health condition (LTHP)." https://www.abs.gov.au/census/guide-census-data/census-dictionary/2021/variables-topic/health/type-long-term-health-condition-lthp, Oct 2021.

[23] Bureau of Meteorology, "Climate Data Online." http://www.bom.gov.au/climate/data/.

[24] Bureau of Meteorology, "Rainfall Data - Climate Data Online." http://www.bom.gov.au/climate/cdo/about/about-rain-data.shtml.

[25] Bureau of Meteorology, "Air Temperature Data - Climate Data Online." http://www.bom.gov.au/climate/cdo/about/about-airtemp-data.shtml.

[26] Bureau of Meteorology, "Rainfall Data - Climate Data Online." http://www.bom.gov.au/climate/data/index.shtml?bookmark=200.

[27] The University of Melbourne's Research Computing Services and the Petascale Campus Initiative, "Spartan." https://dashboard.hpc.unimelb.edu.au/.

[28] "Mastodon explore: Au." https://mastodon.au/explore.

[29] C. J. Hutto and E. E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*, 2014.

[30] V. Stodden and S. Miguez, "Best practices for computational science: Software infrastructure and environments for reproducible and extensible research," *SSRN Electronic Journal*, Sep 2013.

[31] "Improving victoria's air quality, victorian auditor-general's office, march 2018." https://www.audit.vic.gov.au/sites/default/files/2018-03/20180308-Improving-Air-Quality.pdf.

[32] Environment Victoria, "Yallourn, Australia's dirtiest power." https://environmentvictoria.org.au/our-campaigns/safe-climate/yallourn-australias-dirtiest-power/, Jun 2021.

[33] Elastic, "Kibana." https://www.elastic.co/kibana.

[34] "GitGuardian: Git security scanning & secrets detection." https://www.gitguardian.com/.