SOFTWARE SYSTEM FOR LONG-TERM HEALTH ANALYSIS - BIG DATA ANALYTICS ON THE CLOUD

FORMAL REPORT OF SOFTWARE SYSTEM DEVELOPMENT FOR COMP90024 BY GROUP 32

Yuchen Luo (1153247)
UNIVERSITY OF MELBOURNE
Victoria, VIC3000
stluo@student.unimelb.edu.au

Yiyang Huang (1084743)
UNIVERSITY OF MELBOURNE
Victoria, VIC3000
yiyahuang@student.unimelb.edu.au

Jiaqi Fan (1266359)
UNIVERSITY OF MELBOURNE
Victoria, VIC3000
jffan2@student.unimelb.edu.au

Yingying Zhong (1158586)
UNIVERSITY OF MELBOURNE
Victoria, VIC3000
yizhong1@student.unimelb.edu.au

Mingyao Ke (1240745)
UNIVERSITY OF MELBOURNE
Victoria, VIC3000
mingyaok@student.unimelb.edu.au

May, 2024

1 Introduction

In the modern era, data-driven decision-making has become a cornerstone across various sectors, particularly in healthcare. Our project harnesses the power of big data analytics to deliver insightful visualizations of health trends across different Statistical Area Level 2 (SA2) regions. By integrating a wide array of datasets, including weather conditions, income levels, and other socio-economic factors, our system provides a comprehensive and nuanced overview of the health status of residents in each SA2 region. This enriched perspective, together with our pre-identified scenarios, enables stakeholders to identify emerging patterns, accurately predict health outcomes, and make well-informed decisions aimed at improving public health and enhancing the overall quality of life for the community.

The backbone of our system is built on a robust cloud infrastructure, leveraging state-of-the-art technologies such as Fission[1], Elasticsearch[2], and Kubernetes[3]. These advanced

tools collectively enable the scalable, efficient, and reliable processing and analysis of large volumes of data, ensuring our system can meet the demanding requirements of modern big data applications. By integrating these technologies, we provide a powerful platform capable of handling dynamic workloads, offering real-time insights, and maintaining high performance and availability, even as data volumes grow and user demand fluctuates.

In addition, we have deployed a meticulously constructed model in our back-end. This model is thoughtfully organized to support continuous online updates, ensuring it meets long-term demands. This capability ensures that our system not only provides accurate real-time health trend analysis but also continuously improves and optimizes over time, delivering the most up-to-date and reliable health data and predictions to users.

This report will delve into the architecture and functionalities of our cloud-based system, detailing how these technologies integrate to deliver a seamless and insightful user experience. We will explore the data ingestion and processing pipelines, the analytical models used to predict health outcomes, and the visualization techniques employed to present the data in an accessible manner. Through this comprehensive overview, we aim to demonstrate the system's capability to provide valuable health insights at a regional level, thereby contributing to informed decision-making in public health and policy planning.

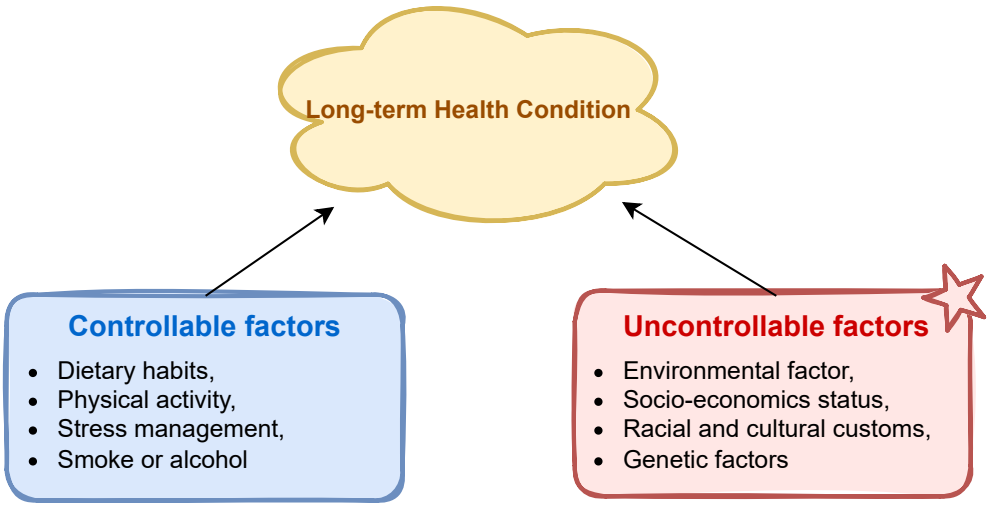

Figure 1: Factors that could make influence to personal long-term health conditions

2 System description

2.1 Scenario Design

In today's world, public health has become a paramount concern for communities and governments alike, especially after the COVID-19 global pandemic. The rise in respira-

tory illnesses, heat-related health issues, and other environment-driven health issues has highlighted the need for awareness of people's personal health conditions.

Imagine the health department of Victoria government tasked with managing and improving the health outcomes of residents across various SA2 regions. The department faces a complex challenge: understanding and mitigating the effects of a variety factors on public health. This includes personal lifestyle choice, environmental impacts, socio-economics status and so on. Among these, the department should prioritize the analysis of those factors that are inherently more significant and have broader implications, as they are often beyond the control of individual residents as shown in Figure 1. For instance, the weather condition, air quality and residents' level of income. Thereby the department can dive into critical, systemic issues that affect the health and well-being of the population at large.

To effectively address these challenges, the public health department will flavor a sophisticated software system that can seamlessly integrate diverse environmental and health data, provide predictive analytics, and offer intuitive visualizations. It must be scalable, flexible, user-friendly, and capable of continuous updates. This system will enable proactive management of health risks and improve community well-being.

2.2 Functionality

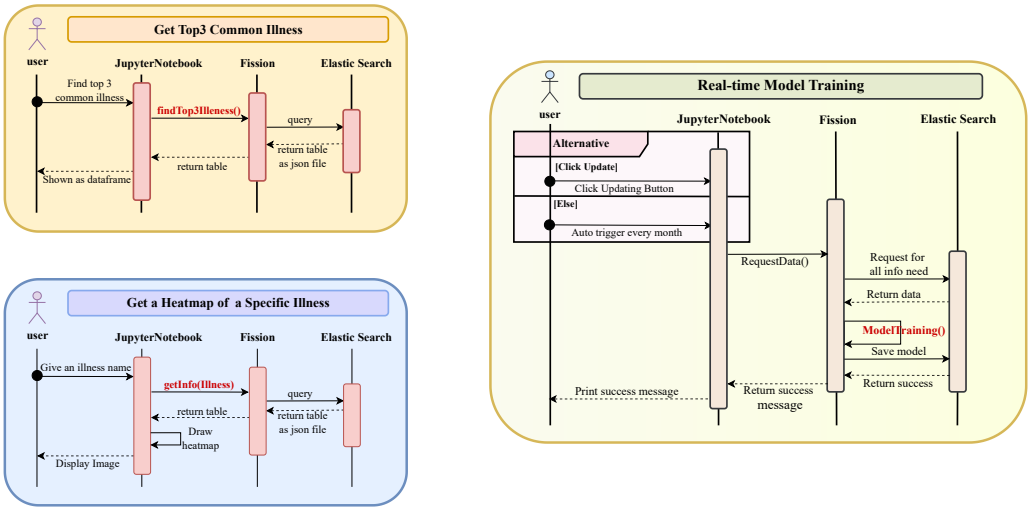

Figure 2: Demonstration of the workflow when the front-end calls content from the back-end to display

Based on the pre-designed scenario, as well as numerous similar ones, we will now delve into the design of the main functionalities of our system. Our endeavor is to craft a robust system capable of meeting the diverse needs under various contexts. The primary objective of our system is to establish an integrated platform, offering a wide array of services tailored to effectively simplify the process of health management. To be more specific, the core functionalities could be split into two stages:

1. An interactive front-end which takes the detailed demands from the client and then displays relative key information and analysis.

2. A strong back-end automatically gathering all datasets and maintain a real-time updated model.

Front-end The primary objective of the front-end is to provide the users with clear and precise overview of the insights generated by our system. We facilitate straight forward display of key statistics as well as intuitive visualizations of data which is demonstrated in Figure 2. For instance, we display live situation of the most severe illness across the entire Victoria at the very beginning to give a overview to the clients. This allows the users to grasp the critical information effortlessly. Furthermore, mining of key statistics enables in-depth analysis on the client side, shedding light on crucial patterns and correlations. Our front-end provides interactive requests, allowing users to freely request data from our integrated database according to their actual needs. For instance, our user might be urgent to gather information regarding a particularly recorded type of illness across a variety of SA2 areas. The system then collects the necessary key data from our database and then processes it into the required format and then present it to the user, displaying all integrated information from various data source. This functionality greatly enhances user convenience, enabling them to easily access the required information and thereby conduct analysis and decision-making more effectively. Meanwhile, we provide instant visualization on the required data, giving a first glance to the desired pattern. For example, we could provide an instantly generated spatial map plot illustrating the severity of the particular illness across the whole Victoria. Besides, we can also meet the clients' demand for real-time, up-to-date analysis. We allow users with specific authority to submit requests through our front-end interface, which will trigger the back-end pre-trained model to immediately update data in real-time and conduct online training to give more precise predictions. It is also possible for the users to get a hint from our model by requiring to see the key features relating to any specific illness. All weights of the features are available to be displayed on the front-end, providing users with the analytical conclusion such as which are the most important weather factors relating to a specific illness.

Back-end To meet the demands of rich data display and model updates on our front-end, our back-end must possess outstanding capabilities in data collection, processing, and mining. Additionally, we need to integrate a pre-trained model that can be updated at any time, ensuring the system maintains the latest data and predictions. To fulfill the duty, we developed a fully automated working pipeline which covers the following:

- Scheduled automatic data harvest
- Predefined data processing function
- Well-structured database answering query requirements
- A model that can be automatically updated at set intervals or on-demand according to user needs.

Apart from the functionality support, we still face the challenge of maintaining the operation of the entire system, especially when receiving large scale of requests from different users at the same time. Therefore we adopted a large cloud-based back-end to fulfill the needs. Further discussions on out choice of techniques and methodologies of construction are presented in section 3.

3 Implementation

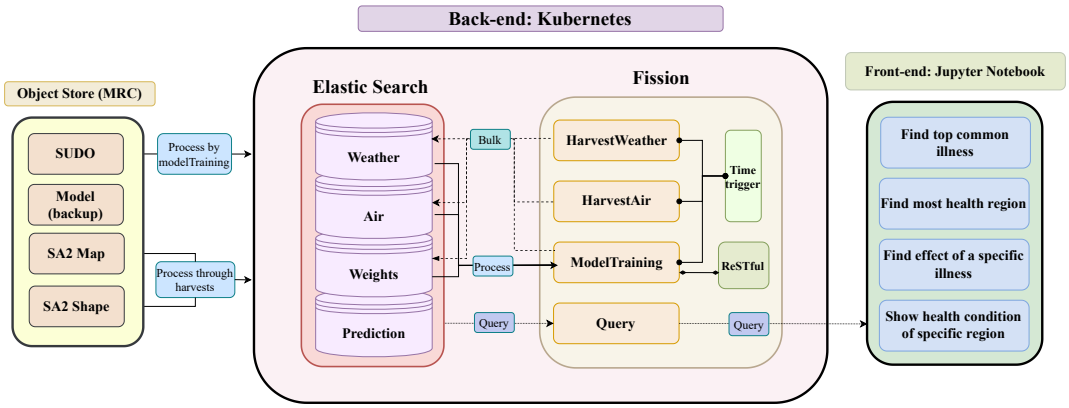

Figure 3: Overall system structure and working pipeline

3.1 Data Sources

3.1.1 SUDO datasets

The project gathers its datasets primarily from the Spatial Urban Data Observatory (SUDO)[4]. From the SUDO website, we specifically chose the "Type of Long-Term Health Condition by Age" dataset[5] with SA2[6] area information. This dataset is particularly relevant to our designed scenario as it provides detailed insights into the prevalence of various long-term health conditions across different age groups and regions. The comprehensive nature of this data allows us to accurately analyse and predict health trends, facilitating a deeper understanding of the health status of residents in each SA2 region. In our project, we choose to use SA2 (Statistical Area Level 2) regions as our primary geographical unit of analysis, rather than suburb postcodes, because SA2 regions provide a standardized and hierarchical geographical framework defined by the Australian Bureau of Statistics (ABS). This ensures consistency and comparability across different datasets and facilitates seamless integration of various data sources. While postcodes may vary in size and population density, SA2 regions are designed to be more homogeneous in terms of population characteristic. This granularity allows for more accurate and meaningful analysis of health trends at the local level.

From SUDO website, we further select the "Total Household Income (weekly) by Household Composition"[7] and "Language used at Home by Sex"[8] because they provide entire information about household income, living environment, and socio-cultural factors, which play a significant role in predicting long-term health outcomes. Household income directly influences an individual's quality of life and health status. The language used at home reflects the family's cultural background and social environment. Different language usage patterns may indicate varying health behaviors, medical preferences, and health literacy levels. Incorporating these datasets into our predictive models improves the accuracy and granularity of our long-term health predictions in each SA2 region. By integrating

socioeconomic and cultural factors, our models can better account for the multifaceted determinants of health. This comprehensive approach enhances our ability to identify at-risk populations, anticipate health trends, and allocate resources more effectively.

3.1.2 EPA - Air quality dataset

The Environmental Protection Agency (EPA) website hosts various datasets related to air quality[9]. we accessed the air quality datasets specific to Victoria (VIC) and New South Wales (NSW) from their respective state EPA websites via API method. These datasets provide valuable information on air pollutant levels in these regions from monitoring stations. Including the external air quality datasets for PM2.5 and PM10 is useful for our project's comprehensive analysis of long-term health. These datasets provide another insights into particulate matter levels, which have the potential to significantly impact health, particularly respiratory conditions such as lung diseases. By adding these datasets, we enrich our analysis, allowing for a multi-faceted examination of long-term health from various perspectives.

3.1.3 BoM - Weather dataset

The Bureau of Meteorology (BoM) serves as a valuable data source for weather-related information across Australia. BoM provides a comprehensive list of monitoring stations, including locations such as Melbourne. This dataset offers geographic information, enabling us to accurately identify and analyze weather patterns in specific regions and provides various weather features from these observations. Each of these features offers valuable insights into current weather conditions, facilitating detailed analysis and forecasting. Sometimes, humid weather conditions can contribute to long-term health issues such as rheumatism. Then, by leveraging BoM's weather data, we enhance our understanding of weather patterns and their impact on long-term health outcomes.

3.1.4 SA2 JSON file

As the datasets obtained from EPA and BoM only provide the longitude and latitude coordinates of the monitoring stations, they do not include information about the SA2 region. Hence, we created our own JSON file for SA2 regions based on the SA2 shapefile obtained from the Australian Bureau of Statistics (ABS) website. This file allows us to determine the SA2 region in which each monitoring station is located. Our custom SA2 region file encompasses all SA2 regions across Australia, with each SA2 region containing its name, code, longitude, and latitude coordinates. This SA2 file we created also plays an essential role in visualizing the number of cases per SA2 region in the front-end. It enables us to generate a heatmap on the map, showcasing the distribution of cases across different SA2 regions. This visualization aids in providing a comprehensive and clear understanding of health trends and patterns, allowing stakeholders to identify areas with higher incidence rates and prioritize intervention efforts effectively.

Table 1: Dataset details. **Source** refers to where we get the dataset. **Author** refers to who creates and manages the raw dataset.**Size** denotes the shape of the datasets. **Frequency** refers to the our predefined collecting interval. **Year** refers to when is the dataset first made up

Dataset	Source	Author	Size	Frequency	Year
Illness	SUDO	ABS	(2453, 15)	Once	2021
Income	SUDO	ABS	(2453, 21)	Once	2022
Language	SUDO	ABS	(2453, 49)	Once	2022
Air Quality	EPA	EPA	()	Daily	Real-time
Weather Observation	BOM	BOM	(66, 48, 13)	Daily	Real-time
SA2 Shapefile	ABS	ABS	(2473, 17)	Once	2021

3.2 Version control - GitLab

Effective version control is crucial for managing the complexity and collaborative nature of our long term health data analytics project. We have chosen GitLab as our version control system due to its comprehensive feature set, which supports collaborative development, continuous integration, and deployment (CI/CD). GitLab offers a centralized platform where team members can manage code repositories, track changes, and collaborate seamlessly.

GitLab provides robust repository management tools that allow for the efficient organization of our code. Each component of our system, from data ingestion scripts to visualization modules, is maintained within the same repository. This centralized approach simplifies code management and facilitates focused development and debugging efforts.

Despite not using branching, we utilize GitLab's commit history to track changes. Each commit is documented with clear messages, detailing the changes made and their purposes. We depend on frequent communication between teams and well-allocated work to minimize conflicts when pushing changes. However, there are still times that conflicting changes occur. Resolve conflict requests are created, enabling peer review and ensuring code quality before integration.

GitLab's integrated CI/CD pipelines automatically build the testing environment and deploy our code. Every commit triggers automated tests, ensuring that new code adheres to our quality standards. Successful builds are automatically deployed to our staging environment for further testing, facilitating rapid iteration and reliable releases. While we currently do not deploy a full CI/CD pipeline due to the absence of a deployment environment for our front-end, we conduct manual acceptance testing for each functionality to ensure comprehensive coverage and quality. This acceptance testing prepares us for future deployment scenarios, providing a foundation for continuous integration.

GitLab includes features for managing user permissions and access controls, ensuring that only authorized personnel can modify critical components of our system. This function remains the security of our code and compliance.

3.3 MRC

The system we have developed is hosted using the infrastructure provided by the Melbourne Research Cloud (MRC), MRC provides infrastructure-as-a-service using OpenStack[10] as the underlying platform. As a open source cloud computing infrastructure project, OpenStack offers various ways of interacting with the resources in the cloud, for this project, we mainly used the Horizon dashboard and the OpenStack client to create, update and monitor the resources.

The are many benefits for using MRC or any other cloud platform. First of all, it is more flexible to host services on a cloud, cloud platform could provide you with the amount of resources you requested, that means the application can be scaled up rapidly as the demand grows or scaled down once demand shrinks. Putting everything on the cloud also means we do not need to worry about the physical hardware that is running the application. On-premises deployment often requires the purchase of the physical hardware, maintaining the system such as cooling, power, network access and so on. For the system we have developed, although we are limited on the amount of resources that we could use in MRC, but that is enough for the current demand of our application, we could handle all the traffic with a cluster formed by a few VM instances, and the cluster could potentially grow if needed.

Accessibility and security is another huge benefit for having the application running on the cloud, there are usually many zones you can choose to create your resources, a region that is closer to the user of the system can reduce the latency for accessing the service. Also, the MRC cloud allows us to control the network by defining ingress and egress rules for different security groups, making the system less vulnerable to cyber attacks. For example, we created a security group to allow port 22(ssh) to be accessed since that is the port we are using to connect to the bastion node and control the system, and for the purpose of demonstration, only ssh tunneling required. If HTTP traffic is necessary for the system, we could simply enable the desired port for the traffic to go through. However, due to the unavailability of the public IP address, we would not be able to host a website publicly.

There are also backup and snapshot mechanisms that comes with the cloud. Backup the volume regularly could protect data from getting lost in case of emergency, incorrect operations and attacks. Creating snapshots when the system is stable and working allows the system to recover once something went wrong and cannot be fixed.

3.4 Kubernetes

We used Kubernetes to manage all the resources we have created on MRC, when we created the cluster using OpenStack, we used Kubernetes as the Container Orchestration Engine(COE). Kubernetes offers features such as horizontal scaling, self-healing, load-balancing which are helpful for managing the cluster, when installing Fission and Elastic-Search on the Kubernetes cluster, we do not need to worry about the physical node the application is running on, these are all handled by Kubernetes itself.

The Kubernetes cluster we have created is composed of 1 master node and 3 worker nodes, each of them has 2 vCPUs and 9GB of RAM. Although each node does not have a lot of computing resources, but since the cluster can scale horizontally, it is much easier to

Table 2: API Interface

Function	URL	Return Type	Description
Affected-percentage	/affected_percentage/{illness}/{percentage}	JSON	Finds the list of areas where a specific illness affects more than a given percentage of residents
Area-analysis	/get_area/{area}	JSON	Finds detailed information about a specified area
Frequent-illness	/frequent_illness	JSON	Finds the list of the most frequent illnesses
Get-importances	/get_importances/{illness}	JSON	Finds the list of important influencing features for a specific illness
Illness-analysis	/get_illness/{illness}	JSON	Finds detailed information about a specified illness
Least-frequent-illness	/least_frequent_illness	JSON	Finds the list of the least frequent illnesses
Least-healthy	/least_healthy	List	Finds the list of the least healthy areas
Most-healthy	/most_healthy	List	Find the list of the healthiest areas
Query-analysis	/analysis/{area}/{illness}	JSON	Finds detailed analysis data for a specific area and illness
Model-training	/train_model	String	Trains and updates the model

scale compared to vertical scaling, as the performance of a single machine would be hard to increase due to hardware limitation. Horizontal scaling would only be a matter of requesting more resources and resizing the cluster to use more nodes and it would be able to handle more traffic.

Declarative management of the application allows Kubernetes to maintain a desired-state that is defined by the configuration. If anything goes wrong, Kubernetes will try self-heal and bring the services back to normal. If a node in the cluster has failed and no longer responding to the master, Kubernetes will try to reschedule the pods that were originally deployed on the failed node, these pods can be scheduled to other worker node that still has capacity. It could also be the case that a particular pod has crushed and stopped working, Kubernetes will attempt to restart the pod and brings the system back to functional state.

Kubernetes also manages the networking within the cluster. Each services are assigned a DNS name that is accessible within the cluster, for example, the ElasticSearch services that is deployed in the *elastic* namespace would have a DNS name of 'elasticsearch-master.elastic.svc.cluster.local'. There are also load balancing mechanisms that distributes the incoming traffic among different pods of the same service, which increases the systems availability and reliability as a failed pod would not impact the deliver of the service.

3.5 Fission

Fission is an open-source serverless framework designed to run functions on Kubernetes, making it a great fit for big data analytics projects. It automatically scales to handle fluctuating data loads by adjusting the number of pods based on demand, ensuring resources are used efficiently. With its serverless architecture, developers can focus on writing the analytical logic without worrying about the underlying infrastructure, which simplifies operations and reduces costs. Fission's ability to deploy rapidly allows for quick iterations, and its integration with Kubernetes ensures high availability and strong resource management. Plus, Fission supports various triggers, making it flexible for event-driven data processing.

We chose Python for our project and configured it using YAML files. For projects that need many external libraries, Fission's package management system is very convenient. Whenever we create or update a function, Fission installs the necessary packages listed in the requirements.txt file during the build phase and packages them into the execution environment. This ensures all dependencies are ready when the function runs, simplifying

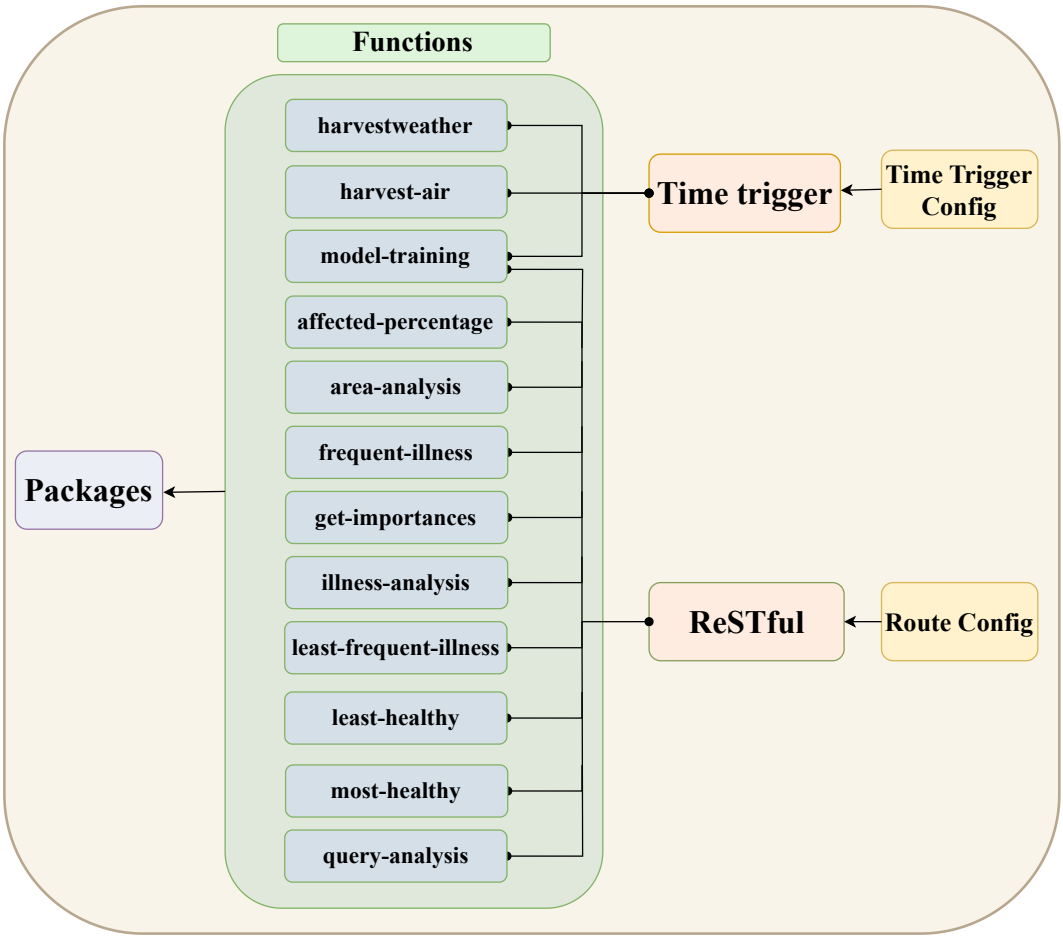

Figure 4: Functions deployed in Fission that are called by pre-set configurations

dependency management and improving efficiency. By creating specification files, we can define the function's name, entry point, environment, triggers and so on.

Fission acts as the back-end of our system, we deployed functions in Fission to ingest data, interact with the ElasticSearch Database and also provide RESTful API endpoints to be used by the front-end.

There are many functions that have been deployed to Fission, a list of such functions could be found in Figure 4. Some of these functions are trigger by a timer periodically, such as data ingestion, and model-training. This is to ensure we harvest the data regularly and update our model so it is up to date, such functions are configured to have a longer function timeout to avoid interruption.

Other functions are mainly used to provide statistics to the front-end application using RESTful endpoints that we have define. The list of endpoints could be found in Table **??**.

But there are still some limitations on the use Fission, for example, the python environments that are provided are limited and restricted. We could not install packages such as shapely and geopandas in the package we created and we need to find workarounds and avoid using these packages.

3.6 Elastic Search

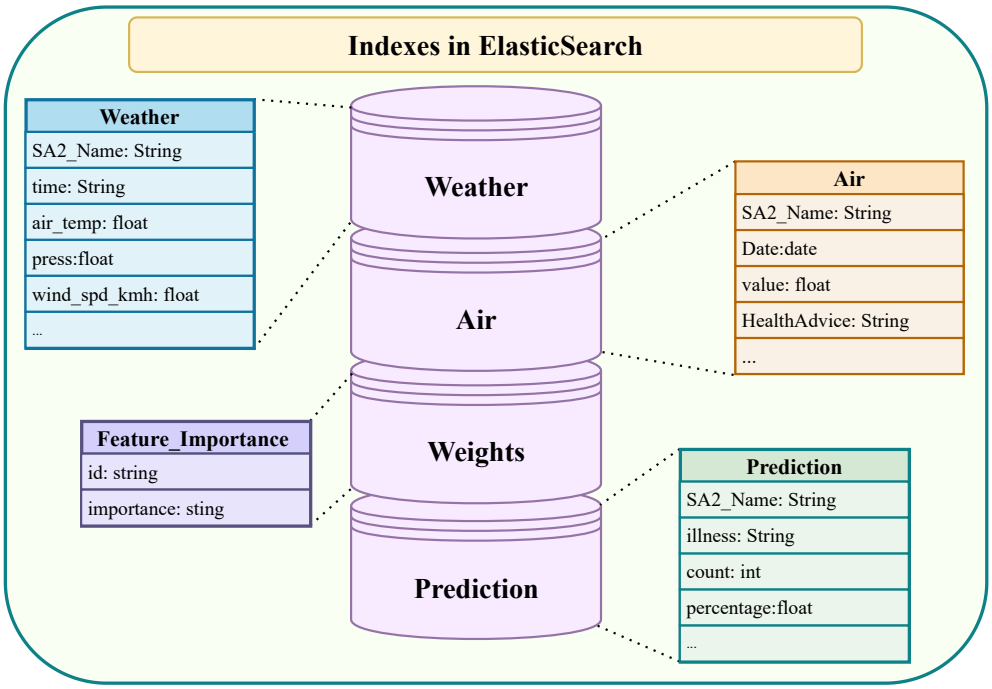

Figure 5: Database structure and data mapping

Choosing Elasticsearch as our database instead of a traditional database is crucial. Compared to traditional databases, Elasticsearch offers scalability, high-performance queries, high availability and fault tolerance, and excellent integration capabilities.

Elasticsearch's distributed architecture allows us to easily scale by adding more nodes to handle increased data and query loads. Compared to the vertical scaling of traditional databases, this horizontal scaling ensures that we can effectively manage growing amounts of data without sacrificing performance.

This distributed setup not only helps manage large volumes of data but also allows for parallel querying across these nodes, significantly speeding up search and retrieval times. Even as the dataset grows, Elasticsearch maintains its performance by balancing the load and effectively utilizing available resources. Moreover, due to its inverted index, Elasticsearch excels at full-text searches and complex queries, making it much faster and more efficient at these tasks than traditional databases. It can handle large amounts of data and return results quickly, which is crucial for our data analytics needs.

In terms of reliability, Elasticsearch uses sharding and replication to ensure data redundancy and fault tolerance. Each index is divided into multiple shards, which can be distributed across different nodes in the cluster. Additionally, each shard can have one or more replicas. When we create Elasticsearch nodes, we distribute them across two nodes. This setup means that if one node goes down, its shards can be seamlessly recovered from the replica shards on the other node, ensuring continuous access to our data and maintaining smooth system operation.

Elasticsearch integrates seamlessly with Kubernetes, allowing us to leverage container orchestration for efficient deployment and management. Kubernetes handles the scaling, load balancing, and failover of Elasticsearch nodes, simplifying our operations and ensuring high availability and reliability. Elasticsearch also integrates tightly with Kibana, providing a powerful platform for data visualization and analysis. Kibana allows us to create real-time dashboards, visualize complex data sets.

We created four indices to store data on weather, air quality, feature weights, and prediction results. During this process, we encountered request timeout errors due to the fact that each individual data entry was making a separate API call to Elasticsearch. To address this issue, we utilized the Elasticsearch client along with the bulk API. This approach allowed us to efficiently batch multiple data entries into single API calls, thereby mitigating the timeout errors and improving the overall data ingestion performance.We also encountered issues when retrieving data due to the large volume, which made it impractical to use Elasticsearch's search method to fetch all records. To resolve this, we utilized Elasticsearch's scroll API, enabling us to efficiently retrieve large datasets in a paginated manner. These challenges are typically not encountered with traditional databases.

3.7 System Analysis

3.7.1 Error Handling

In the attempt to build a robust system that won't easily break, we considered possible failures that could happen and come up with mitigation strategies to avoid them from happening.

1. Data that are ingested through the pipeline could potentially be duplicated, to avoid duplicated data that are inserted into the ElasticSearch, we created a unique id for each document using the time and location of the raw data. This ensures the idempotency of our indexing operation

2. Each service we deployed in Kubernetes has multiple pods running, so that when one of the pod goes down, the system will still be running

3. Data in ElasticSearch is replicated in different volumes, even if one of the volume is damaged or corrupted, the data can still be recovered

4. The cluster can only be accessed through the bastion node, this decreased the likelihood of the cluster being attacked and increased the overall security of the system

3.7.2 Issues and challenges

Through building and implementation of the system, there are several issues that came up.

First is the lack of public IP addresses allocated by MRC. The exhaustion of IPv4 address is a common problem nowadays, but it has caused quite few issues we have to workaround. The direct impact is that the application would be impossible to access without using VPN, not available to the public means it would unlikely to be used by many people and the demand won't be increasing. And it would be hard to integrate CI/CD pipelines with gitlab, since deploying and controlling the system requires connection to the VPN at the first place.

Another issue we have encountered is regarding the exhaustion of disk space on some nodes in the cluster. This issue arises when applying the specification using Fission, the disk space on a node in the cluster might have been exhausted due to cache not cleaned properly, and the new installation of the package would fail under this scenario.

4 Analytics, Modelling and Visualisation

4.1 Data Preprocessing

The *long-term health dataset* includes detailed information on the number of individuals in each age group who suffer from various illnesses within each SA2 region. To make this dataset more suitable for our specific setting and analysis, we have refined it to focus on the total number of people with each illness in each SA2 region. This preprocessing step simplifies the data, allowing us to concentrate on the overall prevalence of each illness rather than breaking it down by age group. By refining the dataset in this way, we are able to create individual predictive models for each illness. Each model is designed to forecast the number of people affected by a particular illness in each SA2 region. This approach enables us to generate more accurate and targeted predictions, facilitating better resource allocation and intervention planning. Based on the refined dataset, we are able to conduct future analyses on various long-term health conditions. The illnesses included in our analysis are as follows:

- Arthritis
- Asthma
- Cancer (including remission)
- Dementia (including Alzheimer's)
- Diabetes (excluding gestational diabetes)
- Heart Disease (including heart attack and angina)
- Kidney Disease
- Lung Conditions (including COPD and emphysema)
- Mental Health Conditions (including depression and anxiety)
- Other Long-Term Health Conditions
- No Long-Term Health Conditions
- Not Stated
- Stroke

By focusing on these specific health conditions, our analysis can provide detailed insights into the prevalence and distribution of each illness within different SA2 regions.

Using the *weather dataset*, we noticed that certain weather features were duplicated with different units. To maintain consistency and simplicity, we retained the most commonly used unit for each weather feature. Additionally, we preserved the latitude and longitude coordinates of each monitoring station. This geographic information allows us to accurately identify the SA2 regions affected by the weather conditions recorded at each monitoring station. By standardizing the units and keeping the essential location data, we can more effectively integrate the weather dataset with our health analysis. This ensures that our predictions are both accurate and relevant to specific regions, facilitating better analysis of how weather conditions influence long-term health outcomes across different SA2 regions.

We retained key weather features including gust speed, apparent temperature, temperature difference, air temperature, atmospheric pressure, rainfall accumulation, relative humidity, visibility, and wind speed. These features were chosen because they are commonly available and easily accessible in real-life scenarios, for example, most weather apps on smartphones provide this data, making it familiar and comprehensible to the general public. We also kept the time and date information for each weather record. Since our weather data is harvested from BoM, including each live monitoring station, the timestamp of each record is crucial for accurate analysis. This temporal data allows us to provide a more dynamic and real-time perspective on how weather influences long-term health trends. By maintaining detailed time and date information, we can track changes and patterns over specific periods, and simplify the process of integrating the weather data into Elasticsearch, our database. The detailed time and date information ensures that each weather data entry remains unique and avoids overlapping.

Due to the lack of SA2 region information in each weather dataset, we utilized our designed SA2 JSON file and the location information of monitoring stations to determine the SA2 name and code associated with each monitoring station, facilitating seamless linkage with long-term health data. We approximated the weather data for SA2 regions without monitoring stations using a 5-nearest neighbors (5NN) approach coupled with interpolation. With 66 weather stations available, we applied interpolation techniques based on the 5NN model to generate approximate weather data for regions lacking a monitoring station. This involved leveraging a linear model that integrates both K-nearest neighbors (KNN) and interpolation methods. By employing this approach, we were able to predict the weather conditions for SA2 regions without monitoring stations, thereby enhancing the comprehensiveness of our analysis.

Regarding the *air quality dataset*, we collected air quality data from both Victoria (VIC) and New South Wales (NSW). The VIC air quality data includes the PM2.5 pollution value along with health advice labels such as Good, Fair, Poor, Very Poor, and Extremely Poor. Conversely, the NSW air quality data provides the PM10 pollution value. Due to the discrepancy in the pollution values provided by the two states, we referenced the health advice labels in the NSW air quality data to standardize the air quality assessment. This approach allows us to unify the air quality assessment across both datasets, ensuring

consistency and accuracy in our analysis.

For the *income dataset*, it includes information on weekly household income, which is segmented into household and non-household weekly income within each income bracket. However, distinguishing between household and non-household income can be challenging and ambiguous. Therefore, we opted to retain only the total weekly income which sums up the household and non-household for each income range in each SA2 region. This simplification ensures clarity and consistency in our analysis, enabling us to focus on the income distribution across different geographic regions.

For the *language dataset*, the dataset provides counts of female and male individuals speaking various languages at home within each SA2 region. Since the dataset already counts for the number of people using each language at home, it effectively captures the unit of measurement by household language use. Consequently, we decided to retain only the total number of persons speaking each language at home in each SA2 region. This approach simplifies the dataset while preserving its essential information, allowing us to analyze the linguistic diversity and cultural composition of different SA2 regions. We included a comprehensive range of languages, such as English, Arabic, Cantonese, Mandarin, French, German, and many others. This dataset is highly detailed, covering all common languages as well as the majority of less commonly spoken languages. By encompassing such a wide array of languages, we aim to reflect the ethnic diversity and multicultural composition of the population in each SA2 region. This linguistic data serves as a proxy for the variety of national backgrounds and the diversity of communities, providing valuable evidence for analyzing long-term health and lifestyle.

4.2 Modelling

We merged the weather, air quality, income, and language datasets with the long-term health dataset based on SA2 name and SA2 code. However, the air quality data contains numerous null values, indicating that the monitoring stations did not publish current air quality data consistently. Consequently, relying on the air quality data significantly reduces the number of SA2 regions with complete and valid values for every feature. Additionally, the limited number of air quality monitoring sites means that not all regions are adequately covered. To address this issue, we decided to exclude the air quality data from certain analyses to ensure a more comprehensive and inclusive dataset. By focusing on the more consistently available weather, income, and language data, we maintain a robust dataset that still provides valuable insights into long-term health trends across the SA2 regions. This approach allows us to perform a more extensive and reliable analysis, highlighting the key socio-economic and environmental factors influencing public health.

We selected the Random Forest Regressor model [11] to build our predictive models for each long-term illness. The Random Forest Regressor model is well-suited for this task due to its robustness and ability to handle high-dimensional data effectively. It combines multiple decision trees to improve predictive accuracy and control overfitting, making it particularly advantageous for our complex dataset that includes various socio-economic

and environmental factors. Additionally, the model's capability to estimate feature importance helps in understanding the impact of different variables on health outcomes, thereby providing more insightful and actionable results.

We developed individual models for each illness to predict the number of people likely to be affected in each SA2 region. Hence, we built up 14 models to predict the number of people likely to be affected by each illness in each SA2 region. The performance of these models is evaluated using Mean Squared Error (MSE), as it measures the average of the squares of the errors, providing a clear indication of the difference between the predicted and actual values. A smaller MSE value indicates a better model performance. The results are shown in Table 3.

Table 3: Mean Squared Error (MSE) values for different long-term health conditions.

Illness	**MSE**
No Long-Term Health Conditions	0.00124
Arthritis	0.00023
Asthma	0.00015
Stroke	0.00004
Kidney Disease	0.00001
Heart Disease (including heart attack and angina)	0.00025
Not Stated	0.00094
Cancer (including remission)	0.00011
Dementia (including Alzheimer's)	0.00013
Other Long-Term Health Conditions	0.00014
Mental Health Conditions (including depression and anxiety)	0.00024
Diabetes (excluding gestational diabetes)	0.00064
Lung Conditions (including COPD and emphysema)	0.00003

Based on the MSE values, for most long-term health conditions, the model's prediction errors are relatively small, indicating good predictive capabilities. For instance, diseases like "Kidney Disease" and "Lung Conditions" have MSE values very close to zero, suggesting that the model's predictions are close with the actual values.

4.3 Visualisation - Front End

In the visualization component of our front end, we have designed two primary scenarios. The first scenario addresses users who want to know the percentage and count information for a specific illness, and the second scenario caters to users who want to know this information based on a specific SA2 code.

For the illness-specific scenario, we provide bar charts that display the percentage and count across different regions in descending order. This allows users to easily identify the most and least affected regions. By presenting the data this way, users can better understand the numerical impact of the illness across SA2 regions.

Additionally, due to the large number of SA2 codes, summarizing all the information clearly in a bar chart can be challenging. To address this, we also offer a geometric approach using a heatmap overlaid on a real-world map to illustrate the distribution of the given illness for both percentage and count. The map is free to zoom in and out which could help to find more specific information if needed.

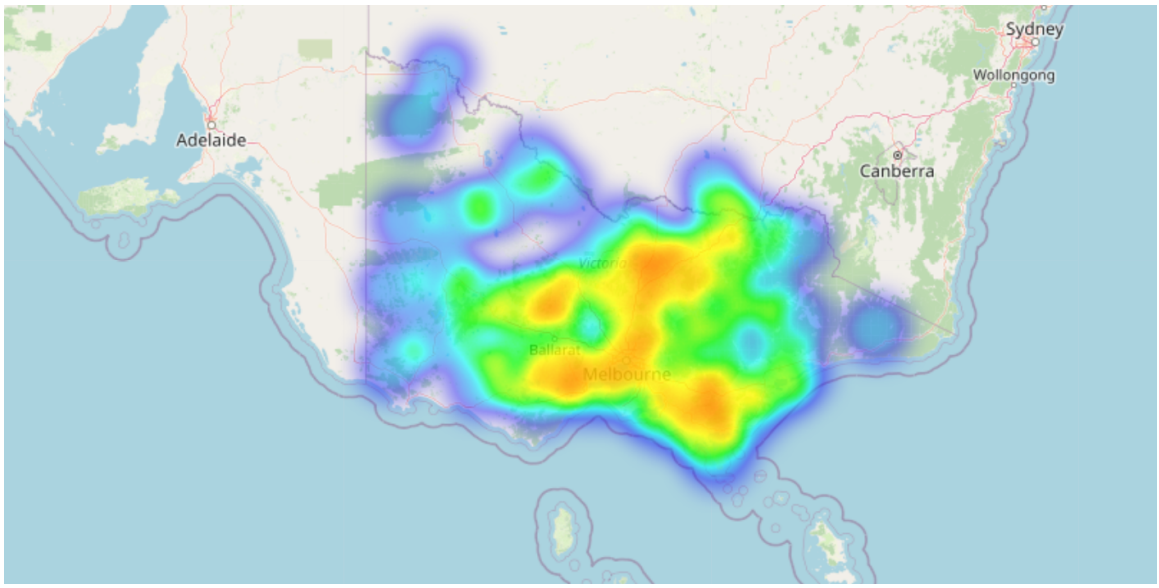

Figure 6: Heatmap for the percentage of influenced patients of Asthma

On the other hand, for the scenario where a specific region is given, we also provide bar charts displaying the percentage and count of illnesses. Since the number of long-term illnesses included in our research is relatively small, it is straightforward to present this information in a bar chart, making it easy to understand the distribution of illnesses within the region.

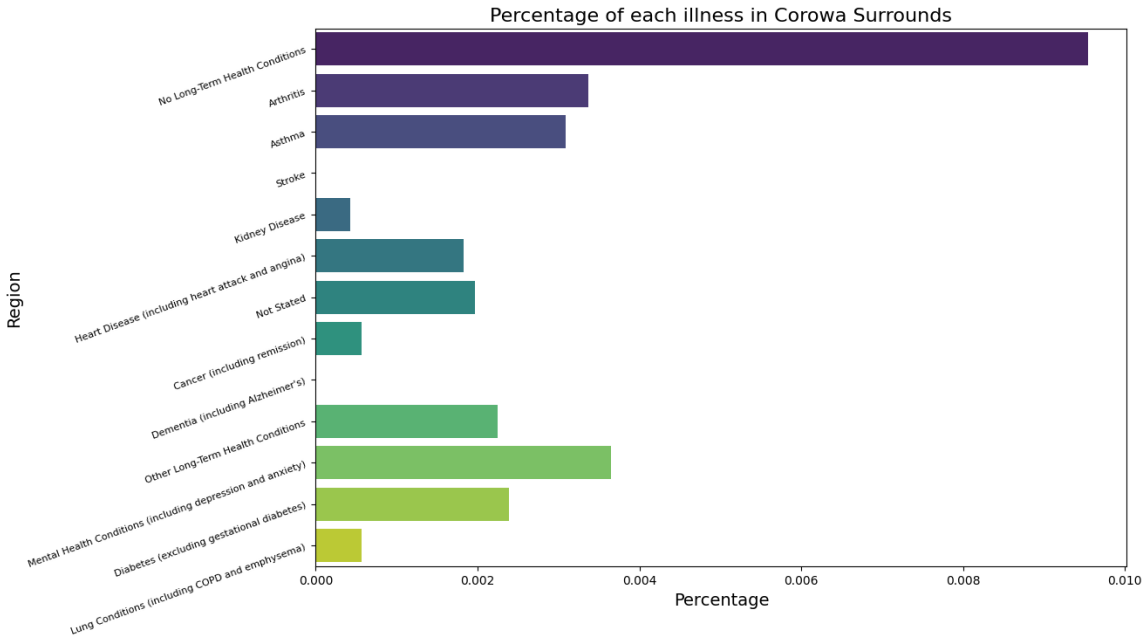

Figure 7: Bar chart for the distribution of probability of infection of each illness in Corowa Surrounds

In conclusion, these two approaches in the interactive display section effectively illustrate the distribution of a given region or illness in both numerical and geometrical ways. This enables our users to easily understand our predictions and to summarize the information through visual representations, providing a clear overview of the distribution. This method is more user-friendly compared to navigating through a detailed dataframe, which can be harder to interpret despite offering more granular information.

5 Limitations

5.1 Dataset

The datasets we utilized have constraints. There is a scarcity of weather stations and air quality stations, which limits the accuracy of weather data in each SA2 region and also leads to the exclusion of air quality features, as fewer SA2 regions could be predicted. Additionally, the SUDO datasets may not be up-to-date as we are using the datasets published in 2021 and 2022, potentially affecting the accuracy of our analysis. Moreover, inconsistencies in the timelines of different datasets pose challenges, as they may not align perfectly, leading to potential inaccuracies in our models. This is because the weather and air quality datasets are harvested in real-time, while SUDO datasets are from the past.

Furthermore, certain datasets, such as air quality and weather data, may be interrelated. Also we lack the geographical background knowledge to ensure their independence. While we attempted to approximate weather data for regions lacking monitoring stations, our approach may overlook geographical nuances, such as mountainous or desert regions, potentially introducing biases into our analysis. These geographical differences can significantly impact local weather patterns and air quality, potentially leading to substantial variations in data values. For instance, despite proximity to a monitoring station, a particular location may

experience vastly different weather and air quality conditions due to its unique geographical features.

5.2 Model

Only relying on Random Forest Regression model might miss out on capturing more detailed data relationships. RFR works fine with big datasets and performs well in current working, but it might not catch all the complex patterns effectively.

5.3 Others

Another limitation related with our assumption that language usage reflects lifestyle habits. While English may be the predominant language spoken, variations in lifestyle habits among English speakers are not accounted for. Therefore, further exploration is warranted to delve deeper into the nuanced relationships between language and lifestyle habits.

6 Future Improvements

To address those limitations, several ways for future improvement can be explored.

Firstly, having more weather stations and making sure our datasets are up-to-date would make our predictions more accurate. Also, updating SUDO datasets to have the latest demographic and health information would make our analyses more relevant and reliable. Integrating different datasets, like air quality and weather data, could help us understand how environmental factors affect health outcomes better. Moreover, getting more information about geography to make sure our datasets are independent and considering geographical differences in our analysis would make our models more accurate and reliable.

Secondly, exploring advanced modeling techniques beyond Random Forest Regression could help capture more nuanced data relationships. Techniques such as neural networks or ensemble methods may better capture complex patterns in the data, leading to more accurate predictions and insights.

Thirdly, continued refinement of assumptions, such as the relationship between language usage and lifestyle habits, is necessary. Conducting further exploratory analysis to understand the details of these relationships and incorporating additional socio-cultural variables beyond language usage would provide a more comprehensive understanding of lifestyle habits and their impact on health outcomes.

7 Teamwork and Contribution

In this section, we will describe the general workflow and the role of each team member during this large project. Given the wide range of techniques involved in the software development process, we have intentionally structured our team as follows:

We have two members with strong IT backgrounds and three others with strong data science backgrounds. This ensures that we always have specialists in back-end construction and maintenance, providing solid support for the development environment. Meanwhile, the

other team members are experienced in data processing, analysing and modeling, delivering insightful contents to display through our project.

In addition, our team employed various tools and practices to ensure effective collaboration and efficient project management. We communicate through WeChat and Zoom for group meetings, conducting at least two meetings per week to discuss progress, address challenges, and plan upcoming tasks. We used WeChat to communicate anytime and anywhere. If any team member encountered a problem or made progress on their tasks, they would share updates and discuss solutions in our WeChat group. This real-time communication ensured that everyone was informed and could contribute to solving issues promptly, enhancing our overall collaboration and efficiency. These tools facilitated real-time communication and decision-making, ensuring that all team members were on the same stage.

The overall duty and contribution of each team member is as shown in Table 4

Table 4: Contribution table.

Team Member	Role	Contributions
Yiyang Huang	Back-End Developer	Deploying applications using Fission, and overseeing Kubernetes (K8s) operations
Mingyao Ke	Back-End Developer	Data management tasks, including integrating data into Elasticsearch and ensuring the database's efficiency. Additionally, he contributed to deployment activities using Fission
Yuchen Luo	Front-End Developer & Project Manager	Finalizing the frontend, and integrating the code
Lora Zhong	Data Scientist	Collected and processed raw datasets, integrated datasets from various sources, developed predictive models
Jiaqi Fan	Data Scientist	Developed predictive models, and conducted key data analysis and insight information based on interpretability of the model

The general workflow of our team could be concluded as the four parts following:

Scenario Design We first identify the main focus of our project by brainstorming scenario and user demand together, pointing out interested topics and potential datasets involved.

Preliminary Construction Then we move on to our primary construction of the back-end as well as the dataset. We are split into two groups to work in parallelized manner. Yiyang and Mingyao are responsible for establishing the structure of the whole back-end, including setting up Kubernetes configurations and Elastic Search deployment. Meanwhile, Yuchen, Jiaqi and Lora delves into searching for datasets relevant to our topic from various databases and starts harvesting them locally.

Collaborate Development At this stage, most data collection, processing and modelling have been done locally and are waiting for deployment into the back-end. This requires frequent meeting and efficient communication within our team since we need complement skills from each other. For instance, Lora and Jiaqi need to clarify the index and mapping of each collected data so that Mingyao can establish the query functions in Elastic Search.

Front-end Design As the final part of the whole project, the front-end is designed to display the key insights to match up with our pre-defined scenarios and demands. This part also requires efficient communication with the back-end as we need to develop the API interfaces to call and display the data on the front-end.

Our team has demonstrated excellent communication skills and a positive attitude, which has helped us quickly complete most of the integration work. Additionally, when detailed issues arise, everyone promptly raises and resolves them. For instance, when developing the API interfaces, the front-end required data linking SA2 areas with a specific illness. The back-end implemented this function but provided SA2 codes instead of SA2 area names, which is not user-friendly for the front-end. Such minor issues are common since our back-end developers are not familiar with data science principles, but they are easily addressed through instant communication. We are always clear and precise in giving feedback on each other's work, and everyone is willing to embrace any negative comments on their own work and make improvements quickly.

8 Conclusion

In conclusion, our project represents a step forward in using big data analytics to address public health challenges at the regional level. By integrating diverse datasets and employing advanced technologies, we have developed a comprehensive system capable of providing valuable insights into long-term health trends across different SA2 regions.

The back-end of our system, built on a robust cloud infrastructure, ensures scalability, efficiency, and reliability in processing and analyzing large volumes of data. Utilizing technologies such as Fission, Elasticsearch, and Kubernetes enables us to meet the requirements of modern big data applications and offering real-time insights.

Through our front-end interface, users can easily access key statistics and intuitive visualizations, facilitating in-depth analysis and informed decision-making.

However, there are still some challenges that need to be overcome in the future to further improve our system.

Overall, our project demonstrates the power of data-driven approaches in addressing public health challenges. Here is a video demonstrating the core functionality of our system is available on YouTube: `https://youtu.be/01mA06YzGE8`. The source code can be accessed on GitLab: `https://gitlab.unimelb.edu.au/yiyahuang/comp90024-2024-grp-32.git`.

References

[1] Fission. https://fission.io/.

[2] Free and open, distributed, restful search engine. https://github.com/elastic/elasticsearch.

[3] Open source system for automating deployment, scaling, and management of containerized applications. https://kubernetes.io/.

[4] Spatial Urban Data Observatory (SUDO). https://sudo.eresearch.unimelb.edu.au.

[5] Long-Term Health Conditions - Australian Bureau of Statistics. aurin:datasource-AU_Govt_ABS_Census-UoM_AURIN_DB_Census2021_abs_2021census_i11b_aust_sa2.

[6] Statistical Area Level 2. https://www.abs.gov.au/statistics/standards/australian-statistical-geography-standard-asgs-edition-3/jul2021-jun2026/main-structure-and-greater-capital-city-statistical-areas/statistical-area-level-2.

[7] Total Household Income(weekly) by Household Composition - Australian Bureau of Statistics. aurin:datasource-AU_Govt_ABS_Census-UoM_AURIN_DB_Census2021_abs_2021census_g33_aust_sa2.

[8] Language used at Home by Sex - Australian Bureau of Statistics. aurin:datasource-AU_Govt_ABS_Census-UoM_AURIN_DB_Census2021_abs_2021census_t10a_aust_gccsa.

[9] AirWatch - Environment Protection Authority Victoria. https://www.epa.vic.gov.au/for-community/airwatch.

[10] Open source cloud computing infrastructure. https://www.openstack.org/.

[11] Random forest regressor. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.