VISUALISING TRAVEL TIMES & ROUTE CHOICES ACROSS GREATER ADELAIDE

PROJECT DELIVERABLE

Phouthun Nay, Jinxi Luo, Vivek Vijayakumar, Vivek Nimmagadda

November 5, 2021

Executive Summary

- AddInsight is an Intelligent Transportation System installed by the SA Department for Infrastructure and Transport (DIT) across Adelaide's Road network that is used to record passing vehicles through their Bluetooth devices. It constantly generates large streams of data by logging data about vehicles passing by its sensor sites.
- The objective of the project was to transform the AddInsight data and derive valuable insight by creating dashboards which could display information such as travel times and most frequently used routes by passengers, therefore assisting in the decision-making process at the DIT
- In the dynamic dashboards, clients are given the option to select the routes and sites of their choice, with the corresponding details being displayed to them.
- A comparison between Tableau and Microsoft BI as visualisation tools for the data was made to find the most suitable tool. Tableau was found to be more capable in visualisation geographical data, particularly SA1 zones.
- The main challenge encountered during development was querying and processing the large amount of data in a reasonable time to derive the necessary insights from it. Optimisation of the code was necessary to reduce code execution time to as short as possible. Further enhancements such as parallelisation of the code may be possible in the future.
- As team members could not work with the entirety of the available data, a single day of data, consisting of about 12 million rows, was used for the development of the data transformation pipeline and visualisations.
- All code deliverables for data transformation, in either Python or R, were written to be scalable to a larger dataset than the one used in the project.
- Multiple visualisation dashboard prototypes were created, displaying the route choices, travel times and its variations at different hours or during different days, based on the 1 day of processed and transformed data.

Table of Contents

1. Introduction

The Department for Infrastructure and Transport (DIT) is the department of the Government of South Australia that is responsible for transportation services, infrastructure planning and provision all over South Australia, they have diverse responsibilities which include implementing good infrastructure planning around the state and proper transport and traffic controls. The main aim of the organisation is to ensure the proper functioning of transport systems and services across the state. More details related to the department can be found at https://www.dpti.sa.gov.au/.

Currently, as Adelaide's population rapidly increases, one of the fundamental areas of concern is the problem of traffic congestion. This sudden increase in the number of people willing to migrate to Adelaide can be attributed to several factors such as an increase in job opportunities, affordability, ease of settling in, etc. Therefore, to mitigate traffic congestion and increase efficiency, the DIT needs to be able to plan development of new infrastructure projects ahead in time. This naturally requires large amounts of data regarding travel times, route choices and other traffic information, gathered through an Intelligent Transportation System (ITS) known as AddInsight, which consists of a network of more than 1000 sensors that are located throughout the Metropolitan regions of Adelaide. These sensors collect Media Access Control (MAC) addresses from nearby passing Bluetooth and Wi-Fi enabled vehicles and devices and store the data for further analysis. An average day of data consists of over 12 million rows of data and about 3.5 million unique probe ids.

This report follows from the project plan and details implementation of previously listed objectives utilising data from AddInsight that was retrieved from the client's AWS data warehouse. Documentation of the processes and data used is included as well as results and encountered issues.

Research was conducted by each team member to discover packages and capabilities of the various available technologies, with a specific look at the functionality offered by AWS, SQL Workbench, Tableau, Microsoft Power BI, Google API and Maps to produce dashboards and transform data. The shortcomings of each technology were identified and through communication between the Mentor, Industry Supervisor and Client, the technologies most suited for the continuation and success of the project were selected. This includes Tableau and Power BI for dashboard creation, Python for processing and travel time estimation and R for time series analysis.

The initial stated project scope was completed with the inclusion of some changes to the plan when re-evaluating the feasibility of the work according to the timeframe with the client. These changes mainly pertain to travel time anomaly detection as potential work that could be implemented if time is available, and the addition of multiple dashboards that will be provided to the client to meet their multiple objectives of interests. See the folder DDTI1 PROJECT FILES.zip and refer to figure 1 shown in the following section for more information on the deliverables that have been produced and delivered to the client, including any code used for data cleaning and feature creation. After the initial few weeks of meetings with the client, the requirements of the project were briefly outlined which are as follows:

- Build visualisations dashboard that lets the users inspect statistical information for travel times between any two chosen locations in the AddInsight network, as well as between different SA1 zones.
- Build the dashboards in both Tableau and Power BI and compare the features available between both software.
- Identify best route in general from an origin and destination and the most popular routes.
- Forecast the traffic patterns for the upcoming months.

1.1 Organisational Benefits

There is a large amount of data adding up to 12 million or more data points per day that is constantly being generated in the AddInsight system and deriving valuable insight from it is a complicated task involving processing big data and feeding the required log files continuously to analysis tools and algorithms. This project provided help for the organisation in understanding the data by presenting a prototype of how the above process can be implemented in an offline setting as well as presenting a discussion regarding the available technologies and results by which the DIT can make future decisions to potentially increase the number of sensors within the AddInsight system, thus allowing better understanding of driver behaviours, road usage and traffic conditions.

The deliverables provide a clear view about the route choices opted by drivers, showing summary statistics of travel times taken by the drivers to reach their destination from an origin. There is a dashboard that is provided to the end-user where they can view the different routes chosen by drivers to reach their destination and show the frequently used routes by other drivers. This analysis is also replicated in an aggregated manner by Statistical Area 1 (SA1) zones, as defined by the Australian Bureau of Statistics, offering a high-level perspective. Knowing this information, the organization can take necessary action in improving the infrastructure across the city and can also help in controlling the traffic across the state.

2. Literature Review

Based on the requirements, a thorough literature review was conducted to explore the existing research in the current domain, and it was found that most of the work revolved around the concept of using either Google maps, Open-Street Maps, or Bing Maps (Bainbridge 2016) for calculating the travel times and displaying that information using visualising software such as Tableau.

According to (Lau 2020), people around the world drive about 1 billion kilometres each day using Google Maps. So, providing the best customer experience in the form of accurate predictions might be extremely vital for maintaining a healthy relationship with them. Although this process appears to be simple and straightforward, it involves a lot of complex computations on the back end to deliver the best possible result in a hassle-free and quick manner.

According to (Barth 2009), whenever the GPS preferences are set to on, the app automatically transmits anonymous bits of data back to their server indicating the speed at which the current vehicle is moving, and all such data collected from different vehicles along the route will then be aggregated and combined with traces of historical data to make the predictions using advanced machine learning algorithms and forecast the duration of the trip.

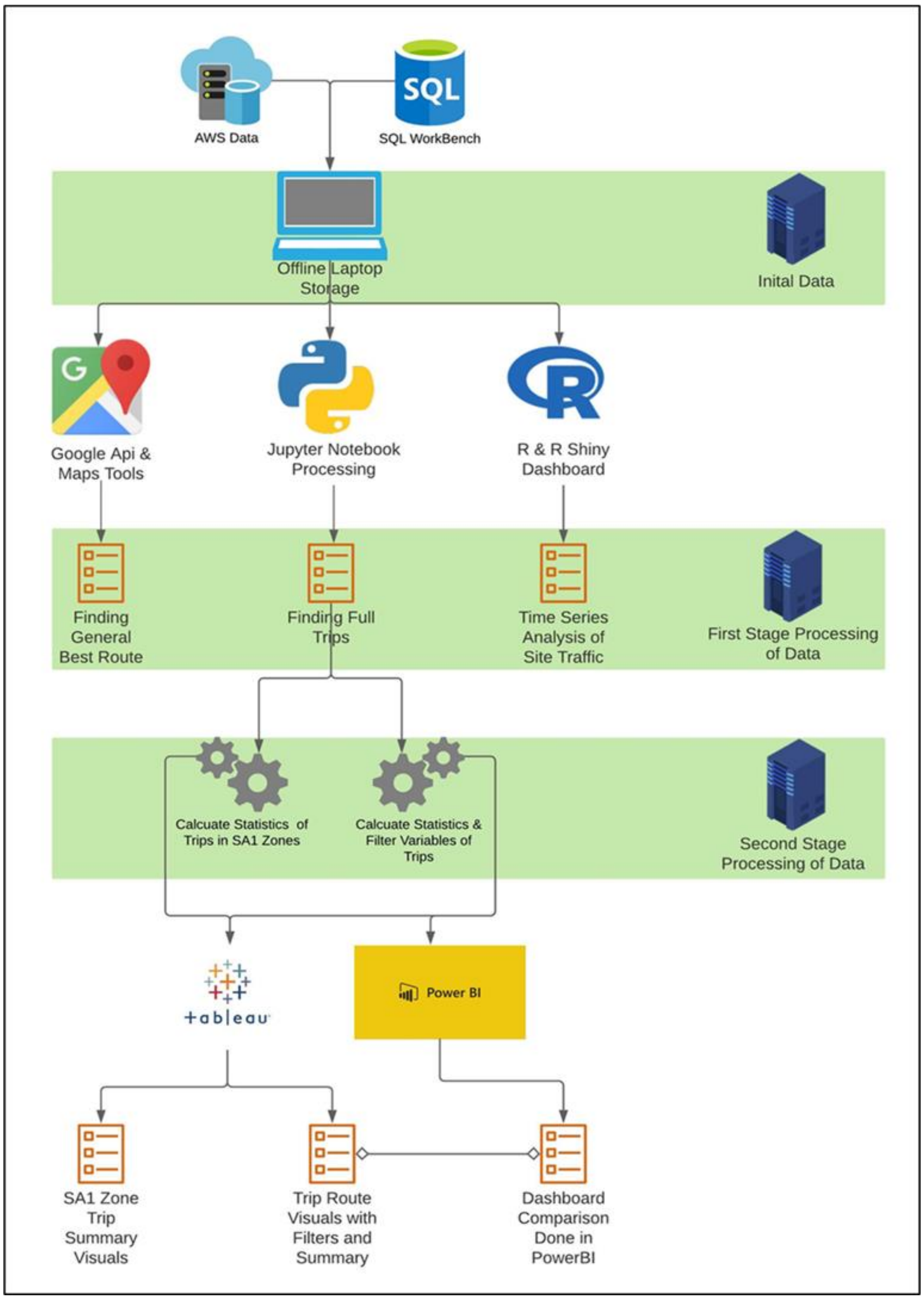
Recent research conducted by (Derrow-Pinion et al. 2021) at DeepMind, an Alphabet AI research laboratory, introduced the concept of applying Graph Neural Networks for cutting down the negative estimated time of arrival (ETA) which had been deployed in the production of Google Maps. This advanced machine learning architecture improved predictions in densely populated cities such as Sydney, Washington D.C., Berlin, and Tokyo by more than 40%. The algorithm also scrutinizes the traffic patterns that are close to the selected path and considers the total number of signals that are encountered on the current path thereby improving the accuracy by lowering the negative ETAs.

(Dauletbak and Woo 2019) shows how Power BI tools was used to visualise areas with heavy traffic. The authors used the add-on Excel 3D Maps tool to build a geographical view of the data using 9 days of UK traffic data, as well as performing some data cleaning and pre-processing steps. The data was then sorted to a range between 1 to 5 according to traffic volume. Afterwards, the processed data was exported and loaded it into Power BI, where they built a geographical view from the data using the Excel 3D Maps tools as well as a heat map, utilising different colour tones to distinguish between different traffic congestion levels. For example, red meant then there likely was an accident and yellow meant that there likely was a road closure at that time. The authors also produced a bar chart to show what the traffic patterns at different times during the day.

Having reviewed the existing literature on the topic to guide our implementation, the various data exploration and pre-processing steps that were undertaken are explained hereafter and aimed to address some of the issues and outline the process by which each deliverable was produced and the reasoning behind how they address the client's requirements.

3. Data and Techniques

This section details an overview of the techniques and data used in achieving the project's objectives. Figure 1 below shows the connection between technologies used and the steps taken which are separated into three sections (Initial Data, First Stage and Second Stage) which are discussed below. Each stage is separated based on different tasks that were performed using increasingly complex data that were pre-processed.



AWS Data
SQL WorkBench
SQL
Offline Laptop Storage
Inital Data
Google Api & Maps Tools
Jupyter Notebook Processing
R & R Shiny Dashboard
Finding General Best Route
Finding Full Trips
Time Series Analysis of Site Traffic
First Stage Processing of Data
Calcuate Statistics of Trips in SA1 Zones
Calcuate Statistics & Filter Variables of Trips
Second Stage Processing of Data
+ableau
Power BI
SA1 Zone Trip Summary Visuals
Trip Route Visuals with Filters and Summary
Dashboard Comparison Done in PowerBI

Figure 1: Shows the data pre-processing steps undertaken and technologies used for tasks performed

3.1 Initial Data and Decision Making

Tools

- AddInsight [Data Context]
- AWS [Data Sourcing]
- SQL Work Bench [Data Sourcing]
- Jupyter Notebook [Joining Datasets]

The project initially began with a look at the data available on the AddInsight system and API, which proved difficult to procure the large amounts of data necessary for analysis from. This was mainly due to the data being retrievable only through a REST API GET method call, and computational processing time limits which our queries exceeded. Following discussions with the client, access to their AWS cloud data warehouse, which backed up the data collected from AddInsight, was arranged, and made accessible through SQL Workbench/J for each member.

Using this access, members had observed that the initial target of real time data processing and target of working with a year's worth of data for 2021 was not feasible. Organizational restrictions from AWS had limited the capability of accessing the data directly and in real time. Additionally, the size of available data compared to the available storage and download speeds to team member's internet and laptop machines, meant that it was not possible to download the full year range of data. In perspective, it took around 13 hours to download a full year's worth of traffic data for all sites available in the AddInsight network.

After, further discussion with the mentor and client, the project scope was adequately re-evaluated to providing a prototype of offline implemented deliverables using just a day's worth of traffic data. Jupyter Notebooks was employed for initial data processing, as requested by the client due to their organisational familiarity with Python and ease of presentation of the necessary data processing steps.

Procedures

The relevant data downloaded from AWS was separated in two different files shown below:

- btrecords.csv [Log Data]
- btsites.csv [Sensor site coordinates]

The task was to first join the files so that a complete data set containing 12,825,998 traffic log records also included the coordinates of each 1,053 unique site ids as well as its name describing the site location in English. Figure 2 below shows the features of interest in each file and the connections:

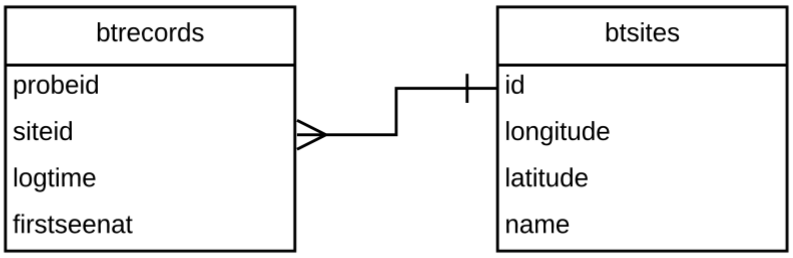


Figure 2: UML diagram of the connections made between the two datasets

3.2 Data Exploration:

The two most important log files for our analysis are btsites and btrecords that were readily available from the SQL Workbench. The data corresponding to these files was dynamic i.e., they get updated daily from the data recorded by the sensors. The former dataset is a collection of all the sensors installed around Adelaide. While the latter is the actual dataset consisting of the log files of unique device information collected by these sensors. These datasets and their exploratory analysis are detailed below.

<u>BTSites:</u> The data in Table 1 allows inspection of each btsites closely using some tables and visualisations.

Variable	***Description***	***Data Type***
dms_update_ts	Updated time stamp of a sensor.	datetime
id	Unique ID representing a sensor.	int
enabled	True/False value indicating whether the sensor is currently active or not.	Boolean
name	Name of the area where the sensor was installed.	varchar
longitude	Geographic coordinate of the sensor's location.	float
latitude	Geographic coordinate of the sensor's location.	float

Table 1: Data dictionary for features in btsites.csv

A basic plot of the sensor locations is shown in Figure 3. The orange points on the map represents sensors which are currently active, and the blue points denotes all those sensors that are either not working or currently under maintenance.

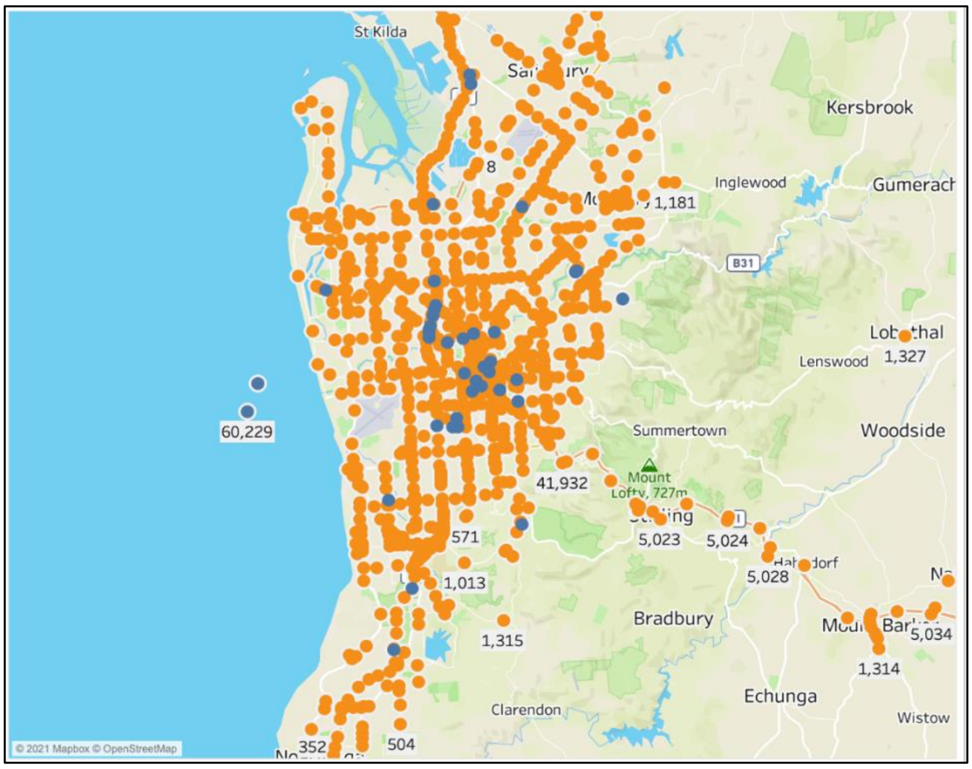


Figure 3: Map of the currently active (orange) and inactive (blue) sensors across Adelaide

A closer look at the distribution of active/inactive sensors can be observed from Figure 4 which shows that there are just over 1100 sensors which are currently active and 50 which are inactive.

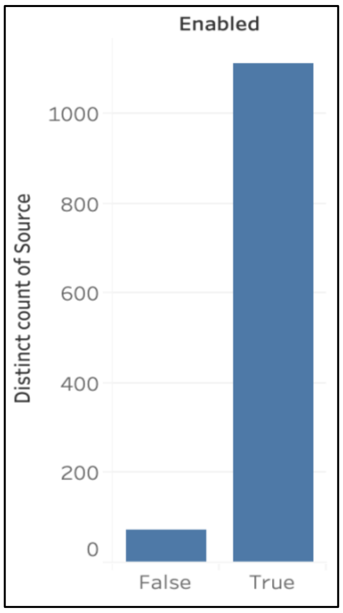


Figure 4: Comparison of active and inactive sensor counts

<u>BTRecords:</u> The data in Table 2 provides a closer look at each driver's activity past the sites.

Variable	***Description***	***Data Type***
siteid	ID of the sensor where the probe was detected.	int
probeid	Unique ID collected from each device.	int
firstseenat	Time when the sensor detected the device.	datetime
logtime	Time when the device left sensor range and record is logged.	datetime

Table 2: Data dictionary for features in btrecords.csv

The initial raw dataset contains records corresponding to individual probe id being recorded by a particular sensor at a specific time. For instance, if a person is travelling from Norwood to CBD and suppose there are 4 sensors installed on the route travelled by the person, there would be 4 separate records in the database all corresponding to the same probe id but with different site ids and time detected. However, it is important to note that all these 4 records can be grouped under a single trip thereby reducing the redundancy in the dataset.

Other common patterns in the log files can also be observed where for instance, the same driver has been traveling from Norwood to CBD for having lunch at a restaurant where they halt for a period of 30 minutes and then drives back home. In this case, the data will be recorded twice by the sensors. Now, if considering the entire travel time of the person as a single trip, the halt of 30 minutes will also be included in the trip duration thereby increasing the scope for fault predictions which also increases the numbers for other statistical metrics such as mean and median travel times, etc. Hence, it is important to split all the round trips into two separate trips or be removed to improve the accuracy of further analysis. As a result, a set of pre-defined rules for grouping the records into trips detailed in the following section were produced to account for such uncommon trip occurrences.

Initial inspection of the monthly data for the sensor 4503 which is located near the Britannia Roundabout was visualized using a time series plot as shown in Figure 5 and it can be observed that the data had a lot of fluctuations which gives an early indication that the predictions might not be accurate. The data also does not look seasonal complicating things even further. As a result, it might be a good idea to inspect the data on a daily or a weekly basis.

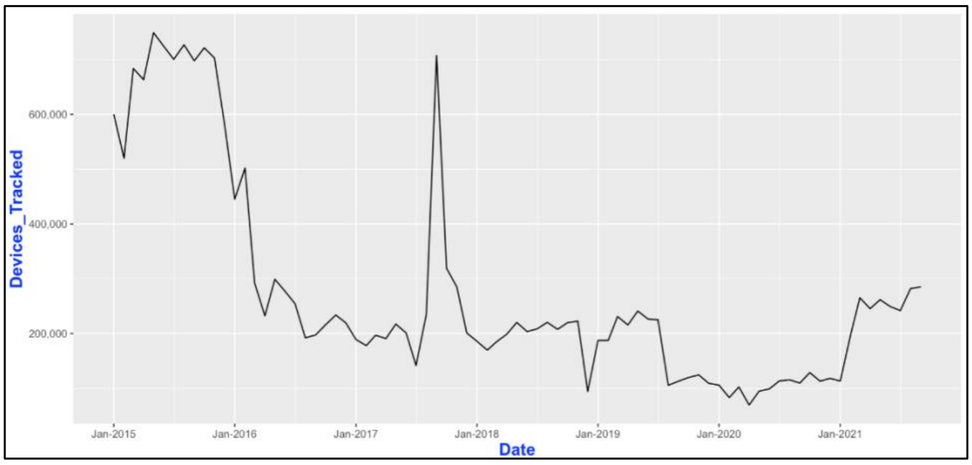


Figure 5: Line graph of the monthly devices tracked over time (Britannia Roundabout)

Figure 6 is a plot of all the probes recorded by the sensor, 4503 daily for a period of 39 weeks from 04 January 2021 to 03 October 2021. A clear seasonal pattern is associated with this data which might be very helpful for making accurate predictions.

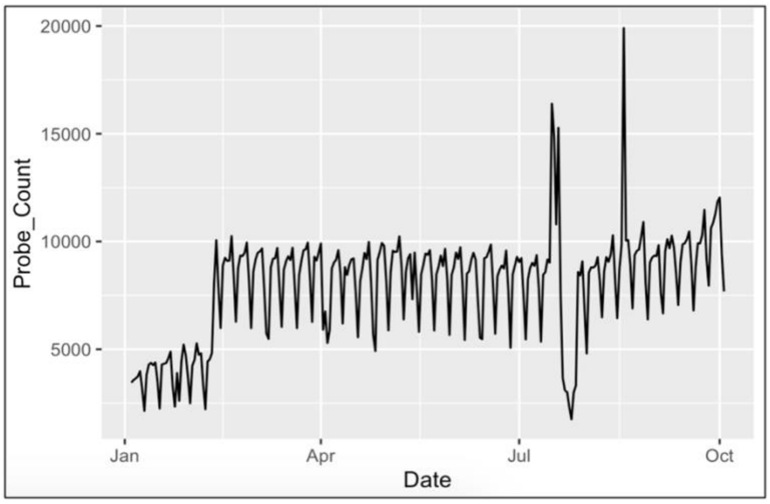


Figure 6: Line plot of the daily probe counts for a period of 39 weeks in 2021 (Britannia Roundabout)

3.3 Data Pre-processing: First Stage Processing of Data

Tools

- Google API and Maps Library in Python [General Route Analysis]
- R & R Shiny [Time Series Analysis and Dashboard]
- Jupyter Notebook [Processing Trips]

To produce deliverables for each of the client's requests, on the advice from the project mentor and after communicating with the clients, the project's objectives were altered to creating several dashboards or tools which may be used in conjunction during analysis. Each of the produced items are discussed in the later section of the report, however Figure 1 shows the three main initial tasks that will be performed requiring the joining and processing of the collected data from the previous initial step.

Procedures

Two different approaches were tried and tested as listed below to find a solution to the current problem of finding the possible routes between an origin and a destination. The data for the first approach comes from a combination of different sources such as the observations recorded by DIT, and those that were extracted from the Google Maps package in Python. While the data used in the second approach is purely obtained from the resources provided by DIT.

Route planner with Google Maps: The first approach involves finding the quickest possible route between an origin and destination using Jupyter Notebook along with assistance from Google Maps and the log data that was collected from the resources provided by DIT. This approach can be used alongside the second approach to compare the travel time and total distance between any two locations to make sure that the log data was transformed properly or report any anomalies in the data. Python was chosen to be the best suitable programming language for this task because of the easy availability of all the required packages. The code for this tool is included in the Task 1 code file: Task1.ipynb. The various steps involved for this approach are briefly explained below:

1. A google cloud account was created and an API key was generated which was necessary for using the Google Maps API in Python.
2. Two drop-down lists for selecting the origin and destination probe ids were created using the sites log data provided by DIT.
3. A datetime picker was created to let the user pick a specific travel time which was a client-side requirement.
4. The dashboard then processes the data and visualises the best possible route and the travel time between the chosen origin and destination.
5. The flexibility of choosing a different route is also made possible by giving an intermediate destination (Source → Intermediate Destinations → Destination).

Route planner with DIT collected data: The second approach involves only the data collected by DIT, with the various steps involved in the pre-processing, transforming, and analysis phases of the approach are as follows listed below. This allows the extraction of trip information from the log data for each driver and allows the identification of routes based on activity:

1. Probe ids that are only seen once in the data are removed as no trip information can be formed from it.

2. Data should then be grouped by probe id and then sort by firstseenat. This will show all the trips that each detected car has made, sorted by the order in which they passed the sites.
3. Then for each consecutive site id that were previously grouped, find the time difference between them. Preferably in seconds so that things are easier to be coded. This creates distributions of travel time for each consecutive site id available.
4. Travel times for a single car between consecutive site id that have been found to be outside -3 or +3 standard deviations are marked as outliers which we then consider as the point at which a trip ends, and another begins.
    - Later, the ending and starting points of trips between different cars are explicitly marked.
    - This also filters unusual behaviours from cars which are staying in one place too long or those that have gone around uncommon routes to reach the next site id.
5. Based on rows which are marked as not being a trip based on the previous step, fill in the gaps between to indicate if it is the start, journey, or end of a trip.
6. Once trips have been identified, they are assigned a grouping based on their origin and destination. Analysis to mark trips as outliers based on travel times outside -3 and +3 standard deviations as in the technique done for consecutive site id is performed.
    - Outlier trips are removed in addition to round trips that have the same origin and destination. This decision was made to simplify processing as it is assumed that enough data would be available when considering the full years' worth to provide adequate summary statistics per route or SA1 zone travel times for normal driving behaviours.
7. Trips routes through site ids are then numbered based on the marking associated with the site id which could be start, journey or end. Likewise, trips are also numbered to differentiate different trips based on its start marking, e.g., a trip is assigned id 1, and passes through 3 different sites which are then given the ids, 1, 2, and 3 according to the order they were traversed.

Note that the decision to identify outliers to be outside ±3 standard deviations is mostly arbitrary, the client may decide later to decrease the range so that more trips are marked as outliers to achieve the best visual results. The code for this analysis is included in the Task 3&4 code file: Task3&4.ipynb.

Forecasting the traffic patterns for the near future: Time series analysis was of interest to be done on a site id basis to view the current and future traffic conditions. New data was required to be sourced from AWS which contains the aggregated probe counts for two sites each day between 04 January 2021 – 03 October 2021 which was available from the Daily_Data.csv file within the Task 2 folder. R was chosen for the analysis as it contained all the essential packages that are required for producing a time series plot; fitting a suitable ARIMA model to the series; plotting an autocorrelation (ACF), and partial autocorrelation (PACF) graphs; and performing forecasting. The Shiny library then provides a clear dashboard layout in which the analysis graphs can be easily displayed to the client. The steps involved are as follows:

1. Extract daily recorded data grouped by Probe IDs from the SQL Workbench and store it as a text file.
2. Load the data into R.
3. Create a function that converts the data into a time series and impute the missing values with the appropriate values according to the data.
4. Create a function for fitting a suitable ARIMA model to the series and forecast the traffic at respective probes for the next 2 weeks.
5. Compare the forecasted values with the actual values and note the results.
6. Make a dynamic user-friendly R Shiny Dashboard that visualises the forecasts and other statistics.

The code for this analysis is available from the Task 2 folder under the name: Daily_Time_Series.R.

Advantages of using Google Maps data:

The data is directly transmitted to the server from a mobile device in the form of bits without the need for collecting it using sensors. So, this eliminates the possibility of extracting and storing inaccurate information, avoids redundancy. It is also highly impossible to predict and impute the missing values in the data if the sensors are either under maintenance or not working properly which can be avoided altogether by using the data collected using this method.

It is also worth noting that the traffic patterns not only in Adelaide but all around the globe have been impacted significantly by the COVID-19 pandemic. This means that the historical data might not be too relevant for forecasting the duration of the current trips anymore. So, all these parameters were taken into consideration and Google already updated their current algorithms to accommodate these changes by prioritising traffic patterns extracted most recently (2 – 4 weeks) and deprioritized those patterns from records prior to that time frame (Lau 2020).

A similar approach was followed in our case for making the predictions where we prioritized all the data that was collected from the beginning of 2021 and eliminated all prior data which not only made logical sense but also improved the performance of the fitted model.

3.4 Data Pre-processing: Second Stage Processing of Data

Tools

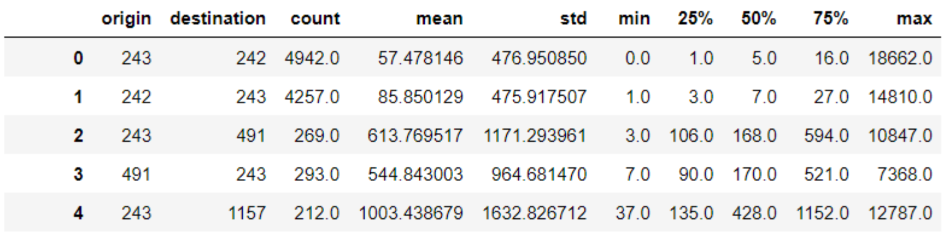
- Tableau [Visualisation of SA1 Zone and Specific Route Traffic and Statistics]
- PowerBI [Visualisation of Specific Route Traffic and Statistics]
- Jupyter Notebook [Feature Creation]

The last stage of data processing mainly involves the creation of features for filtering and calculation of summary statistics in Jupyter Notebooks, allowing for the creation of high-level summaries. The final data is then imported into Tableau and Power BI to produce visualisations for dashboards that the client requires. The use of both Tableau and Power BI is mainly for comparison purposes, and for which we have found that Tableau offers a more intuitive, clear, and simple way to produce and present visualisations, particularly when geographical data is concerned.

Procedures

Once all full trips and the sites through which the cars pass through during the trips are identified, we then calculate the summary statistics (count of trips, mean, std, 25% quantile, median, 75% quantile, min and max of travels times) between sites. As we do not know which pairs of sites the user will be interested in, we therefore calculate the summary statistics for all possible pairs of sites within the dataset. We also consider that for each pair of sites A and B, the travel times for A → B and B → A may differ. With over 1100 active sites in the data, this therefore translates to over 1100^2 or over 1,210,000 possible pairs, each of which we need to calculate the statistics for by all the trips which included both sites, then calculating and summarising the travel time between the sites.

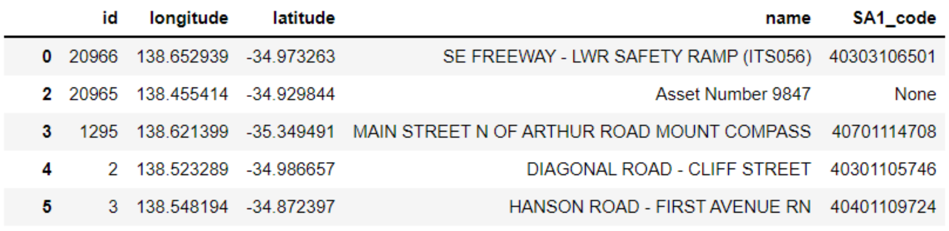
This proved to be an extremely time-consuming process, estimated to require well over 14 hours to process on the team members' local machines. Optimisation was then done by creating a lookup table where each site and all the rows in the data containing that site was registered, dramatically reducing the lookup time for rows containing any given site. Furthermore, for each iteration of our calculation loop, both the travel times for A → B and B → A were calculated, effectively halving the require number of iterations. The overall required processing time was therefore drastically reduced to about 90 minutes and summary statistics data as shown in Figure 7 was produced. The code for this step can be found in Task5_stats_calc.ipynb and the resulting data in travel_stats.csv.



	origin	destination	count	mean	std	min	25%	50%	75%	max
0	243	242	4942.0	57.478146	476.950850	0.0	1.0	5.0	16.0	18662.0
1	242	243	4257.0	85.850129	475.917507	1.0	3.0	7.0	27.0	14810.0
2	243	491	269.0	613.769517	1171.293961	3.0	106.0	168.0	594.0	10847.0
3	491	243	293.0	544.843003	964.681470	7.0	90.0	170.0	521.0	7368.0
4	243	1157	212.0	1003.438679	1632.826712	37.0	135.0	428.0	1152.0	12787.0

Figure 7: Example dataset showing summary statistics between sites

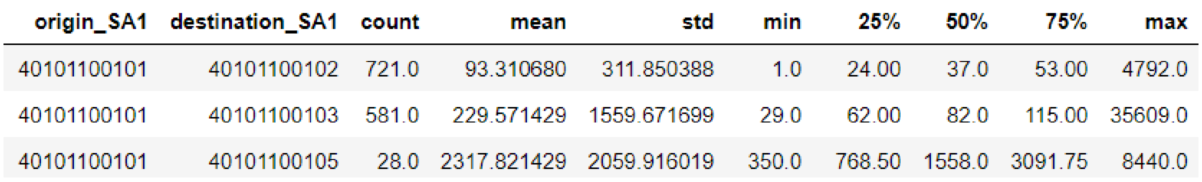
Next, to produce the summary statistics of trips between SA1 zones, we first need to identify the SA1 zone each of the sensor sites in the data is located in. Each SA1 zone is identified by a unique number, with the area and shape of each zone being defined in a ESRI shapefile SA1_2021_AUST_GDA94.shp taken from the Australian Bureau of Statistics website (Australian Bureau of Statistics, 2021). Using a library for processing GIS data, we combine the shapefile with the coordinates of each site to locate the SA1 zone they fall within. The code for this task can be found in Task5_assign_SA1.ipynb and the resulting data in travel_stats.csv (Figure 8).



	id	longitude	latitude	name	SA1_code
0	20966	138.652939	-34.973263	SE FREEWAY - LWR SAFETY RAMP (ITS056)	40303106501
2	20965	138.455414	-34.929844	Asset Number 9847	None
3	1295	138.621399	-35.349491	MAIN STREET N OF ARTHUR ROAD MOUNT COMPASS	40701114708
4	2	138.523289	-34.986657	DIAGONAL ROAD - CLIFF STREET	40301105746
5	3	138.548194	-34.872397	HANSON ROAD - FIRST AVENUE RN	40401109724

Figure 8: Data containing SA1 information for each sensor site

Then, based on the previous table, the summary statistics between sites are aggregated again in Task5_stats_calc.ipynb based on SA1 zones for each origin and destination to produce the following example dataset (Figure 9), which can be found in stats_SA1_agg.csv:



origin_SA1	destination_SA1	count	mean	std	min	25%	50%	75%	max
40101100101	40101100102	721.0	93.310680	311.850388	1.0	24.00	37.0	53.00	4792.0
40101100101	40101100103	581.0	229.571429	1559.671699	29.0	62.00	82.0	115.00	35609.0
40101100101	40101100105	28.0	2317.821429	2059.916019	350.0	768.50	1558.0	3091.75	8440.0

Figure 9: Example dataset showing final aggregated data of trips by SA1 zones

The same procedure to find the summary statistics of each trip in the other analysis is very much the same without the aggregation by SA1 zones. An additional seven features, as listed below, were also created so that filtering and visualisation can be performed:

1. AMPM
   - Indicates when the travel occurred as AM or PM time
2. TravelRange
   - Indicates if the travel occurred during the day (before 9am) midday (9am to 3am) or night (3am and after).
3. Weekday
   - Day of the week in which the trip occurs.
4. Month
   - Month in which the trip occurs.
5. Quarter
   - Quarter of year in which trip occurs.
6. SiteScore
   - Shows how often the site id had been visited for trips with same origin and destination.
7. PopularityOfTrips
   - A count of how often a route is chosen for a trip. The sum of values should equal the count of how often a particular trip (with same origin and destination) occurs.

In practice for the visualisation of individual routes and filter variables it was required that the data be further divided so that computation times are reduced with aggregation for route visualisation and so that each row shows a unique trip for filter variable visuals. The aggregation resulted in a reduction from 816,919 to 360,413 trips remining in the data where trips from the same origin and destination with the same route are removed.

4. Dashboards

This section of the report shows the working and construction of each of the five deliverables and how the end user can navigate their controls for effective use. Please refer to the list below to find each deliverable's title in the report and the corresponding task and files associated in the DDTI1 PROJECT FILES folder.

- Task 1: Finding General Best Routes with Google API

- Task 2: Time Series Dashboard
- Task 3: Visualisation of Routes in PowerBI
- Task 4: Visualisation of Routes in Tableau
- Task 5: Summary Statistics by Sites and SA1 Zones in Tableau

4.1 Finding General Best Routes with Google API

Initially, two drop-down lists to select the source and destination probes were created using the data from the btsites.csv file. The coordinates of these selected probes were then extracted and stored inside two separate variables which were used in the later parts of the analysis.

The option of selecting a date and time for the trip is also provided using a drop-down functionality which is once again retrieved and stored in another variable. This date time field captures the time from the current time zone of the machine. However, google maps uses the UTC time format and hence the above stored datetime field must be modified to accept this change. Another challenge associated with this problem is the conversion of datetime object from UTC format to UNIX format as google maps uses the difference in time interval in seconds between the selected time and datetime (1970, 1, 1).

The distance between any two selected locations in Kms, normal time interval and peak-hour time interval in minutes can be extracted as shown below (Figure 10):

```
The distance between points A & B is:  10.8 km
The normal time interval is:  15 mins
The estimated time of arrival during peak hours is:  21 mins
```

Figure 10: Table showing key metrics such as distance, ETA (normal & peak-time) between A-B

It is also possible to extract all the coordinate changes between any 2 given locations using the Google Maps package in Python i.e., all the coordinates between points A and B were extracted and visualized using red dot points in the image (Figure 11).

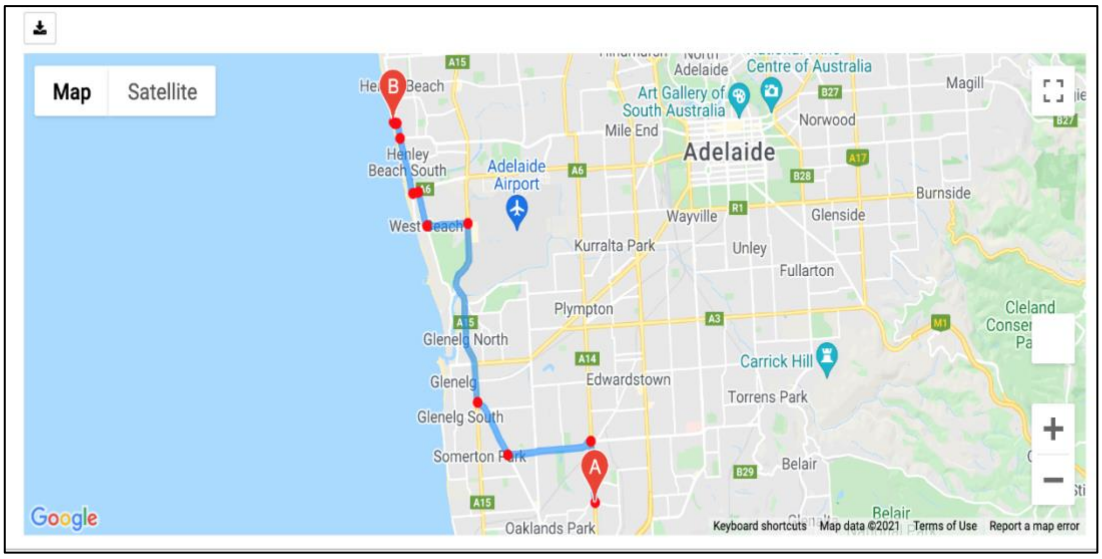


Figure 11: Red dots corresponding to coordinate changes as detected by Google Maps

Next, a distance matrix was created (in Kms) between these coordinates and all the coordinates of the probes using the haversine formula which takes the Earth's shape into consideration unlike the normal Euclidean distance and hence is more effective in calculating the geographical distance between any two locations.

Based on the data from this matrix, all the probes that were within a radius of 0.5 Kms to at least one of the Google Maps coordinates were then selected and plotted in green as shown below. The plot also includes a traffic layer to forecast the traffic patterns of different road networks which helps the user to plan their route accordingly. Routes displayed in green are the most preferable, while those in red are the least preferable because of congestion (Figure 12).

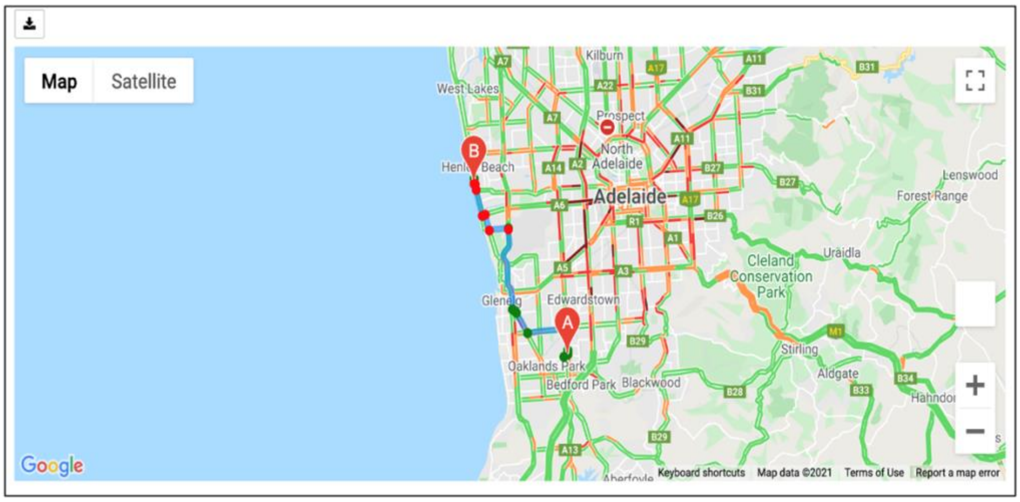


Figure 12: Final map showing the quickest possible route for the chosen date and time

4.2 Time Series Dashboard

The following R Shiny dashboard is interactive and allows the user to select a site id from a drop-down list. The algorithm then extracts all the data related to the selected site id and performs various calculations in the background such as imputing missing values if any present, selecting and fitting the best possible ARIMA model to the series which generates a model that is as close as possible to a perfect-white noise model, and make predictions for the upcoming two weeks which are then compared to the actual values.

The daily extracted data for sensors located near two sites namely the University of Adelaide and Henley Beach Road was extracted for demonstration purposes which is available from the Daily_Data.txt file. (Figure 13) is a basic line graph of the data corresponding to one of the above-mentioned site ids. The model then selects the data and fits an appropriate ARIMA model in the background and makes forecasts (Figure 14).

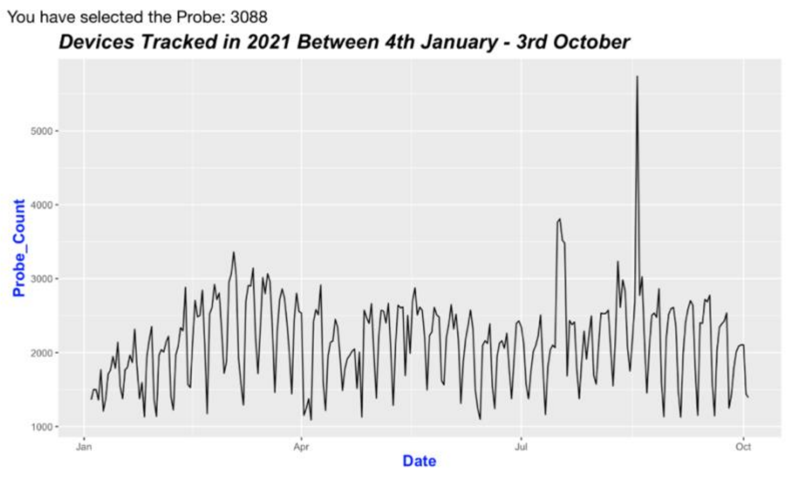


Figure 13: Line plot of the daily probes recorded at site id 3088 from 4th Jan – 3rd Oct 2021

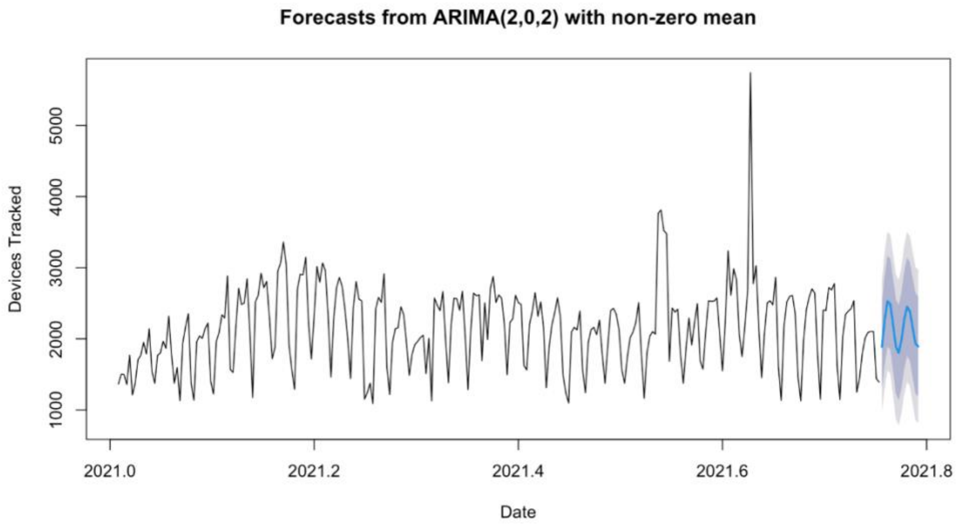
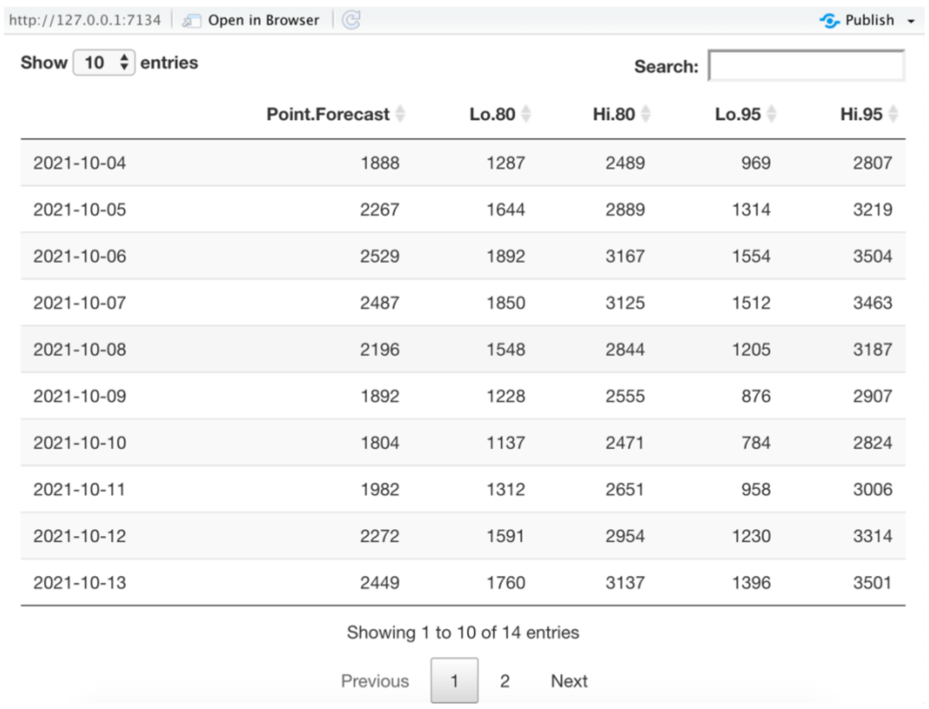


Figure 14: Forecasting two weeks ahead using the fitted time series model

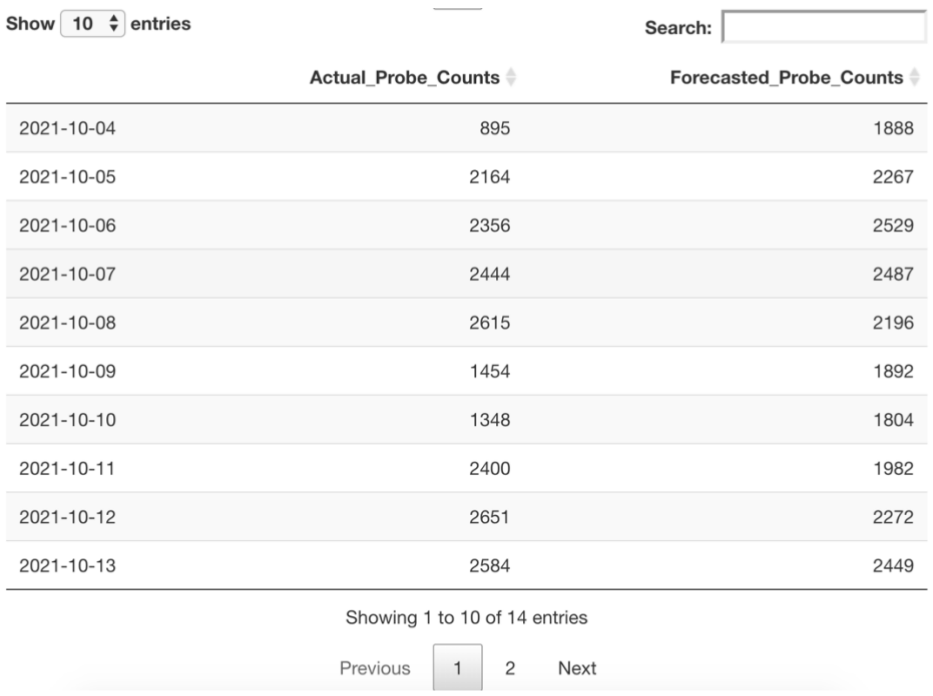
The 80% and 95% confidence intervals along with the point forecasts (Figure 15) are then displayed. Finally, a comparison table is provided between the actual numbers recorded against the sensor and the forecasted values to assess the fit of the model on the series (Figure 16).



	Point.Forecast	Lo.80	Hi.80	Lo.95	Hi.95
2021-10-04	1888	1287	2489	969	2807
2021-10-05	2267	1644	2889	1314	3219
2021-10-06	2529	1892	3167	1554	3504
2021-10-07	2487	1850	3125	1512	3463
2021-10-08	2196	1548	2844	1205	3187
2021-10-09	1892	1228	2555	876	2907
2021-10-10	1804	1137	2471	784	2824
2021-10-11	1982	1312	2651	958	3006
2021-10-12	2272	1591	2954	1230	3314
2021-10-13	2449	1760	3137	1396	3501

Showing 1 to 10 of 14 entries

Figure 15: 80% & 95% confidence intervals along with the point forecasts



	Actual_Probe_Counts	Forecasted_Probe_Counts
2021-10-04	895	1888
2021-10-05	2164	2267
2021-10-06	2356	2529
2021-10-07	2444	2487
2021-10-08	2615	2196
2021-10-09	1454	1892
2021-10-10	1348	1804
2021-10-11	2400	1982
2021-10-12	2651	2272
2021-10-13	2584	2449

Showing 1 to 10 of 14 entries

Figure 16: Comparing the forecasts with the actual values

Although minor modifications can be made to the built model, the results were found to be near the actual values and hence the model can be accepted, and the modifications can be accommodated in future updates. Finally, as the data was found to be valid, the concept can be extended to other site ids as well.

4.3 Visualisation of Routes in PowerBI

The same datasets had been used in both PowerBI and tableau to compare the best suitable tool for the project and below is the explanation that shows how it is been used to build the visualization.

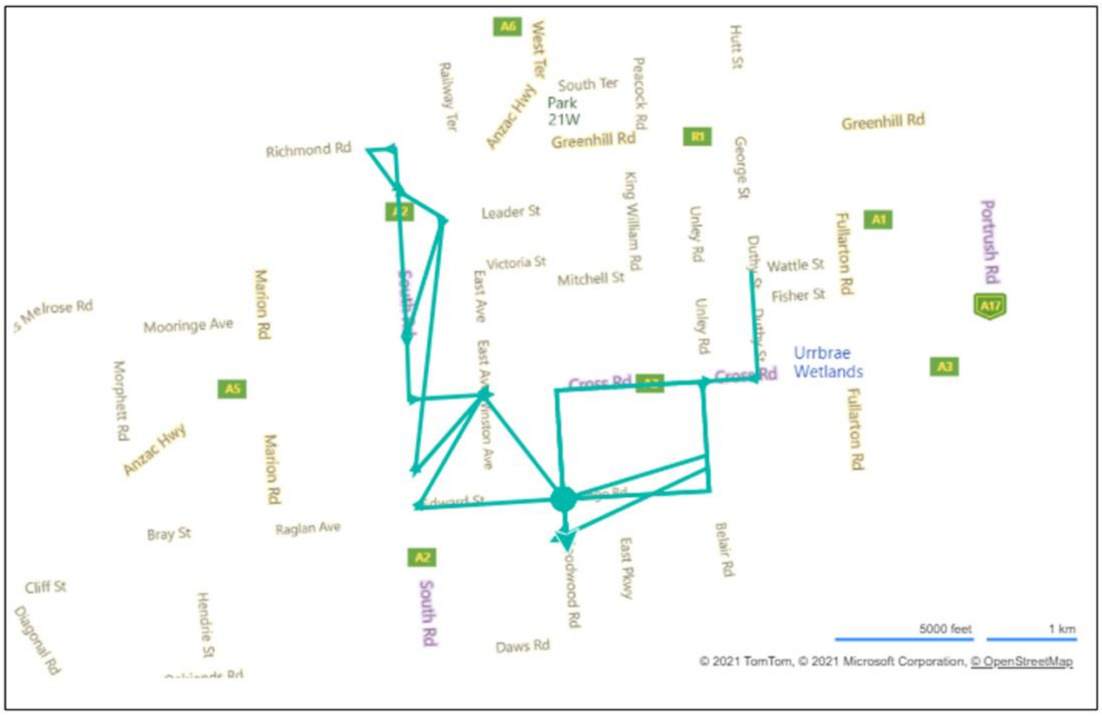
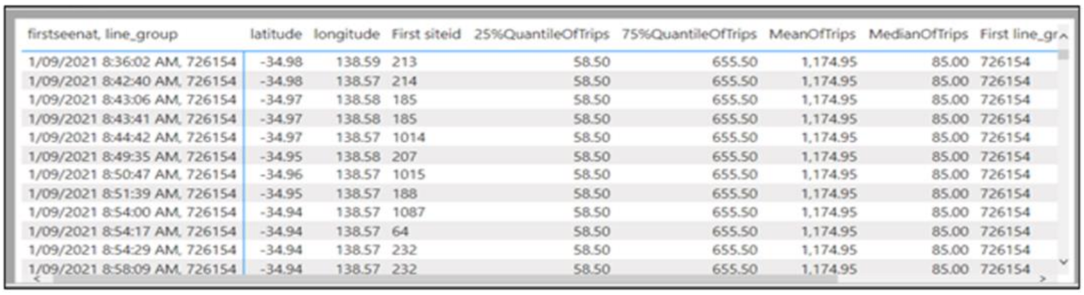


Figure 17: Dashboard showing different routes to get from an origin to destination in Power BI

The data that was used for the above visualisation (Figure 17) was the aggregated trips based on unique routes data (highLevelData.csv). This data was selected to show the routes and the different direction the vehicle had travelled before reaching the destination. The end user is given options to select the origin, destination, time of travel, travel range, week, month, and quarter of travel, all of which are in drop down menus where the user can select the desired options. Once these are selected the prototype displays the route and shows the different site id it had visited before reaching the destination. In addition to the statistical summary view of the trip. The below is the output which shows the table view of the trip's details (Figure 18-19).



firstseenat, line_group	latitude	longitude	First siteid	25%QuantileOfTrips	75%QuantileOfTrips	MeanOfTrips	MedianOfTrips	First line_gr
1/09/2021 8:36:02 AM, 726154	-34.98	138.59	213	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:42:40 AM, 726154	-34.98	138.57	214	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:43:06 AM, 726154	-34.97	138.58	185	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:43:41 AM, 726154	-34.97	138.58	185	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:44:42 AM, 726154	-34.97	138.57	1014	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:49:35 AM, 726154	-34.95	138.58	207	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:50:47 AM, 726154	-34.96	138.57	1015	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:51:39 AM, 726154	-34.95	138.57	188	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:54:00 AM, 726154	-34.94	138.57	1087	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:54:17 AM, 726154	-34.94	138.57	64	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:54:29 AM, 726154	-34.94	138.57	232	58.50	655.50	1,174.95	85.00	726154
1/09/2021 8:58:09 AM, 726154	-34.94	138.57	232	58.50	655.50	1,174.95	85.00	726154

Figure 18: Summary statistics table for the trips in Power BI

The above visualisations provide a clean-cut view showing the different quantile of trips and the mean and median time of the trips that had taken place. Not that these are for the single trips and not for the aggregated data. The aggregated data by unique trips (fullDataWithFullTripsBeforeAggregation2.csv), is used for the below prototype. The end user can select the origin and destination from the drop down and it will show the popularity of the trip and the different time ranges in which this route has been chosen, providing an understanding of traffic volume in the selected routes.

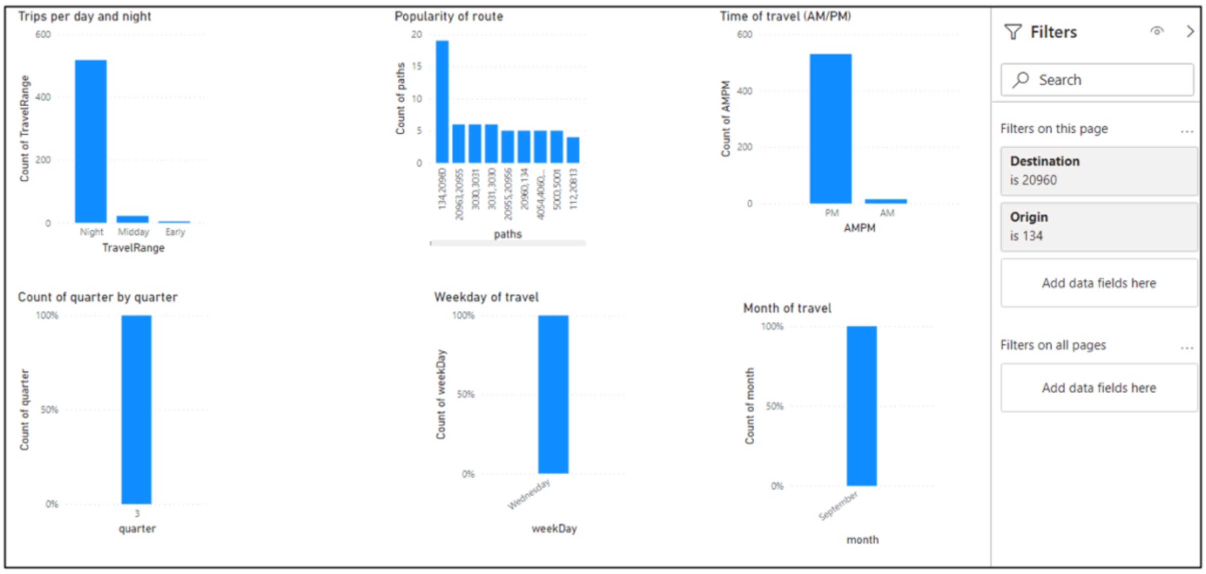


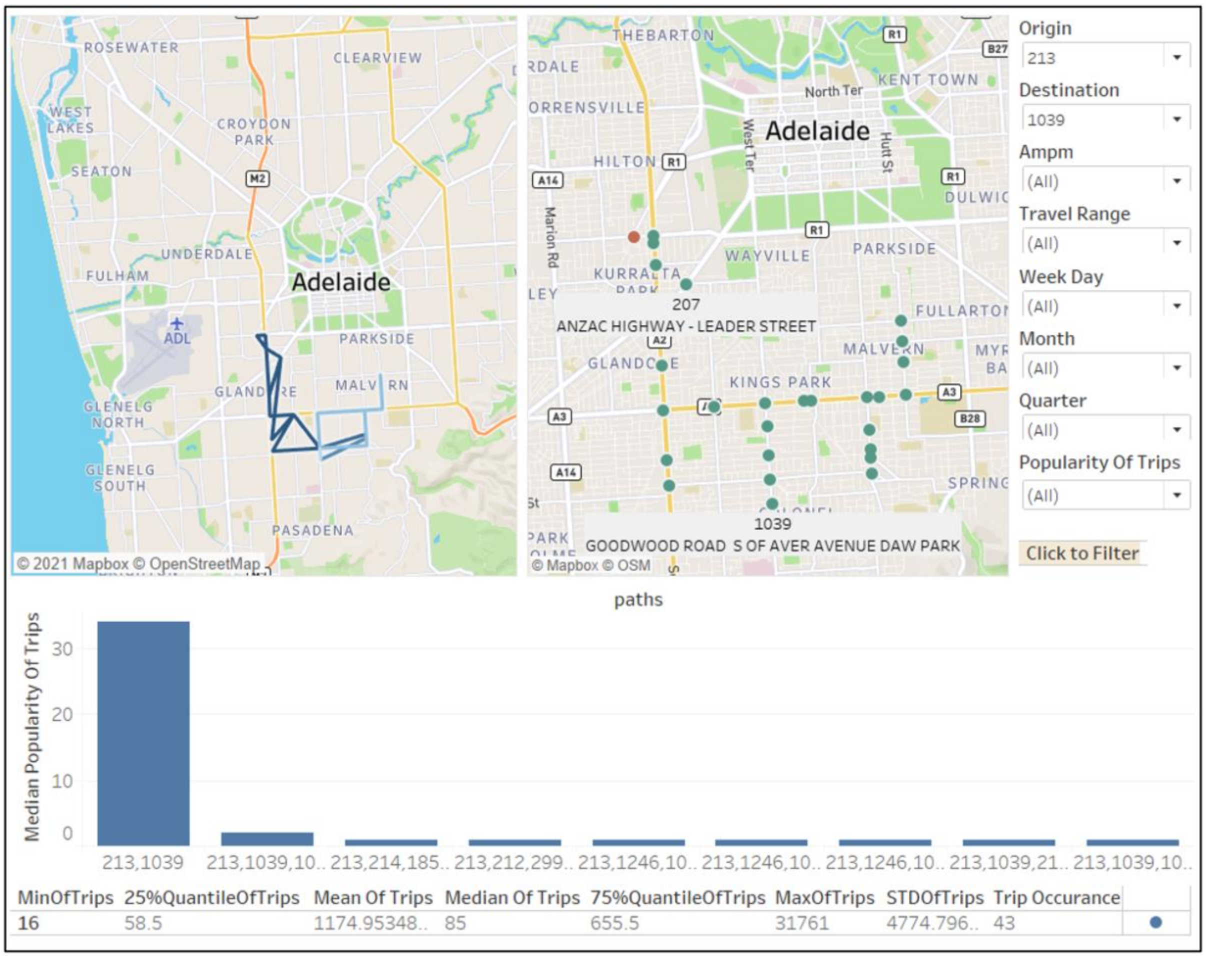
Figure 19: Bar charts for each of the filter variables in Power BI

4.4 Visualisation of Routes in Tableau

The dashboards show the unique routes and summary statistics that exist for an origin and destination (Figure 20). Filters are also available to subset the data. Additional visuals also show the number of trips that exist for each filter. This allows the user to observe specific route choices by drivers from all possible origin and destination. To better understand the travel times and different choices in routes that are chosen. Visuals for each filter also adds depth to the understanding in which traffic occurs most often (Figures 21-22).

Often the user only needs to select filters for origin, destination and then popularity of trips. As an initial refinement with other filters may return no results given that only a day worth of data is used to produce the dashboard and the data may not exist for all filtering options. Popularity Of Trips should be the last filter that users choose, as in some trips users may experience routes with various popularity. To interpret this information, users may be interested in the most common or reasonable trips which have the highest popularity, while other trips that have a relatively lower count show trip that are less common and often may be described by unique routes that some only one driver has taken. Mentioned in the first stage of data processing, the client may change the outlier filtering range to be smaller than ±3 std to encompass only trips with high popularity and remove those with only 1-2 popularity.

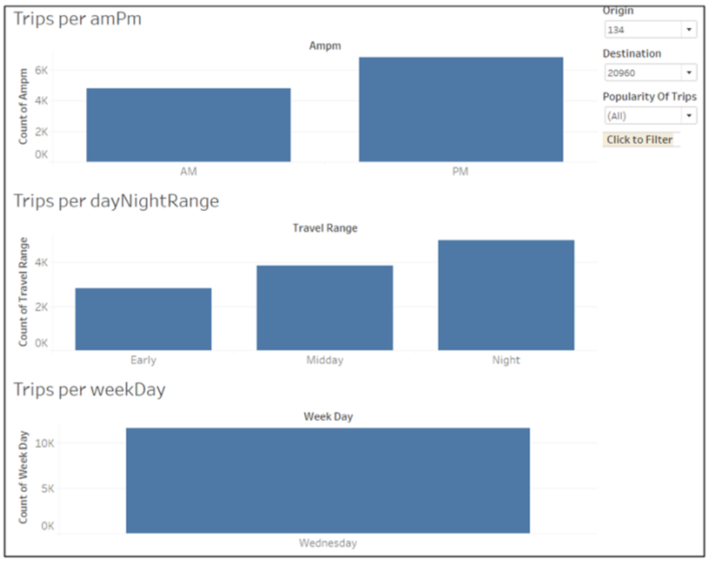
As mentioned in the previous section the data used to create the visuals in Figures 20 is based on highLevelData.csv while Figures 21-22 use fullDataWithFullTripsBeforeAggregation2.csv. Note that end users may ignore the description of "Median" in the column "Median Popularity of Trips" visual as it was a result of an artefact of how the data is aggregated. With the variable Trip Occurrence describing the total number of trips that exist from an origin to destination, the count of all different routes and their popularity multiplier should add up to Trip Occurrence.



Origin: 213
Destination: 1039
Ampm: (All)
Travel Range: (All)
Week Day: (All)
Month: (All)
Quarter: (All)
Popularity Of Trips: (All)
Click to Filter

MinOfTrips	25%QuantileOfTrips	Mean Of Trips	Median Of Trips	75%QuantileOfTrips	MaxOfTrips	STDOfTrips	Trip Occurance	
16	58.5	1174.95348..	85	655.5	31761	4774.796..	43	●

Figure 20: Dashboard showing different routes to get from an origin to destination in Tableau



Origin: 134
Destination: 20960
Popularity Of Trips: (All)
Click to Filter

Figure 21: Bar chart for week-based filters in Tableau

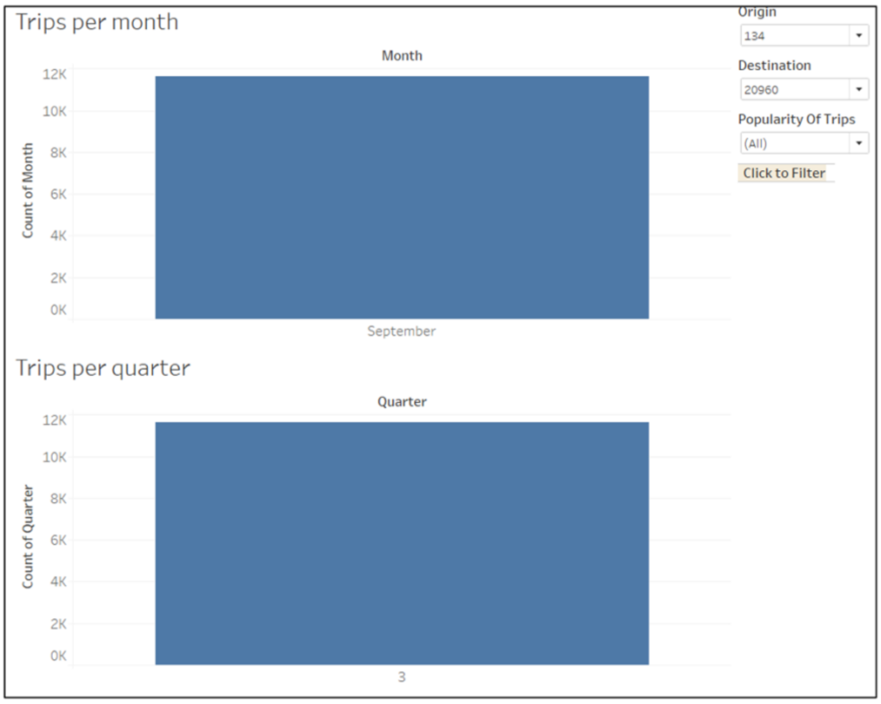
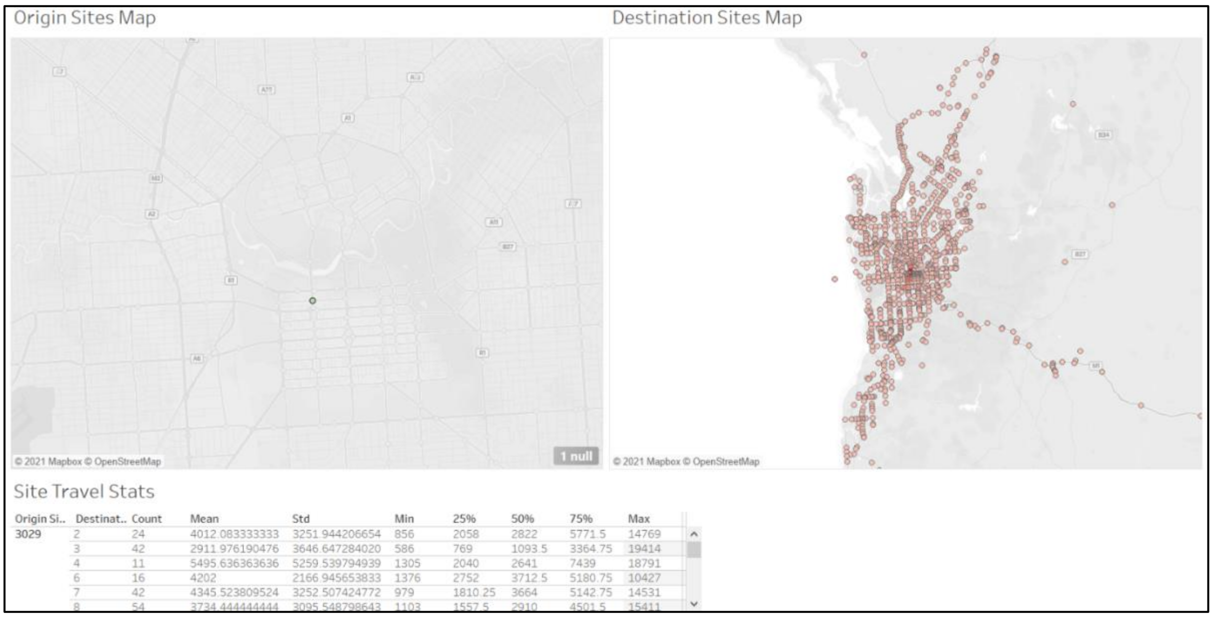


Figure 22: Bar chart for year-based filters in Tableau

4.5 Summary Statistics by Sites and SA1 Zones in Tableau

These Tableau dashboards include two different views: one for visualisation of travel statistics between different sites and the other between different SA1 zones, both having a similar layout and functionality.

The first dashboard for site-to-site travel statistics (Figure 23) includes two interactive maps containing all sensor sites available in the network. Users can select any number of origin sites from the map on the left, as well as any number of destination sites from the map on the right, with the travel statistics such as count of trips, mean, standard deviation, minimum, 25%, 50% and 75% quartiles, minimum and maximum travel times between all selected origin and destination sites being displayed in the table below. This dashboard is built using the file travel_stats.csv, and the maps are filtered according to the availability of the data within it i.e., when selecting an origin site, only destination sites which have had at least 1 trip originating from that origin site will be shown on the map, and vice-versa.

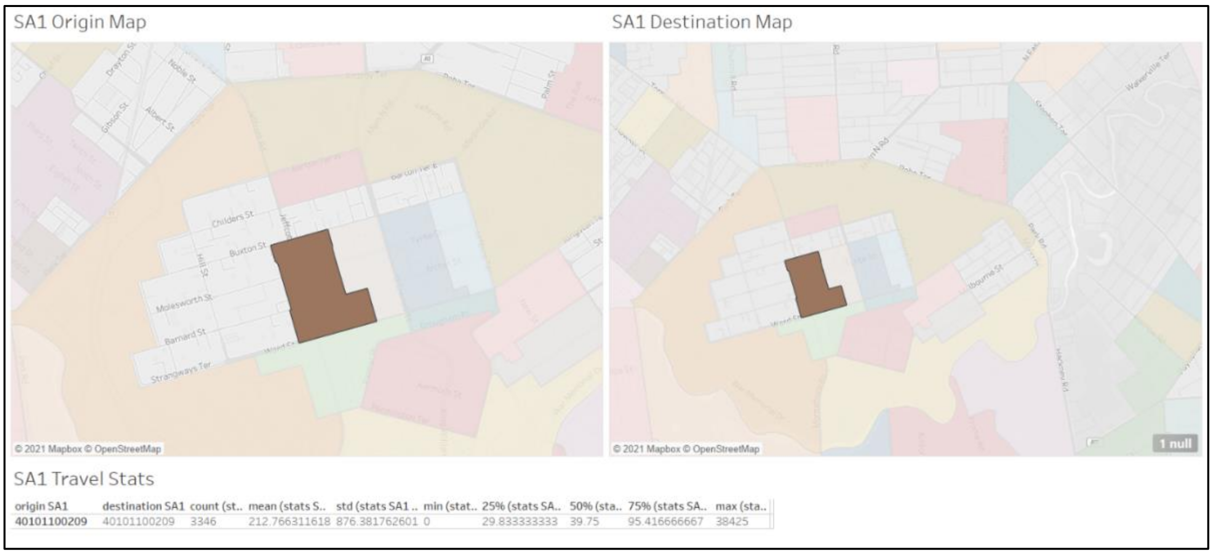


Origin Si..	Destinat..	Count	Mean	Std	Min	25%	50%	75%	Max
3029	2	24	4012.083333333	3251.944206654	856	2058	2822	5771.5	14769
	3	42	2911.976190476	3646.647284020	586	769	1093.5	3364.75	19414
	4	11	5495.636363636	5259.539794939	1305	2040	2641	7439	18791
	6	16	4202	2166.945653833	1376	2752	3712.5	5180.75	10427
	7	42	4345.523809524	3252.507424772	979	1810.25	3664	5142.75	14531
	8	54	3734.444444444	3095.548798643	1103	1557.5	2910	4501.5	15411

Figure 23: Dashboard for travel summary statistics between sites in Tableau

The second dashboard for travel statistics between SA1 zones (Figure 24) offers the same functionality as the first dashboard, such as the ability to select multiple zones and automatic filtering for each map. Once the user has selected the SA1 zones they are interested, the aggregated travel statistics between the relevant zones will also be displayed in the table underneath. This dashboard was built using data from stats_SA1_agg.csv as well as the shapefile SA1_2021_AUST_GDA94.shp for drawing the SA1 zones.

These two dashboards therefore enable users to quickly view the travel time summary statistics between any sites or SA1 zones they are interested, which can be useful in establishing what expected travel times are or if there are any disruptions if travel time are longer than what should be expected. Combined with Tableau's export data feature, the summary statistics between specific sites or SA1 zones can be exported to a file for use in reporting.



SA1 Travel Stats

origin SA1	destination SA1	count (st..	mean (stats S..	std (stats SA1 ..	min (stat..	25% (stats SA..	50% (sta..	75% (stats SA..	max (sta..
40101100209	40101100209	3346	212.766311618	876.381762601	0	29.833333333	39.75	95.416666667	38425

Figure 24: Dashboard for travel summary statistics between SA1 zones in Tableau

5. Future work and Issues

The following dot points lists the major issues that had been encountered throughout the project detailing additional problems that were encountered but had not been resolved or investigated in full. These issues may need to be addressed in future iterations of the project to improve results.

- There is an issue with full trips where a car id is continuously redetected over a site within a short period of time. This time frame is too short to be detected as an anomaly and thus these outlier trips need to be removed in the future.
- In some cases, a car is not detected when it goes through a site id which causes visuals on the map to not show a consistent path. This may be caused by missing site ids that are later added or due to another issue.
- Tableau sometimes has an issue of adding a site id marker onto the map, in such cases when filtering is done the route displayed will look incomplete and not display all the site ids for the route.
- A day's worth of log data is required to be downloaded and processed offline, this results in code or dashboards that may not be optimized or suitable for data streams or a larger volume of data due to computational times. Code will need to be parallelised and improved for efficiency in the future. Alternatively, only data relevant to selected sites or SA1 zones can be queried and processed during runtime, minimising the required data and processing times.
- When handling large amounts of data in Power BI, it is possible for operations to take a long time to execute, and even possibly stopping or freezing unexpectedly.
- The monthly aggregated data for all the sites was also not found to be consistent over time making things extremely hard for making the predictions.

6. Conclusion

In this project the team extracted a day's worth of log file data from AWS to achieve four main client requests and individual team member investigations. These tasks pertain to producing visualisation of statistical trip information between SA1 zones, specific route choices based on popularity and general best course, a comparison of PowerBI and Tableau functionality, and forecasting future traffic.

Much data pre-processing work was necessary given that the data collected by the client through the AddInsight network sensors sites across the city included some redundancies for parts of the data as well as missing values whenever the sensors were under maintenance or non-functional which created a considerable number of errors in the predictive models and dashboards.

Five separate dashboards were produced to provide analysis tools for each of the tasks using R (R Shiny), Tableau, PowerBI, and Python (Jupiter Notebooks, Google Maps API). These tools can be used together in conjunction to provide both high and low-level understanding of trips and driving behaviour over an area or specific points.

From the team's experiences, Tableau was found to include better functionality when visualising geographical data, particularly when displaying aggregated data based on SA1 zones as well as displaying different specific route choices and creating filtering options.

The main challenge encountered during the project was working with a very large quantity of data, requiring optimisation of the data transformation code to reduce execution time to an acceptable level. Improvements for each analysis tool however had to be left as future work, due to short comings of deadlines and computational capabilities, such as parallelisation of the code.

References:

Australian Bureau of Statistics, 2021. Digital Boundary files, viewed 29 October 2021, <https://www.abs.gov.au/statistics/standards/australian-statistical-geography-standard-asgs-edition-3/jul2021-jun2026/access-and-downloads/digital-boundary-files>.

Bainbridge, L 2016, 'How to create a travel time data visualization map with Leaflet', TravelTime, viewed 26 October 2021, <https://traveltime.com/blog/travel-time-data-visualization-map-leaflet>.

Barth, D 2009, 'The bright side of sitting in traffic: Crowdsourcing road congestion data', Google, 25 August, viewed 23 October 2021, <https://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html>.

Bhardwaj, A 2020, Calculating distance between two geo-locations in Python, '*Towards Data Science*', 13 June, viewed 03 October 2021, <https://towardsdatascience.com/calculating-distance-between-two-geolocations-in-python-26ad3afe287b>.

Bugnion, P 2019, 'gmaps Documentation', Build Media, 13 January, viewed 15 September 2021, <https://buildmedia.readthedocs.org/media/pdf/jupyter-gmaps/latest/jupyter-gmaps.pdf>.

Dauletbak, D, and Woo, J, 2019, 'Traffic Data Analysis and Prediction using Big Data', viewed 29 October 2021, <https://www.calstatela.edu/sites/default/files/groups/High%20Performance%20Information%20Computing%20Center%20(HiPIC)/papers/trafficapic-ist2019.pdf>.

Derrow-Pinion, A, She, J, Wong, D, Lange, O, Hester, T, Perez, L, Nunkesser, M, Lee, S, Guo, X, Wiltshire, B, Battaglia, PW, Gupta, V, Li, A, Xu, Z, Gonzalez, AS, Li, Y, Veličković, P 2021, 'ETA Prediction with Graph Neural Networks in Google Maps', arxiv.org, 25 August, viewed 24 October 2021, <https://arxiv.org/pdf/2108.11482.pdf>.

Prabhakaran, S 2021, 'ARIMA model – complete guide to time series forecasting in Python', Machine Learning Plus, 22 August, viewed 15 October 2021, <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>.

Saslow, E 2018, 'Getting started with Google Maps in Python', Medium, 03 August, viewed 23 September 2021, <https://medium.com/future-vision/google-maps-in-python-part-2-393f96196eaf>.

'*The basic parts of a Shiny app*' 2017, R Studio, 28 June, viewed 14 October 2021, <https://shiny.rstudio.com/articles/basics.html>.

Xie, Y 2017, 'How to use data tables in a Shiny app', R Studio, 28 June, viewed 21 October 2021, <https://shiny.rstudio.com/articles/datatables.html>.