How Late Is Too Late For a Bus?

Final Report

COMP5703

Information Technology Capstone Project

Group CP 13-2

Group Members

Peng, Qinqing	**(470476783)**
Wang, Xuying	**(480346863)**
Wang, Zijun	**(470162738)**
Xu, Yuansi	**(460349011)**
Zhang, Kun	**(420033455)**

ABSTRACT

This group project is designed to address service reliability issues of public buses in Sydney. Furthermore, this project will analyse which factors could have significant impact on the buses' on-time performance. The main deliverable of the project was a website with analytical insights provided. There are three types of clients in this project: passengers, bus operators and government agencies. For passengers, this project could help them to plan their future trips efficiently and wisely, while knowing if there is a stop more likely to encounter bus delays. For bus operators, this project could provide useful information on how their services perform and whether they can optimize their current bus schedules. For government agencies, this project would assist them to make predictions, with which they can plan new bus route to improve public buses' performance. The overall approach for this project is to assess the performance of buses' while introducing visual aids based on real-time data. This project would let all involved clients to better understand the public bus transportation in Sydney and further assist them to make the public transportation system better.

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

In December 2013, Transport for NSW released Sydney's Bus Future which was aimed to restructure the current bus service system into a simpler, faster and better one (NSW, 2013). Yet, in 2016 there were thousands of complaints about 50 inner west bus routes (O'Rourke, 2016), although Guardian Australia stated that when considering the number of passengers, the inner west is not the worst (Evershed, 2017). No matter whether the inner west bus routes are the worst routes or not, it has been almost 5 years until the release of Sydney's Bus Future. The bus passengers in Sydney did not make fewer complaints through this period. Instead, people become much more unsatisfied with the bus services in Sydney, especially the usual delays.

This project aims to analyse the bus services in Sydney by comparing scheduled bus timetable and real-time bus data. The analysis results will be presented on a website for the clients to access. Meanwhile, a predictive model will be applied to predict a new route's delay performance based on the existing dataset. The clients could be passengers, bus operators and government agencies. Based on this analysis, passengers will be able to search any bus route in Sydney to have a rough understanding of its trip times, delay rate and other information; bus operators can adjust schedules of existing bus route services; and government agencies may refer to the analysed results and decide whether to adjust the bus routes distribution in the future.

Therefore, the motivation of this project is to provide a useful analysis and construct a predictive model to passengers, bus operators and government agencies to improve the bus services in Sydney, which will finally increase the people's level of satisfaction with the bus services, especially on-time performance.

2. LITERATURE REVIEW

In this chapter, literature related to this project is reviewed, which focuses on the importance on analysis of public transport reliability, analysis measurements and analysis methods of public transportation.

2.1 Why analysis public transport reliability is important?

Unlike the trains, light rails and ferries, buses do not have undisturbed routes since buses have to share the lanes with other vehicles while trains and light rails have their own tracks. This means that compared to other public transports bus is more likely to be unreliable. A report from the Commission of the European Communities stated that encouraging people use public transport instead of private vehicles is vital to keep the city congestion mitigated (European Communities, 2009), but unreliable public transports undermine this positive strategy.

In order to achieve this goal, Sydney's Bus Future made a customer research showed that there were four main factors could encourage people to shift from personal cars to buses, which were convenience, frequency and reliability, connectivity and comfort (NSW, 2013).

2.2 Review on analysis measurements of public transportation

The definition of public transport reliability is varied with respect to different groups or studies. Some studies connect reliability to timetable schedules (Bates et al., 2001), while others associate reliability with consistency in journey times (Polus, 1978). A measure of the probability that a bus trip could be consistent in the expected factors includes cost, comfort, travel time, etc. (AP. Sorratini et al., 2008). To be stated more detailed, the definition could be represented on trip time, waiting time, passenger load, safety, vehicle quality, amenities and information (Ceder 2016).

However, it is difficult to analyse all these factors of public transport. Firew studied the reliability of bus line 550 in the Helsinki Capital where he used five different measures which were on-time performance, headway adherence, vehicle trip-time variability, passenger wait time and passenger travel time (Firew, 2016). However, on-time performance and headway regularity are schedule-based and there is no universal benchmark in defining how many seconds or minutes late a bus is considered delayed or the difference between frequent and infrequent route services (Ma* et al., 2013). Nakanishi recommended that on-time performance could be useful for both passengers and operators to understand the service performance of a typical route (Nakanishi, 1997).

On the other hand, there are many studies giving equations to indicate wait time, but all these equations are based on some assumptions, such as random passenger arrivals, regular vehicles arrivals and passenger catching the first bus (Ma* et al., 2013). For the conditions of irregular bus arrivals or routine passenger arrivals, there is no typical detailed criterion. Therefore, in

this project, to give the clients a convincing analysis, bus on-time performance and trip-time variability will be analysed.

2.3 Review on analysis method for public transportation

Since the pattern of work days and weekends and passengers demand varies, the public transportation service frequency is different throughout a week. Hence, it is necessary to do a cluster analysis to group the similar times of days as well as days of week together for the analysis described (Ma et al., 2013). Broadly, Skogen applied machine learning techniques to predict bus travel times, including k-nearest neighbours, artificial neural network and support vector regression, which could be also applied on the bus delay information (Skogen, 2014). His results showed that these three models had their own strength in different areas. Therefore, in this project the dataset will be analysed first to determine which models could be applied for predictive purposes.

2.4 MVC and MVA Architectural Pattern

An architectural pattern commonly adopted for developing web application is MVC (model-view-controller) pattern, which separates presentation layer from internal information layers such as business logic layer, persistence layer and database layer (Leff & Rayfield, 2001). Specifically, model directly manages persistent data and provides business logic to the data, view presents information of internal layers and controller defines the application's behaviours. These three layers interact among each other directly. MVC allows simultaneous development and reuse of code because its components are loosely coupled (Davis, 2008). MVA (model-view-adapter) is a variation of MVC pattern. Though both traditional MVC and MVA attempt to solve the problem of separating data handling and user interface events, MVA arranges the adapter to accept all the information and interaction from model and view which means view and model would not communicate and connect with each other directly and the adapter would act as a communication hub (Zamudio, 2012). MVA architecture realizes easier evaluation and debugging of application.

3. PROJECT PROBLEMS

By studying issues related to the public and NSW traffic managers' needs, we have identified project questions and established project aims. In addition, to successfully achieve the aims and solve these questions on schedule, a clearly project scope is stated in this section.

3.1 Project Questions

Nowadays, public transport has played an essential role in people's lives. However, bus delay is a significant issue that could influence public transport reliability. The reason is that comparing with other public transport, there are more factors could influence bus traffic such as weather, road maintenance and traffic accident. We have identified two critical issues to address as follows:

1. Understanding current bus traffic operational status. Bus traffic system is huge and complex, making it difficult to identify factors that related with the current bus traffic status. In addition, since there is no available analysis tool for bus traffic data, it is difficult to present the traffic status in an understandable manner.

2. Managing a better bus traffic system for the future. How to reduce bus delay rate in the future is an important topic for TfNSW. The essentiality and impact of bus traffic system adjustment should be justified by a scientific data analysis method.

3.2 Project Aims & Scope

After having analysed the questions, aims need to be achieved are listed below at three levels:

- Data level: acquiring and managing the huge volume of traffic data in a logical manner.
- Analysis level: building models to show current bus traffic status and predict future traffic changes and trends.
- Visualisation level: Present analysis results in an easily-understood format.

Our project consists of four parts: data warehousing and wrangling, data analysis, website design and a final analysis report. The scope of each part is provided below:

1. Build and design a relational database to store real-time transport data and support data analysis. The analysis results should also be kept in the database and can be used by the web server.

2. Analyse current bus traffic system status, including comparing the percentage of delay for each route, analysing which stops are more likely to have delays, and finding which trips are more likely to be delayed for a specific route.

3. Construct mathematical models according to independent variables to predict dependent variables.

4. Design an interactive web application to present key results.

5. Some analysis findings which may not be suitable to display on the website will be incorporated in the analysis report.

4. METHODOLOGIES

4.1 Software Development Methodology

The requirements of the Capstone Project are somewhat broad, giving students the opportunity to explore all possibilities and realize their ideas. As such, prior to the development stage of the project, feasibility of the ideas was unknown and system requirements may be vague.

The project team is made up with students from Master of IT and Master of Data science. Members of the team have exposures to and experiences in a variety of fields in IT including but not limited to data analytics, software engineering and security.

Given the nature of the project and team, Agile methodology has been adopted for our software development as it embraces changes and develops the projects by incremental and iterative improvements. Specifically, after each consultation with the tutor, the team should identify the works to be done within a week and divide them into functional increments (user stories). Each of these increments is expected to be useful and builds upon previous versions. The team may also revisit and modify the user stories should requirements and ideas change. Because we have a cross-functional team, SCRUM sub-methodology is preferred over the alternatives. The weekly reports may serve as product backlogs that record lesson learnt during the current increment and plans for the next increment. The next increment should be developed and validated in a 7-day Sprint, after which the backlogs (weekly reports) should be written and analysed.

4.2 Database Design

In this project, data are all stored in Azure cloud SQL server and database schema need to be designed at first for further data analysis and web application development. For the database schema design, the main entities and relationships employed are showed in the figure below.

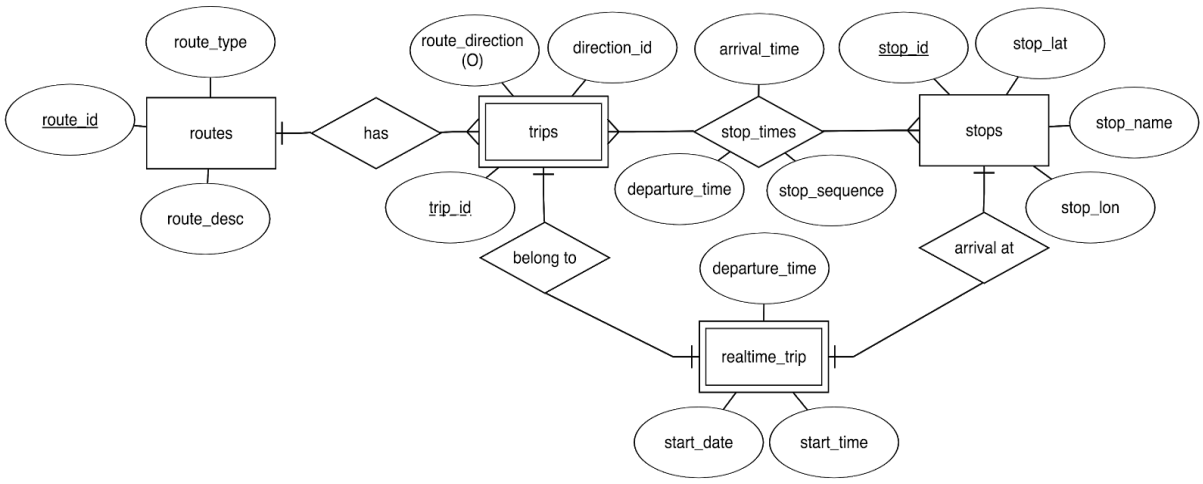

Figure 1: Database schema design

Additionally, we also set up firewall rules and grant different roles to our group members to keep data security. Firstly, the firewall would block all Transact-SQL access to Azure SQL server. We have set the server-level firewall rules while only adding IP addresses of our group members to the firewall list. Therefore, it can be guaranteed that only group members can access the cloud database. Moreover, authorization is needed for database users with different roles and object-level permissions. We grant different roles to group members with different responsibilities as: db_owner role (database manager), db_datawriter role (dcvcata analysis group), db_datareader role (web group).

4.3 Data Warehousing

To analyse service reliability of bus routes in NSW, we fetch real time data from Transport for NSW Open Data and Developer Portal (TFNSW Opendata), which is endorsed and funded by NSW government. In the scope of this project, we use the Public Transport datasets including Realtime Timetable and Realtime Trip Update datasets, both of which are accessible by public application programming interfaces (API) developed by TFNSW Opendata.

The Realtime Timetable dataset contains plain text files with data presented as comma-separated values (csv files). TFNSW Opendata updates this dataset only when major changes occur. On the other hand, the Realtime Trip Update dataset is updated every 10 seconds (Transport for NSW, 2017).

We automate the data collecting process on Azure Cloud so that it can be independent to possible network and computer issues on local devices. Moreover, it also speeds up the data warehousing process as the hardware and resources on Azure cloud are. Since Azure Automation is not available to free-trial users, automation is accomplished by WebJob service of Azure Web Server. The WebJob services runs on the background of a web application in time-triggered mode.

The design of the data warehousing module follows the classic Extract-Transform-Load (ETL) process as shown.

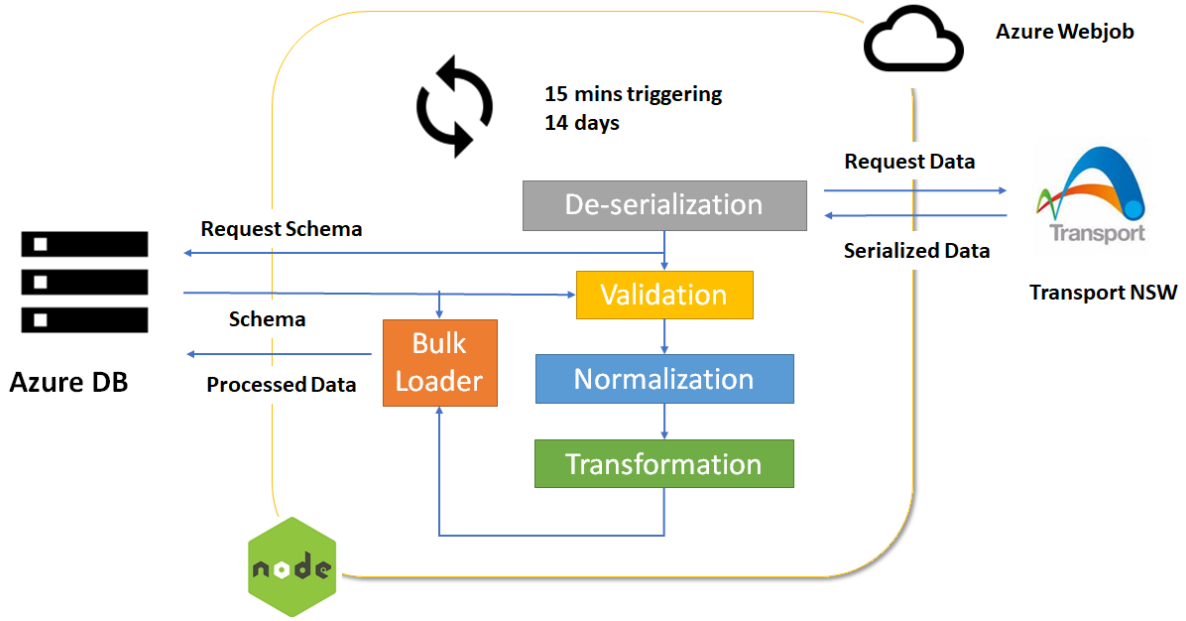

Figure 2: Data warehousing design flowchart

For Realtime Trip Update dataset, the data is in the format of Google General Transit Feed Specification based on Protocol Buffers. Protocol buffers allows structured data to be sent in a serialized manner, providing a fast and data-saving means of transferring chunks of data over the Internet. In this context, the structured data are Transport feed entities defined by a Google gtfs-realtime.proto file, which is used to deserialize the responding byte streams from Opendata API.

Integrity of each field in every record is validated against specifications provided by TFNSW (Transport for NSW, 2017) to eliminate false records, as well as against our predefined schema to ensure data types and nullability of each field matches. If inconsistency occurred, we would try to convert the record to match our requirement, or else reject the record completely.

The schema is retrieved from our database before the validation process initiates. Beside its usage in validation, the schema is also fed into the bulk loader which inserts records in aggregation. Comparing to multiple simple insertions, bulk loading speeds up the insertion process by eliminating redundant and repetitive tree operations, such as traversal and rebalancing.

As the raw data objects are nested with child objects, we need to normalize them to obtain fields that suit our data analysis purposes. The selected fields are then reconstructed based on our predefined schema and transformed into array/list structure for insertion purposes.

4.4 Data Analysis

4.4.1 Data Wrangling

We have collected 15-days realtime update data (more than 41 million pieces) and 3 versions of timetables and data wrangling is a crucial step for any further analysis due to the complexity nature of the realtime data and multiple versions of timetable. The process can be divided into four parts.

1. Union selected timetables to reduce execution time of data analysis modules.

2. Clean realtime update data.
   The retrieved realtime data is expected to have missing, repeated, or noisy records. In data wrangling, we eliminated some useless information of stops that would waste the computer storage and execution time, and only selected information of 10 stops every time and set the upload interval as 10 minutes. The data can cover nearly all stops information of the trips due to buses can only travel no more than 10 stops in 10 minutes. In addition, we also filter the data to keep only the newest one when the same stop information of the trip repeated.

3. Calculate realtime bus delay by comparing realtime data with timetables.
   The departure and arrival delay in the realtime update trip dataset is different with the delay calculated with the scheduled time in the timetable. To get the correct data of delay time, we choose to compare the real departure time with scheduled time in the timetable.

4. Add reasonable indexes for data in the database.
   Processing time is quite a significant issue due to this huge volume of the dataset. To lessen time consummation, reasonable indexes are needed. Instead of scanning through the whole relational table every time, using indexes makes it much faster to navigate and has significant impacts on the query performance. We had created several indexes before performing data wrangling on the datasets.

4.4.2 Significant Measurement Parameters

On-time performance of stops

This parameter is applied to classify the on-time performance of a particular stop in a trip. It is calculated by matching and comparing the real-time dataset with the bus timetable dataset. The measurements of a bus arriving to a certain stop as table is shown below (Table 1).

Measurement	Departure
Early	More than 3 minutes early
On-time	3 minutes early to 5 minutes late
Late	5 minutes early to 15 minutes late
Very Late	More than 15 minutes late

Table 1: Benchmark of on-time performance of stops

On-time performance of trips & routes

This parameter is applied to calculate the probability of whether a particular trip of a bus route or a particular route is on-time. For both trips and routes, all the on-time performance results of stops in each trip or route of are used to determine the on-time performance of each trip or route. The evaluation depends on the ratio of on-time and early stops on a trip or a route. The benchmark of on-time performance of trips and routes is shown as follows (Table 2).

Conditions			On-time performance of trips and routes
Percentage of on-time stops (%)		Percentage of early stops (%)	
>= 60			On-time
< 60 & >= 40	and	>= 30	Early
		< 30	On-time
< 40 & >= 30		>= 35	Early
		< 35 & > =30	On-time
		< 30	Late
< 30 & >= 20		>= 40	Early
		< 40	Late
< 20 & >= 10		>= 50	Early
		< 50	Late
< 10		>= 60	Early
		< 60	Late

Table 2: Benchmark of on-time performance of trips & routes

For instance, if a route has 100 stops, in which 50% of stops are on-time and 40% of stops are early, this route's on-time performance will be classified as early. Or for a trip or route with

more than 60% of its stops are on time, it would be considered as an on-time trip or route regardless of the total number of stops of this trip or route.

4.4.3 Descriptive Analytics

First of all, basic investigations of on-time performance are made at different layers of the dataset: bus stops, bus trip and bus routes. The distribution of each layer's on-time performance was designed to be plotted in histograms. In the initial phase, descriptive analytics is performed to see the overall performance of bus stops. A close observation on the most delayed stops would be performed.

Secondly, in order to learn about whether the delay pattern is related to the time period of each day, we would look into the stop's delay time for different peak hours in each day. Here is the table showing how the peak hours' ranges:

Morning Peak	Evening Peak	Other time
6:00 - 10:00	16:00 - 20:00	10:01 - 15:59 & 20:00 - 6:59

Table 3: Benchmark of peak hours

Thirdly, visualization on the distribution of "early", "on-time" and "late" routes will be presented to provide further insights on the performance of public buses in Sydney. After looking at the overall bus performance in NSW, a closer observation on the most delayed bus route were designed to be presented using visualization methods.

Last but not the least, after plotting the basic observation on the on-time performance, some correlations between on-time performance and other related variables are expected to be observed. During this step, cluster analysis will be used to testify our findings. Two initial focuses in the correlational observation are stated as follow.

- whether a route with more frequent bus trips is more likely to delay.
- whether the distance of the route has certain relationship with its on-time route performance.

4.4.4 Predictive Analytics

The reason for constructing this prediction model is to help managers or government officers to make decisions on adding new routes or adjusting the existing routes. In order to extract and make use of information in the predictive model, a geographical information was added to the dataset – Whether across CBD. Whether across CBD was used to determine whether a stop is located in the CBD area and whether there is any part of a route is in CBD. This feature was

generated using the stop's longitude and stop's latitude in the original dataset. Information on the coordinates of CBD area were referred to the Sydney Australia Tourist Guide.

Since the outcome for the prediction of on-time performance of routes is scoped to early, on-time and late class, several classification algorithms are used for performing the predictive analysis. And the classification algorithms used to build the model are Logistic Regression, K Nearest Neighbours (KNN), Gradient Boosting and Random Forest Classifier. The predictive analysis of this project owns two important steps among the process.

The first one is to select features from the data to predict the on-time performance for routes. In order to provide an additional insight to the prediction model, a new feature called "whether across CBD" is generated, which use the longitude and latitude information to determine whether a route would across the defined CBD area.

The second one is to calculate the on-time performance of routes which is the target classification results of the prediction model. The calculation benchmark of the on-time performance of routes has been introduced in the above part. In addition, the dependent variables of this calculation are rates of on-time stops and early stops for each route, which are calculated with the following equations:

On-time rate = Number of on-time stops / Total stops
Early rate = Number of early stops / Total stops

The predictive modelling in this project is initially set up using some the independent variables, including travel distance of routes, number of stops, routes whether across CBD. Then we study on how each independent variable has its impact on the on-time route performance. Finally, the above machine learning classification methods will be applied to achieve our goal of building the prediction model.

4.5 Web application development

The beginning to implement this web application is the overall design of website structure including the appearance and functionalities. We utilize Dashboard UI Kit of Adobe XD to make the mock-up of our website interface. The mock-up aims to provide a guidance of our website development. The general appearance features in the mock-up could be achieved by Bootstrap library and the functionalities implementation could also be obtained by using a few frameworks and developer APIs.

MVA design pattern is used to construct the structure of website code. The process of events flow for this website under MVA structure is shown as follows (Figure 3). As the flow shown in the figure, when users visit our website and make a request of any type, the request would flow forward to the Router as the intermedia to Controller. Then the Controller passes the input

to Model and we conduct all the logic operation in the Model to get the required data from our Azure database through Tedious module. Since we would use Node.js to develop the server-side application in Router, Controller and Model, we use Tedious package for Node.js to connect the database and read data from database. Furthermore, we use Express.js framework to write our server-side code more flexible and succinct. For the client-side application, we would all finish in the View part and Controller updates received data to specific html pages written in EJS templates in the View. The rendering method of received data on the webpage would be decided by the corresponding JavaScript file for each page. For instance, we employ the Google Map JavaScript API to render any received data for one selected route as route line and stop markers onto the map on the webpage. And Google Charts is utilized to turn data of analysis results to different categories of charts and display on the page. For the prediction model, we choose to export the prediction model in the python script into JavaScript functions and then present the prediction result as a spider web chart on the predictive page.

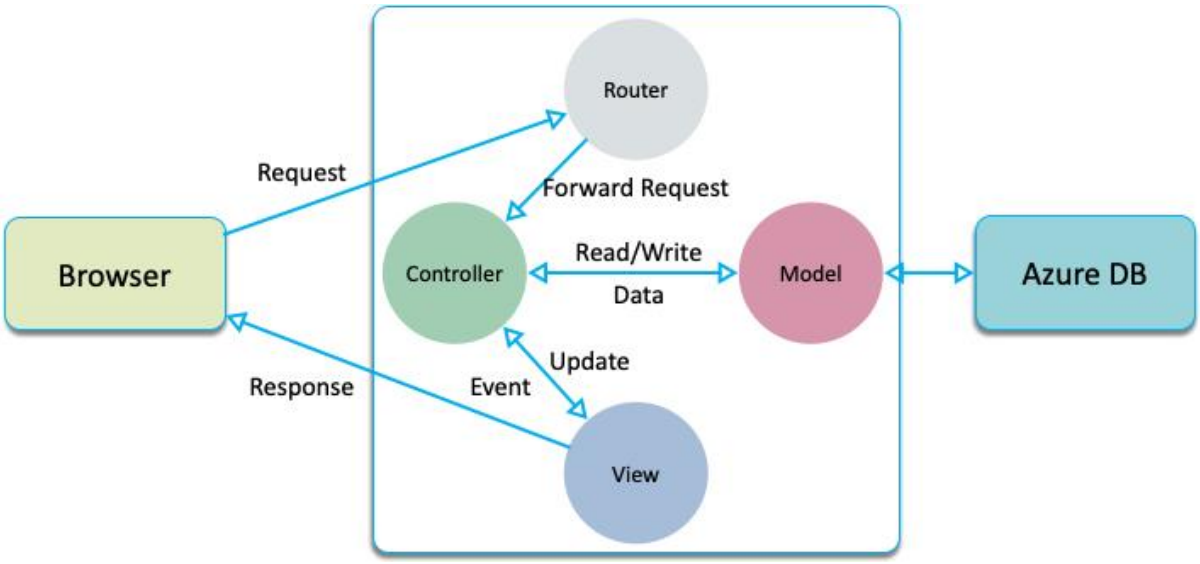

Figure 3: The process of events flow under MVA structure

4.6 Deployment

The necessity of web deployment comes in two folds:

- The port 1443, that Azure SQL uses, may be blocked by University's firewall. This has been tested and confirmed by using network tools (e.g. Telnet, Wireshark). As a result, the application running locally will not be able to retrieve data from our database.
- The web application, by definition, should be accessed by anyone on the web.

The web application is deployed on Azure Web Server. Because Azure Web Service (and most web services) is not compatible with ECMAScript 6 (ES6) syntax and our web application is written mainly in ES6, we need to trans-compile ES6 to ES5 before deployment using Babel. In addition, since some of the front-end JavaScript files are dependent on back-end libraries, they need to be bundled into front-end files to ensure functionalities. The front-end files also

need to be minimized for network performance purposes. In this way, server-side is solely responsible for all logics for the application, making it easy to conduct upgrades, maintenance and bug fixes.

4.7 Testing

For Agile methodology, it is recognized that testing should be an integral part of software development alongside actual coding. Naturally, we are inclined to use Test Driven Development (TDD) approach where a test is written just before enough production code is written to fulfil the test. The advantage of TDD is that it allows small low-level units to be validated and functional before integrating them into a higher-level component or a system. This implies that all members of the team need to participate in testing during developing.

Following the Agile Testing Quadrants (Crispin, 2008), we implement tests at four levels, namely unit testing, integration testing, system testing and acceptance testing. The unit testing is conducted using unit test tools/libraries, such as JavaScript testing framework Mocha and assertion library Chai. At integration test level, all functional requirement of the integrated component and interfacing between units are examined and validated. After that, we perform tests on the entire system to test its functionality, performance and reliability. Soak test and load test are not implemented at this stage, because our prototype is not expecting many visits. Finally, before going on to deployment, we have asked the clients (in our case, our tutor) to use the system and make changes according to their feedbacks.

5. RESOURCES

5.1 Hardware & Software

Cloud Service

The resources used from cloud service are as follows

Application Type	Service Type	Account Type
Azure SQL	Standard Services Tier 1 (S1)	Free trial
Azure Web Server	Free Services (F1)	Pay-as-you-go

Table 4: Resources for Cloud Service

Note that

- The Standard Service Tier 1 service is free under free-trial account that lasts for 30 days. It provides production-grade resources but will cost around $400 per months after free trial period. Our database server runs on S1 whereas our web server is on F1.
- Azure SQL is a relational database-as-a service (DBaas) that uses a specific dialect of Transact-SQL (T-SQL).
- Our web application/data warehousing automation runs on Azure Web Server under NodeJS version 8.9.1 environment.

Data Warehousing

The resources used for data warehousing are as follows.

	Language	Libraries/Tools/Frameworks
Production dependencies	JavaScript (ES5) Transact-SQL	Node, Tedious, Request, GtfsRealtimeBindings
Development tools	JavaScript (ES5)	Npm, Jsdoc
Testing tools	JavaScript (ES5)	Mocha, Chai

Table 5: Resources for Data Warehousing

Note that

- Node.js version 8.9.11 is used for ES6 compatibility purpose, although any Node.js with version 5 and above should also suffice.
- GtfsRealtimeBindings is used for deserialize GTFS entities, compatible with Node.js 6 +.

- Npm is a package manager and automation tool for JavaScript, like Maven or Gradle. Npm makes sure that all dependencies are installed with versions with Node.js 6+.
- Tedious is used to interact with instances of Microsoft's SQL Server (including Azure SQL).
- Request handles application layer communications with TfNSW.
- Jsdoc automatically produces documentation based on comments.
- Mocha is a JavaScript testing framework which is used alongside with Chai, an assertion library.

Data Analysis

The resources used for data analysis are as follows.

	Language	Libraries/Tools/Frameworks
Descriptive Analysis	Python, Transact-SQL	Jupyter Notebook, Pyodbc, Pandas, Sqlalchemy, Numpy, Plotly
Predictive Analysis	Python, Transact-SQL	Jupyter Notebook, Sklearn, Pyodbc, Pandas

Table 6: Resources for Data Analysis

Note that

- Pyodbc and Sqlalchemy are collectively used to access and query our database.
- Plotly is a plotting library that plots mathematical graphs and store them in Plotly's cloud server.
- Sklearn is a machine learning library for Python.

Web Application Development

The resources used for web application development are as follows.

	Language	Libraries/Tools/Frameworks
Client/Browser	HTML, EJS, CSS, SCSS, JavaScript (ES6)	Bootstrap 4, Font-awesome, Google Map API, Google Chart API jQuery, Popper
Web Server	JavaScript (ES6)	Node, Express

Database Integration	JavaScript (ES6), Transact-SQL	Tedious
Development tools	JavaScript (ES5/6)	Babel, Eslint, Nodemon, Webpack, Npm

Table 7: Resources for Web Application Development

Note that

- Bootstrap 4 is used for the client-side development which is a useful framework for writing HTML, CSS and JavaScript.
- jQuery helps for using JavaScript on the website which makes it easier to simplify front-end scripts.
- Google Map API is a tool for developers to present maps along with information on the own application.
- Google Chart API provides packages for developers to draw charts with provided data on the application.
- Express is a framework simplify the development of serve side using Node.js.
- Babel package can trans-compile JavaScript ES6 to ES5 before deployment.
- Webpack bundles and packages the JavaScript files for using in one browser.
- Nodemon is an auxiliary tool, which can automatically restart the changed application once detected, to make debugging and evaluation easier.

5.2 Roles and Responsibilities (Human Resources)

Name	Role	Responsibilities
Kun Zhang	Project Manager	Manage all the aspects of project including change management, task assignment, conflicts resolution. tracking project schedules, etc.
Kun Zhang & Yuansi Xu	Data Engineer	Data collection and cleaning, database construction and database management.
Qingqing Peng & Xuying Wang	Data Analyst	Perform data analysis using python.
	Machine Learning Analyst	Train prediction models using machine learning techniques.
Kun Zhang & Zijun Wang	Web Application Developer	Design and develop all client and server logic of the web application and conduct white-box testing.

Table 8: Roles and responsibilities of the project

6. MILESTONES AND DELIVERABLES

The table below summarizes the milestones of the project.

Date	Milestones	Deliverables/Reporting
10-08-2018	Team formed.	
17-08-2018	Work allocation confirmed. Local data collection module development started.	Weekly Progress Report 3.
24-08-2018	Local data collection module completed. Started to pulling data from TfNSW Opendata API to local directories. Data analysis development plan designed.	Weekly Progress Report 4. Local data collection module. Data analysis development plan.
31-08-2018	Project plan formulated. Project proposal enacted. Database design started.	Week 5 Summary Report. Project plan. Project Proposal.
07-09-2018	Azure Cloud database & automation services ready. Database design completed with timetable data imported. Local data pulling finished ---- data stored as csv files in local directories. Descriptive analysis on local data started.	Weekly Progress Report 6. Local data (7-day).
14-09-2018	Descriptive analysis on local data completed. Automated data warehousing module development started.	Weekly Progress Report 7. Descriptive analysis module based on local data.
21-09-2018	Data warehousing module completed.	Week 8 Summary Report. Data warehousing module.
01-10-2018	Data warehousing module started to pull data from TfNSW Opendata API to cloud database. Web application UI design started.	

05-10-2018	Descriptive analysis on two days' data completed. Web application UI design finalized.	Weekly Progress Report 9. Descriptive analysis module based on two-day's data. Web application UI design.
12-10-2018	Web application development plan devised. Predictive analysis development plan devised.	Weekly Progress Report 10. Web application development plan. Predictive analysis development plan.
15-10-2018	Data warehousing module finished running ---- data ready. Predictive analysis development started.	15 day's data.
19-10-2018	Descriptive analysis on 15 day's data completed.	Weekly Progress Report 11. Descriptive analysis module based on 15 day's data.
26-10-2018	Predictive analysis completed. Web application development completed. Integration and system tests started.	Weekly Progress Report 12. Predictive analysis module. Web application running on local machines.
31-10-2018	Web application deployment completed. Integration and system tests finalized.	Web application running on the Web.
02-11-2018	Final Presentation	Final Presentation.
16-11-2018	Final report completed. Github documentation written.	Final Report, Github documentation.

Table 9: Milestone Descriptions

Deliverables Description

- Project requirements documentation defines specific requirements for this project contains meetings with clients, contact with the original project contributor and discuss among project team members.
- Database/Data-Analysis-Model/Website design: The database models was designed at early stages of this project as well as the overall design of data analysis model frame and website structure.
- Cloud database server: A cloud database server is deployed to store all project data and data analysis results. The database model is applied at this step.
- Data analysis results: Predicted delay results is achieved through multifaceted analysis of the bus trip data using a predictive model. The results present the main information to provide for clients.
- Website: The website is accessible via https://bus-delay-analysis.azurewebsites.net. The design of the website oriented to different clients such as government, passengers and bus operators. The web service interface supplies various query functions to show the data analysis results graphically for all bus trips in Sydney.
- Final Report: The final report the detailed description of the project.

7. RESULTS

7.1 Data Warehousing Results

The data warehousing module ran on a single process on Azure Webjob. After having run for 15 days at 15-minute intervals, the data warehousing module was able to retrieve and store the data needed for data analysis purpose. It had been found that the execution time of the same script under comparable conditions was much shorter on WebJob than that on a local machine, as showed in an example in figure (Figure 4) below.

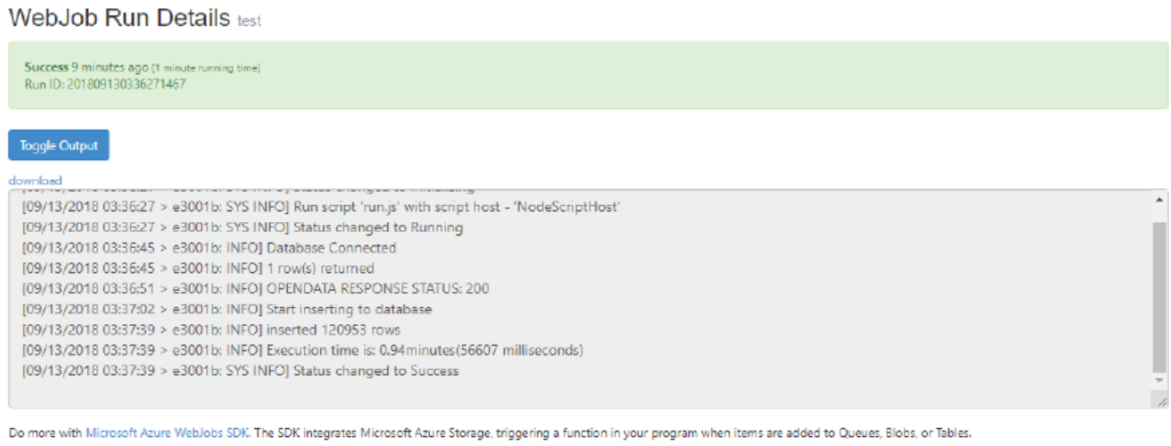

Figure 4: Data Collection Script running on Azure WebJob took 0.94 minutes

It can be seen that the module running on Azure WebJob only took less than one minute to insert 100000+ rows (one session).

7.2 Data Analysis Results

For data analysis part, we have completed all the requirements pointed in the project's aims and scope of this part. The outputs of data analysis are mainly divided into two aspects which are the data analysis results of current bus system and a predictive model for the on-time performance of future bus routes. There are five main outputs for data analysis of current bus routes including an overview of on-time performance for stop level, trip level and route level respectively, delay time distribution of bus stops for different time periods of one day, the top 10 most delayed bus stops, the top 10 most delayed bus routes and the detailed on-time performance for each bus route, which are all presented by charts or tables. Moreover, our predictive model can give an optimal result among five different machine learning models, which will be interpreted in the following part.

Figure 1 shows the line chart of on-time performance for stop levels during the period between 1st of October and 15th of October in which the vertical axis is the number of stops and the horizontal axis is date. It is obvious that except the first day, i.e. 1st of October, the number of on-time stops are much more than other three different performances, which means that current bus system could give an on-time service generally. The reason for the first observation day

has the lowest number of stops with on-time performance may be the fetched data for October 1st are not the integrated data for this day. The whole trends of these four different performances are similar that the number of bus stops get lessened on weekends due to the less number of total bus trips compared to weekdays. Giving insight into the on-time performance for trip level shown in Figure 6, which has a very similar result with on-time performance for stop level, the number of on-time trips are much greater than that late and early trips and all four types of performance owns a decreasing trend on weekends.

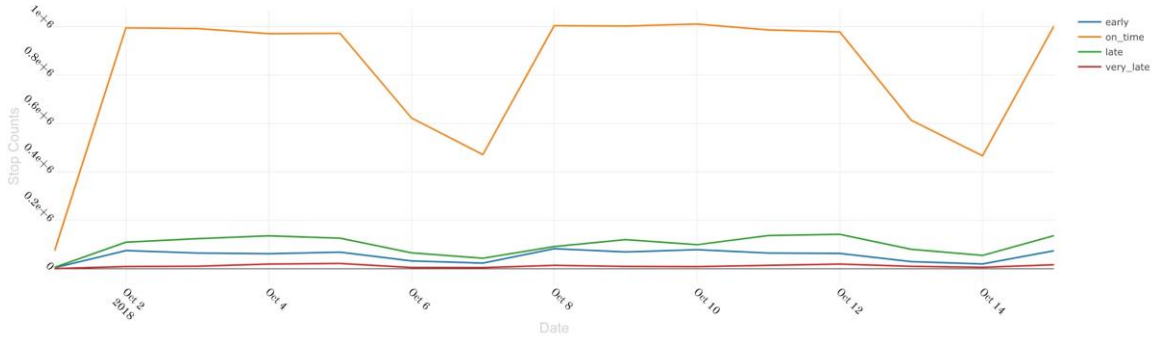

Figure 5: On-time performance for stop level

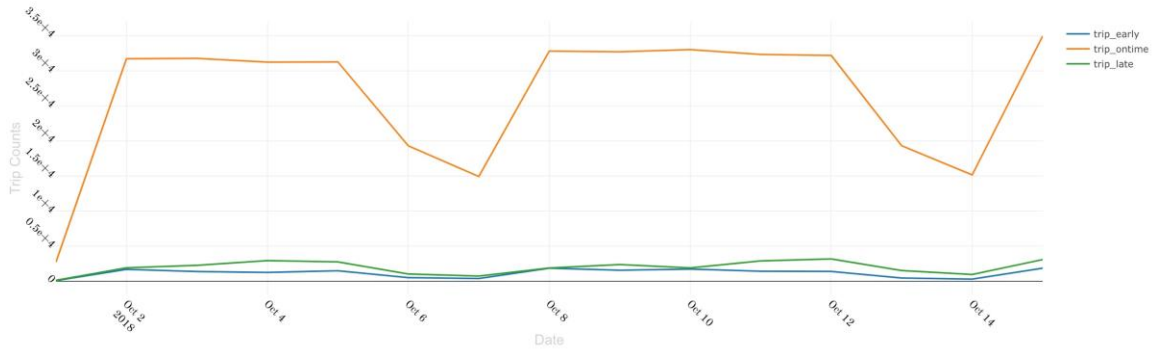

Figure 6: On-time performance for trip level

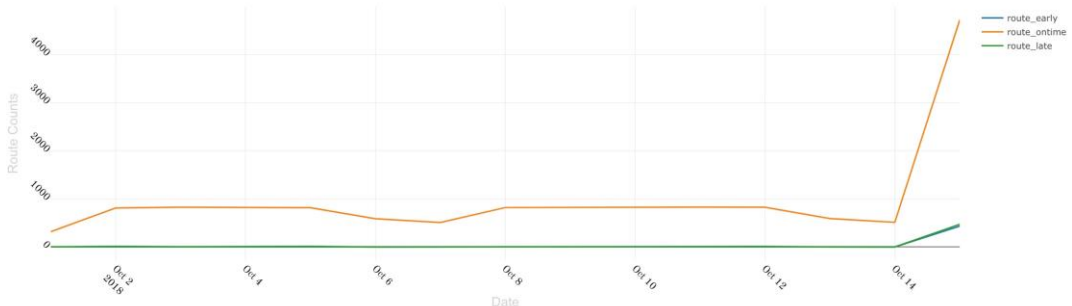

Figure 7: On-time performance for route level

However, the on-time performance for route level in Figure 7 does not show the similar trend. It can be found from the chart that except the last observation day, there are no early or late routes. The weird result of the last day is on account that all NSW schools started their fourth-term semester on this day. Therefore, on this day both the number of late and on-time routes are dramatically increased.

Meanwhile the clients may be interested in whether different time periods of one day would have impacts on the delay time distribution of bus stops or not. The figure below (Figure 8) shows the boxplot of delay time distribution of bus stops for morning peak, evening peak and other normal time in a specific day. It seems there are no significant difference among the delay time performance of these three time periods generally. However, if we observe the data distribution more carefully, it can be found that the evening peak owns the widest interquartile range with two most widely separated minimum and maximum values compared to the two other time periods which means during the evening peak period, the on-time performance of Sydney bus system is worse than other time periods in one day.

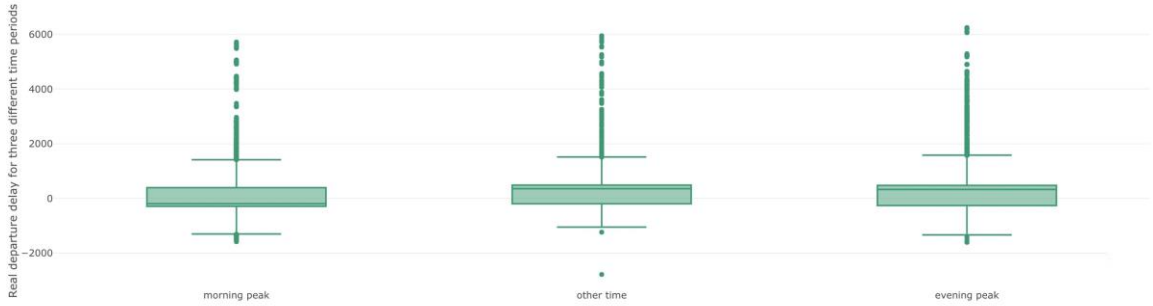

Figure 8: Delay time distribution of bus stops for different time periods of one day

Apart from the above overview information of Sydney bus system, clients may also pay special attention on some typical bus routes, hence the top ten most delayed bus routes (school bus routes are excluded) is figured out and shown in Figure 9 below. It is notable that more than 5 bus routes are across or near the CBD area.

route_id	route_long_name	total_delay_time	total_stops	average_delay
2459_X04	Opal only - City Domain to Chiswick (Express Service)	704695	998	706.107214
2433_738	Mount Druitt & Rooty Hill to Eastern Creek & Horsley Park (Loop Service)	2496347	6886	362.524978
7000_SL1	Epping to Chatswood	3955842	12499	316.492679
2439_275	Castlecrag to Chatswood	792149	2509	315.722997
2439_X18	PrePay Only - City Town Hall to Denistone East (Express Service)	327730	1060	309.179245
2439_X06	PrePay Only - City Domain to East Ryde (Express Service)	475162	1548	306.952196
2433_745	St Marys to Castle Hill via Stanhope Gardens	5418615	19320	280.466615
2459_526	Burwood to Rhodes Shopping Centre	9172580	32716	280.369850
2433_723	Mount Druitt to Blacktown via Eastern Creek	5554890	20125	276.019379
2459_502	Five Dock to City Town Hall	1649864	6121	269.541578

Figure 9: Top 10 most delayed bus routes

Moreover, the detailed on-time performance for each bus route has been worked out through our analysis. In this report, we choose the most delayed bus route named X04 with route from City Domain to Chiswick as an example to present the detailed analysis results of the single route. The overall percentage of performance for route X04 shown in Figure 10 is distributed as 63.3% of late performance and the rest of on-time performance. Also, the distribution of on-time performance for bus route X04 during the observation period of 15 days from October 1st to October 15th is shown in the Figure 11 below which indicates that no stops in this route have early performance and usually more than half percentage of stops performed late for each day of the explored period as the number of late performed stops are greater than on-time performed ones for 12 days out of 15 days.

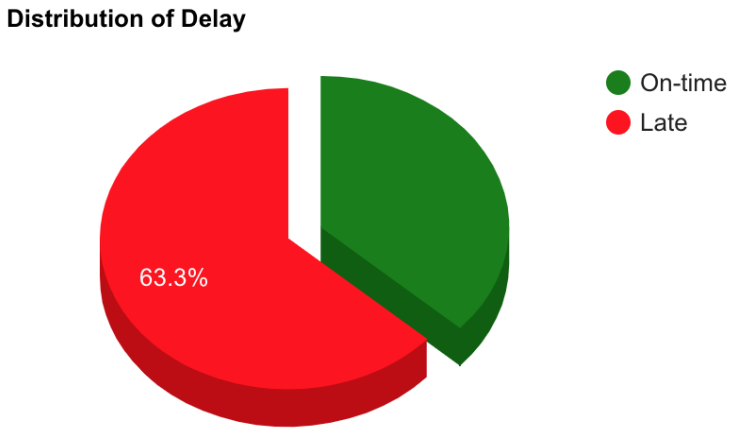

Figure 10: Distribution of delay performance for bus route X04

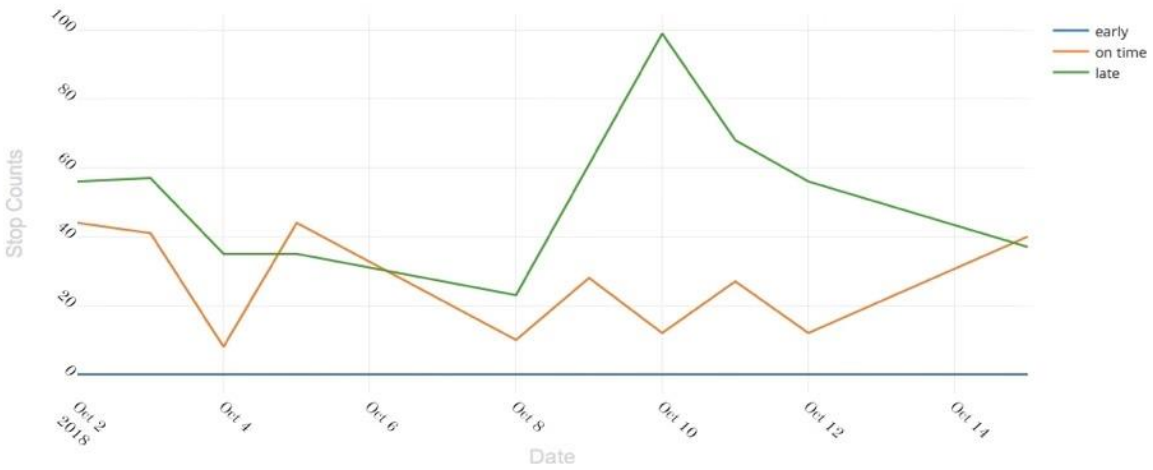

Figure 11: Distribution of on-time performance for bus route X04 during the 15 observation days

Furthermore, the on-time performance of each stop for each route is also figured out and provided for clients. Figure 12 provides an example of on-time performance of the last stop for this route X04.

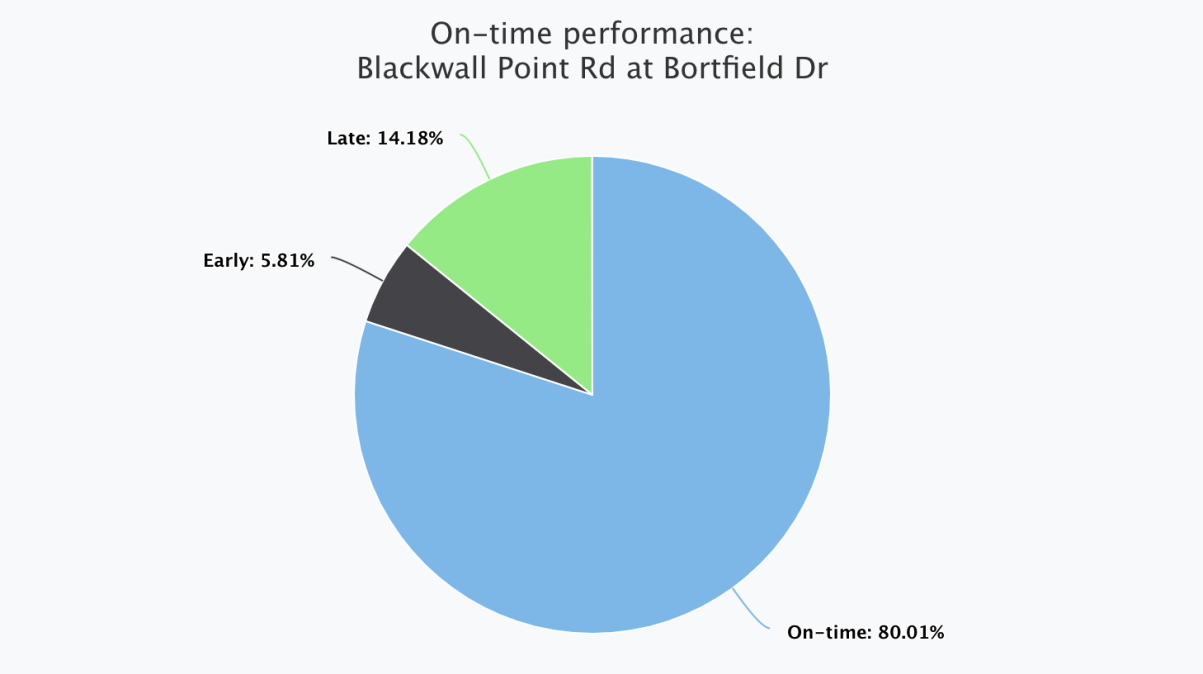

Figure 12: On-time performance of one stop for bus route X04

Besides the above data analysis results of current bus system, this project also provides a predictive service for clients that utilizing our predictive model, the on-time performance of a future route could be predicted. Four different models are trained and tested, and the results of prediction performance are presented in both Figure 13 and Table 10. According to these two results, the best model is Gradient Boosting classifier since its accuracy, 10-fold accuracy as well as confusion matrix performance are all the best ones among these four classifiers. Therefore, gradient boosting classifier is chosen to be the optimal and final predictive model for our project.

Model	Accuracy	10-fold Accuracy
Logistic Regression	0.678	0.686
KNN	0.984	0.988
Gradient Boosting	0.992	0.997
Random Forest	0.988	0.995

Table 10: Accuracy performance of prediction for four predictive models

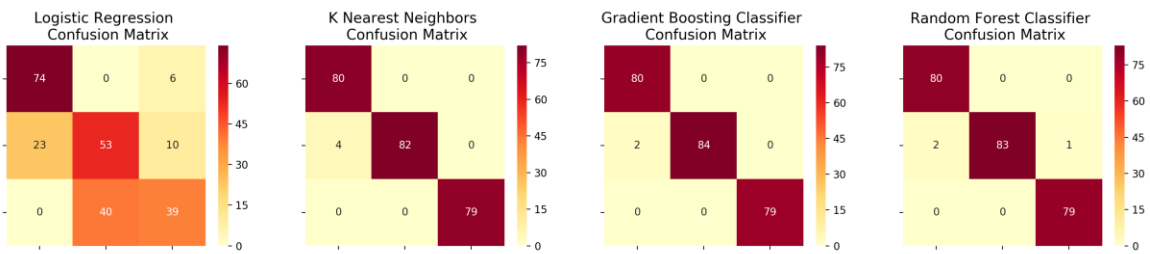

Figure 13: Confusion matrix of four predictive classifiers

We present an example prediction result here in Figure 14. When setting 1000 meters as the route distance, choosing route not to be across CBD and setting the total number of bus stops as 80 in one future route, the prediction model would give a prediction result of 100% early performance of this route.

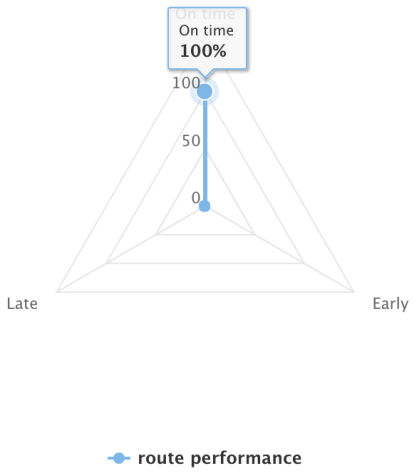

Figure 14: Prediction result of one selected future route

7.3 Web Application

The final web application we have developed can be found by visiting the URL below:

https://bus-delay-analysis.azurewebsites.net

or checking the source code at

https://github.com/kunzhang1110/COMP5703-Capstone-Project/

This website titled with "Sydney Bus Delay Analysis" contains four main pages which are the overview page, route analysis page, predictive analysis page as well as the about page.

When a user enters the url address provided above, the overview page would show up as this page also serves as the index page of our website. As the original view of this overview page shown in Figure 15, all pages in this website compose of a navigation bar and a sidebar as well as the main content part of each page. Since this is a "Single-Page" web application, the changes of content for each page would only show on the main content chunk with the navigation bar and sidebar remaining unchanged. For the navigation bar, this is one button beside the text "Bus Delay Analysis" at the left side which can control the showing and hiding of the sidebar and at the right side there is a search bar for users to search for bus routes. Meanwhile, on the sidebar, four buttons with icons and names are provided for users to visit different pages. And for the main content part of each page, firstly, the overview page provides a glance of overall on-time performance for the Sydney bus system to users including three

parts. The first chunk as shown in Figure 16 would display the distribution of delay performance by date on stop, trip or route level as bar charts. Users can choose these three levels in the drop-down box to get the corresponding level of bar chart. For instance, an example of the distribution of delay performance by date on trip level is shown on Figure 16. And the second chunk presents the trend of delay performance by date on stop, trip or route level as line graphs similarly to the first chunk and Figure 17 is one example for trip level. For the third chunk, a box plot showing delay spreads for different time intervals would be displayed once this page is loaded as in Figure 15.

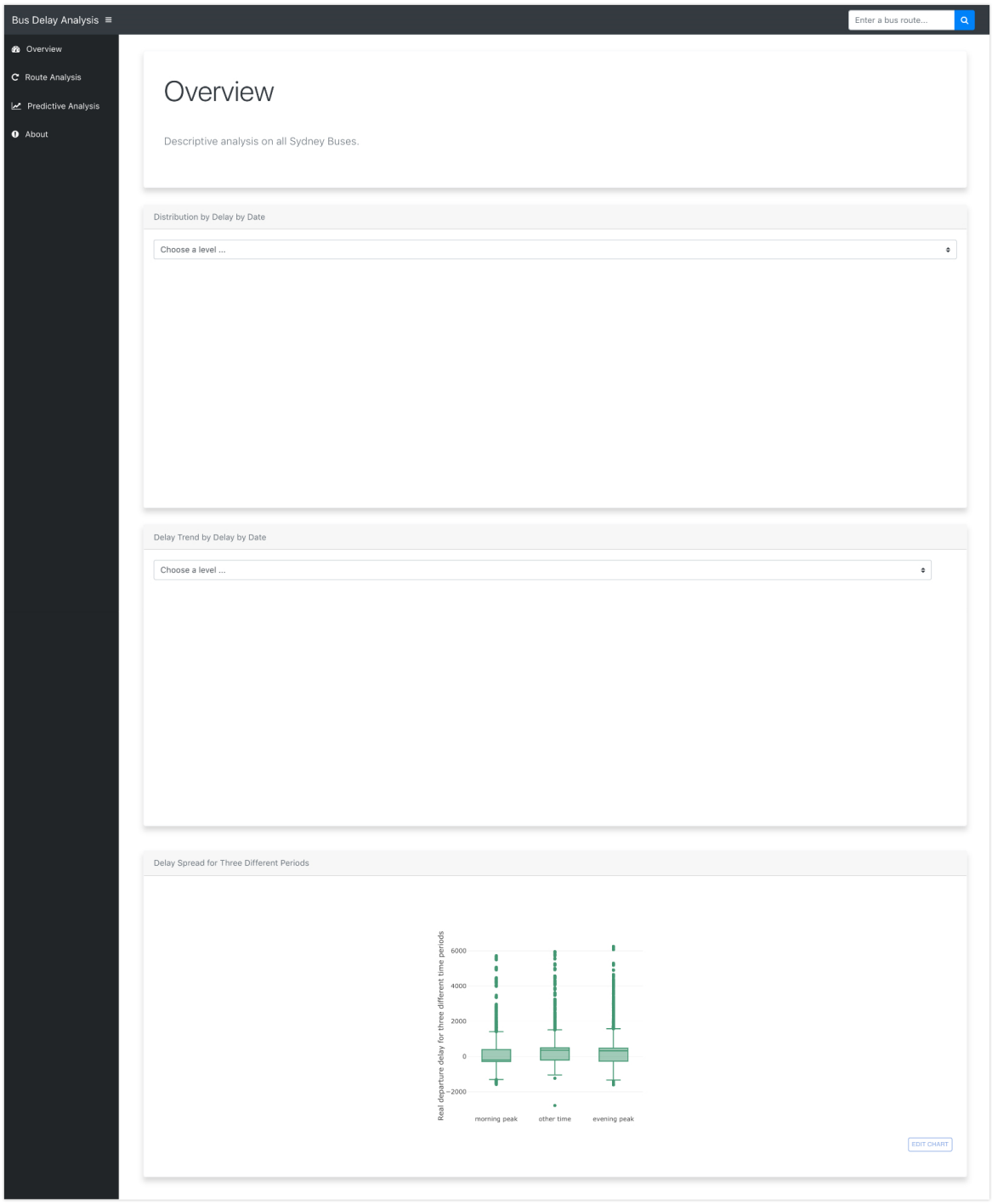

Figure 15: View of the overview page

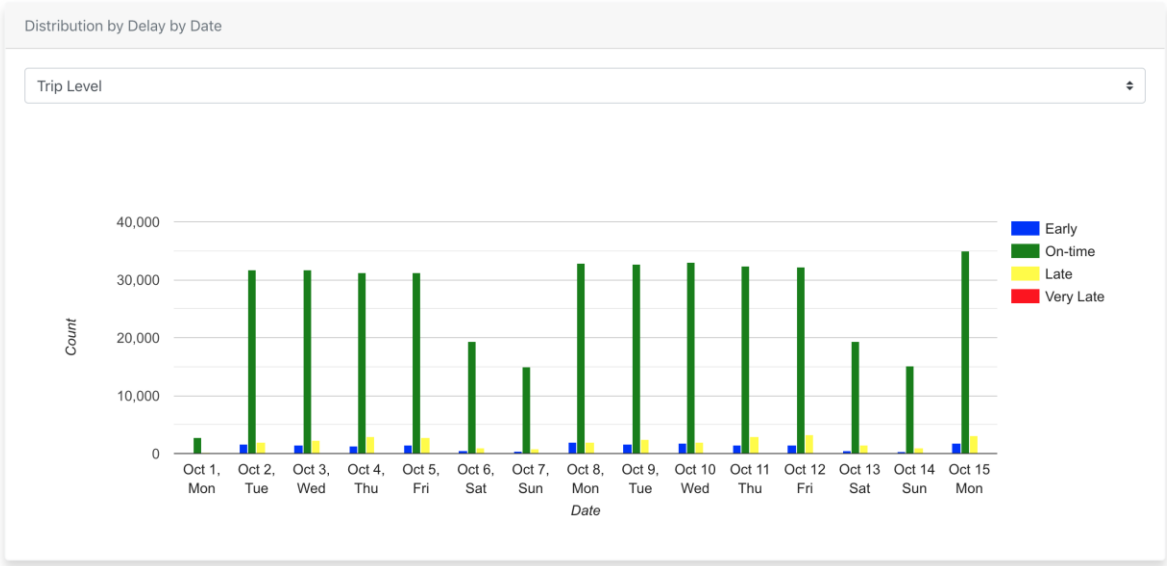

Figure 16: Distribution of delay performance by date on trip level

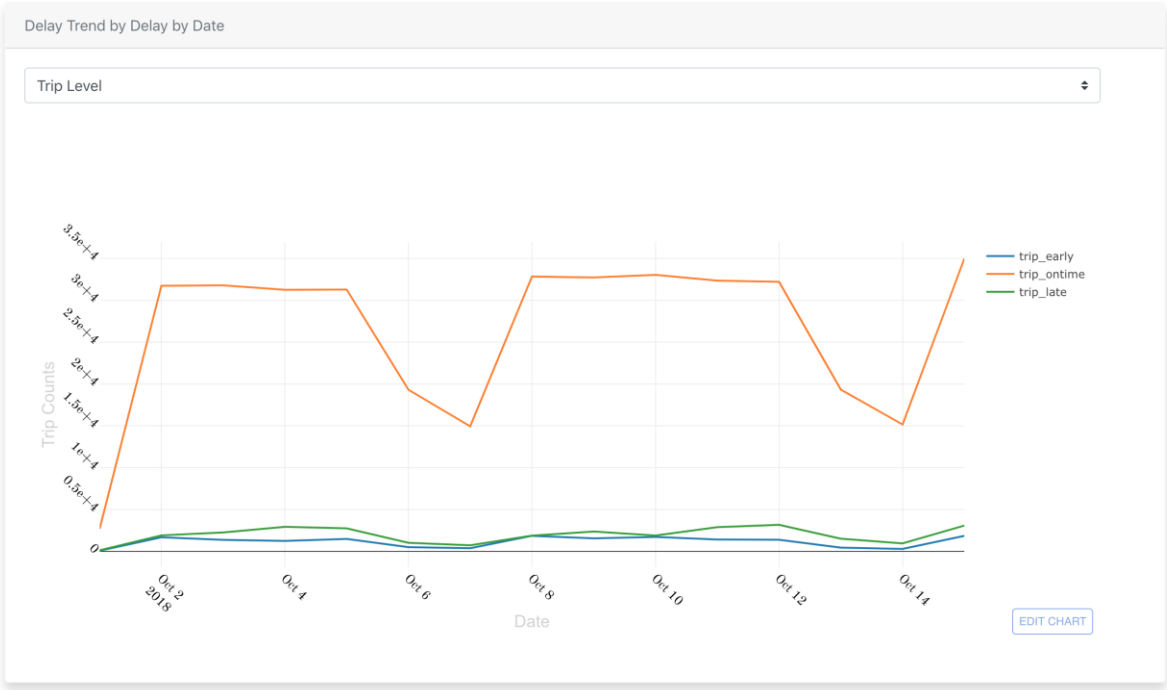

Figure 17: Trend of delay performance by date on trip level

The main content part of route analysis page as in the Figure 18 shows a search bar having the same function with that one on the navigation bar to allow users to enter the name of a bus route and redirect it to a page that contains static information as well as analysis results of the searched bus. Also, links for the analysis results pages of several popular or notorious routes such as route 370, X04 are listed below the search bar on the route analysis main page like Figure 18 shows. We choose bus route 370 as the example to show the analysis results page. Figure 19 shows the top half information on the analysis results page of route 370. The static

information such as the route name, route direction, network and operator are displayed on the top part of this page. Below the common information, it would show the on-time performance of this route by both card form with text and pie chart.

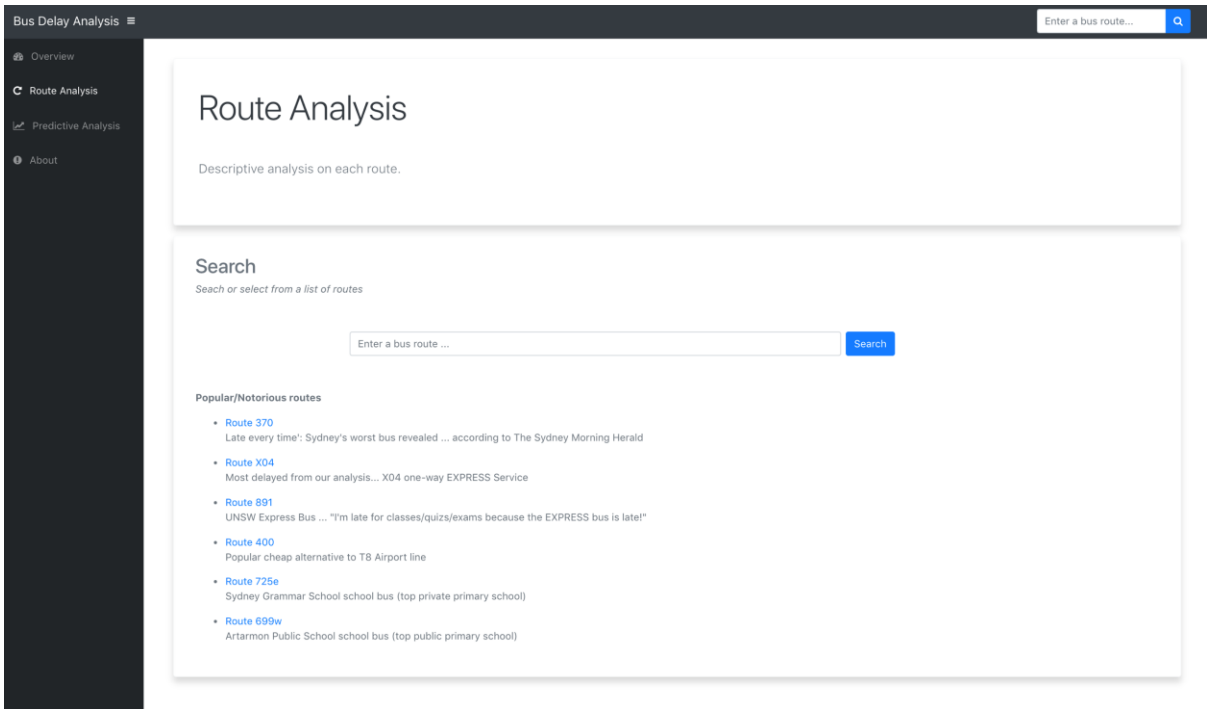

Figure 18: View of the route analysis page

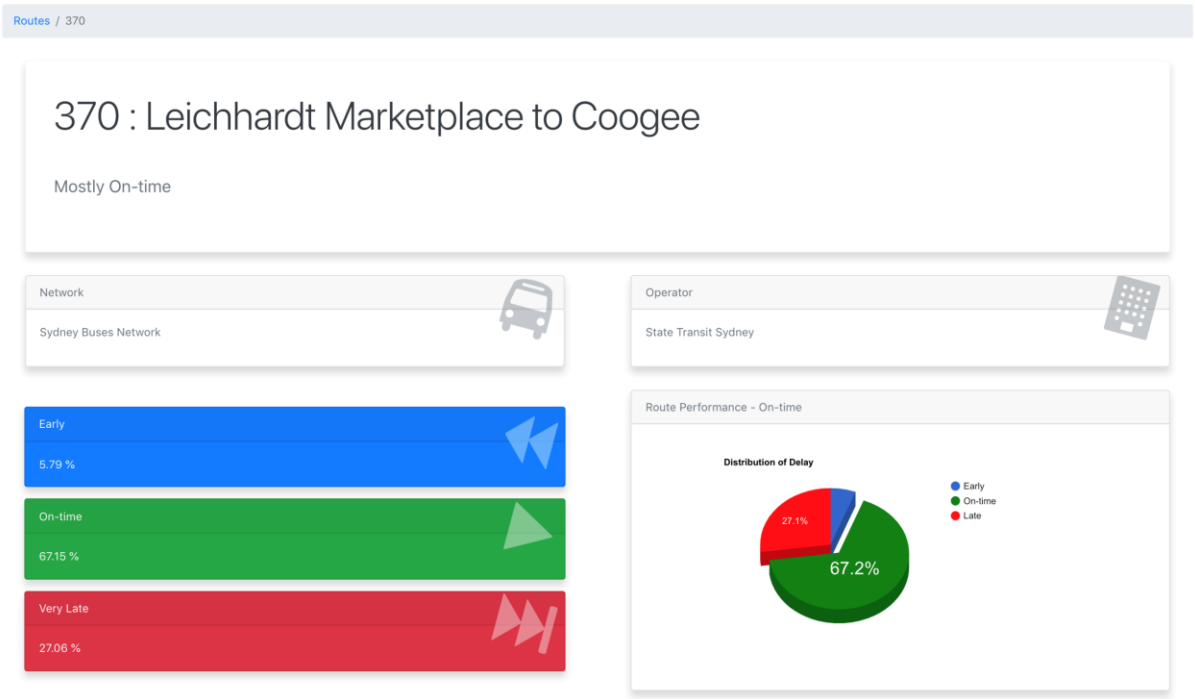

Figure 19: Analysis results for route 370 on route level

Furthermore, the bottom half information of this page is shown on Figure 20. The page also shows names of all stops in this route in sequence along with the map. On this map, we show

the virtualization of route lines as the figure shows. It can be seen from the figure that the bus route is drown by blue line and all the stops of this route are highlighted by blue markers with white border in sequence. Besides, the map would always be centred at the middle position of the route line with suitable zoom size to give a clear presentation of the bus route to users. Just as the figure shows, when we click the name of one stop of the route, the "stop details" button of this selected stop would appear and meanwhile the information window of this stop would show above the corresponding stop marker on the map with the sequence number and stop name on it. Furthermore, if we click the "stop detail" button, the on-time performance of this stop would present on the right side of the view of stop names as pie charts. In addition, for routes with trips of two opposite directions like route 370, we can click the "Change Direction" button beside stops title to get the new stops information and new map with the route line of trips with the adverse direction of the same route.

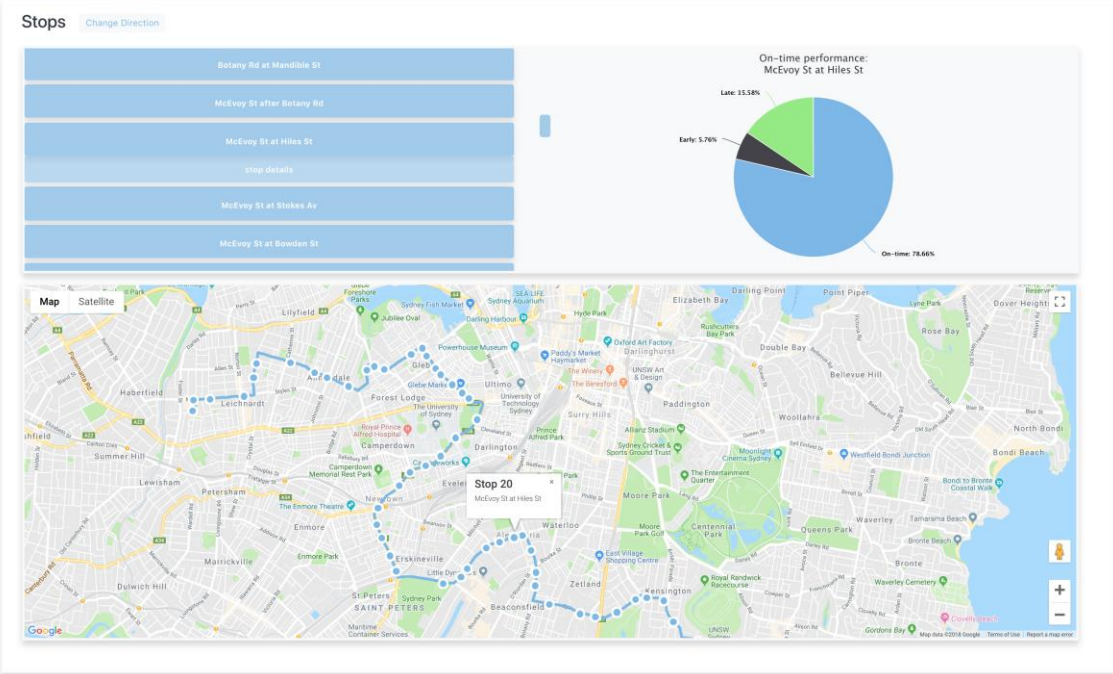

Figure 20: Analysis results for route 370 on stop level

For the predictive analysis page, we put three form items for users to enter inputs value for our predictive model as shown on the figure (Figure 21) below. The first one on the left is to enter the distance of the future route to be predicted. The middle one is to choose whether the route would across CBD and the last one is to enter the number of stops for this future route. When users input a set of values and click "Predict" button, the prediction result of this future route would be presented below the form items like the example in Figure 22.

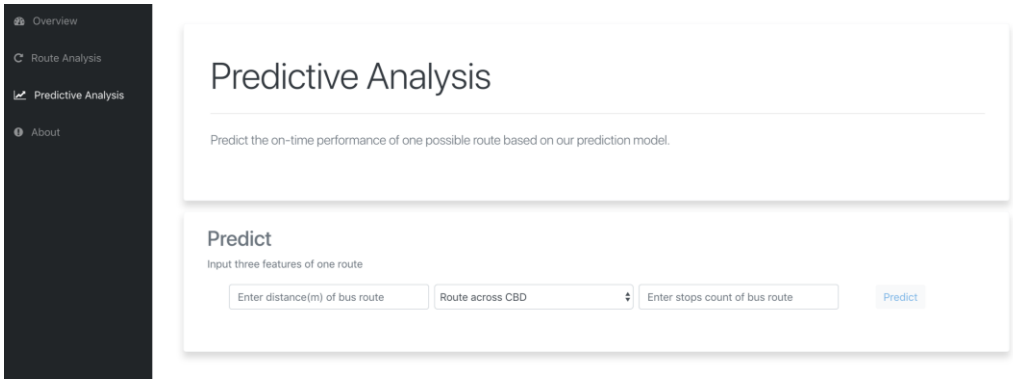

Figure 21: View of the predictive analysis page

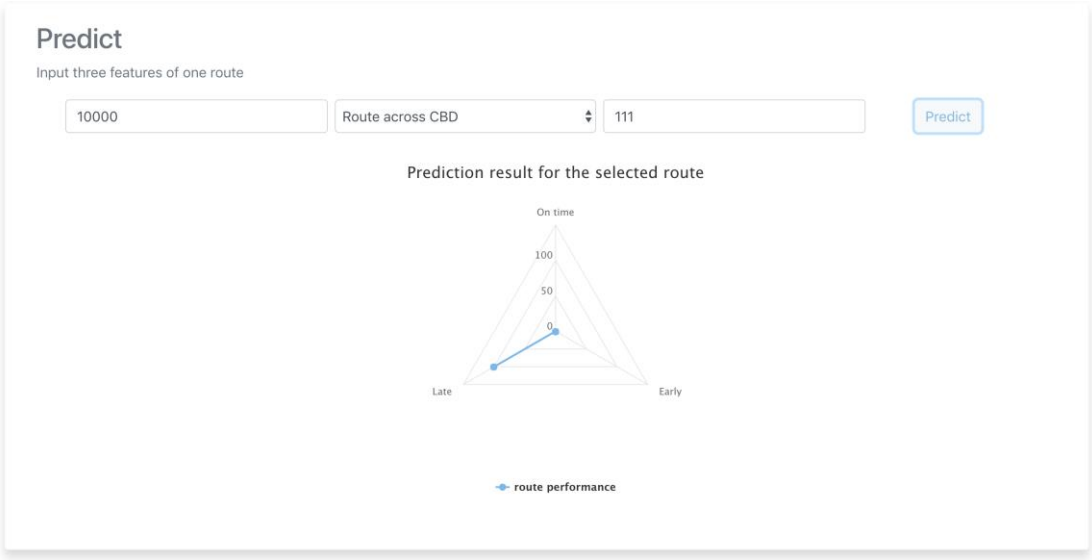

Figure 22: Example of prediction result

Finally, for the about page, the purpose of this project is displayed. Additionally, several developer API are also supplied for users to access some of data in our database. For instance, users can click the "/api/getRoute" link to get the all route-level analysis results of our project.

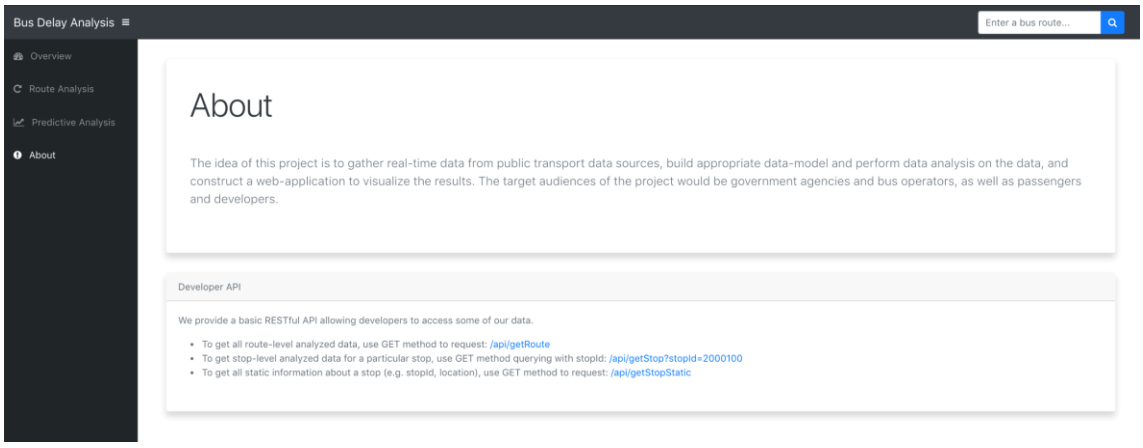

Figure 23: View of the about page

8. DISCUSSION

As mentioned in Methodology section of the report, our database schema had been designed and created prior to and independent of data warehousing module. The data warehousing module would request and retrieve schema from the database where tables had been created and schema defined. However, the common workflow for data warehousing would include the design of database. Specifically, the creation of tables should only before the data have been transformed and loaded into database. We chose not to follow the common paradigm because we wanted to separate design concerns so that we could have one individual specifically in charge of database design, who could make sure that our tables/entities conform to 3NF (Normal Forms) to avoid functional dependency issues such as update anomalies.

In data wrangling part, more than 41 million real-time update data need to be cleaned. Processing cost is always an important topic for big data. To optimize this query processes, we use a set of rules to transform the query-tree which including performing selection early, performing projection early and performing most restrictive selection and join operations before other similar operations. We compare two ways of query processes of one day dataset and analyze the processing cost. The following queries are aimed to select the latest record of the same stop in same trip, same day:

1. Normal nested query. The processing time: 12’08’’.

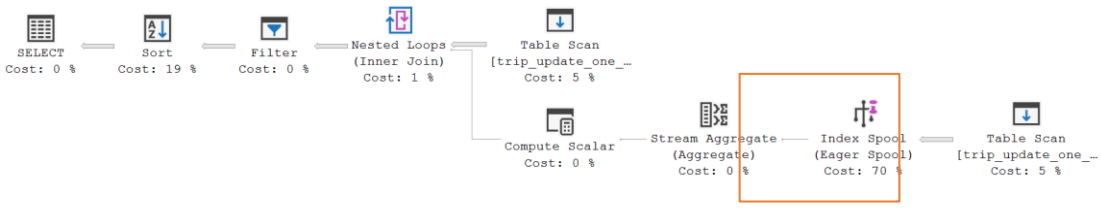

Figure 24: Process of normal nested queries

2. Applying OVER, PARTITION BY, ROW_NUMBER () function yields a shorter processing time of 4’54’’

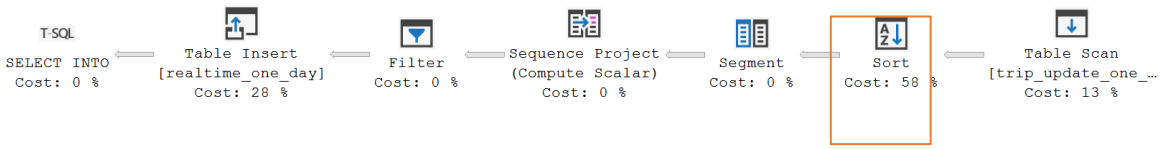

Figure 25: Process of queries using partitions

From the nested query execution plan, it is obvious that the query scans the real-time update table twice and spend 70% time on index spool. Although, it is a left semi join, the actual processing time is 12’08’’. However, the second query only scan the real-time update table only once. The part that takes the most time is sort which is 58% of the total cost. As a result, the second query is adopted in the final data wrangling.

According to the results of data analysis stated above, the overall performance of Sydney bus system is not as bad as people considered before the analysis conducted with not too many late routes based on the definition of on-time, early and late performance provided in the Methodology part. And the majority of the bus stops, trips and routes in Sydney are analyzed to be on-time in this project.

However, there are still some typical bus routes with high percentage of possibility for late performance such as X04 mentioned in the above section. Meanwhile, the delay time distributions for morning peak, evening peak and remained time period do not differ too much, which are not expected. This is due to most of buses are on time performance and the delay time mostly happens within the on-time section, which leading to the boxplot of delay time for three different time periods could not provide too much useful information. But this could be improved if only late and very late performance stops are analysed for morning peak, evening peak and other time period.

Through the study of the predictive model for future routes, the distance of routes, their location and the number of stops for routes would have important impacts on determining whether the route could be on-time or not. However, due to the limited information of the data source, this predictive classifier is only a very basic model. More information is necessary to be collected for expanding model's features to improve its accuracy.

From the point of view of government, they could study on factors, which are analysed in our project and found out to be having influence on the on-time performance of bus routes, to make their decision on the adjustment of bus routes, bus timetables, the ticket price and even the planning of new bus routes. From bus operators' perspective, these results could be a reminder for them to improve the bus services supplied through redressal on structure of staff and bus routes especially those heavily delayed ones like bus X04. From the perspective of passengers, they can have a better understanding of conditions of bus system in Sydney and the analysis results may have an impact on their choice to take buses, which would further affect the decision of government and bus operators on bus system.

Furthermore, it is expected by our group that our study results would give rise to the attention of the public on the bus delay problem in Sydney. Through this presentation of our study results, people could pay more attention to the condition of bus system in Sydney and the government could attach more importance to bus system adjustment.

In addition, unlike the majority of previous bus service studies, this project focuses more on the influence of relational factors, such as bus routes, bus stops, bus stop location as well as travel time and date, etc. on the on-time and delay rates. These analysis results can all contribute to identifying the real reasons of some bus delay in Sydney. Our study and analysis are more

detailed and multi-faceted than many existing studies for Sydney bus system and can affect the direction of further study to the aspect of prediction of future route performance.

In terms of web application development, it can be seen through the resulted website that we choose the present all our analysis results and routes information by charts or map on the webpages. This vivid and humanized form of presentation of information makes it more effective for users like the government officers or passengers to get an intuitional and distinct knowledge of the status of bus route system from our analysis results. And the developer API can supply users an alternative to obtain the dataset of analysis results directly instead of exploring the result of one specific route or stop by searching.

9. LIMITATIONS AND FUTURE WORKS

For prototyping purpose, we use free-trial service on Azure Cloud which put an upper limit to our database to 250GB. This translates to our decision to pull data at 15-minute intervals so that the size of data would not exceed the limit. However, the Realtime Trip Update dataset updates every 10 seconds. This means by retrieving at a frequency of 15 minutes, we have omitted a large portion of valid information, hence our data analysis is far from being accurate. To get the most accurate result and avoid possible noise and glitches in transmission, it is suggested that Nyquist rate should be used (Marks, 1991). Specifically, the data pulling should be pulled twice as fast as the original data updates, translating to an interval of 5 seconds.

As mentioned before, the data warehousing could pull one-session of data (approx. 200MB) under one minute, using a single process on Azure Webjob. If we are to use 12 processes running in parallel, the 5-second interval can be accomplished. Or a better solution would be to re-write the code to run 12+ threads in serial on a single process, as all threads sharing the same process would significantly reduce resource usage.

The data analysis could be expanded for some typical conditions, such as analysing the difference of number of delayed and very delayed stops for morning peak, evening peak and other time period and analysing Sydney bus system performance before and after the bus time schedule's adjustment, etc.

Furthermore, the predictive model only considers three features for data training. For our predictive model, only the data volume of one-day data is used to generate this model which is quite a small size for training effective models. The number of features for training data are also too few to give a high-quality predictive model. Therefore, for future research, more information about bus routes, such as population size near each bus stop and their age distribution and economic capability distribution, etc. can be explored and with more potential information more features for the training data could be mined to make the predictive model more complex and meaningful for the purpose of development of future bus routes.

Meanwhile the clients may also have interests on what a new bus route's on time rate is, which means the above predictive model is not functional enough to give more information. So future work could be done for predicting the detailed delay time or on time rate of a new route based on the collected information.

Moreover, if the complexity of the predictive model increases, the data size and number of data analysts also need to be augmented correspondingly for the improvement of prediction accuracy and efficiency to figure out the predictive model.

10. CONCLUSION

Unlike majority of the bus service study, our project focuses more on the influence of relational factors, such as bus routes, bus stops, bus stop location as well as travel time and date, on the on-time and delay rates. The analysis results contribute to identifying the real reasons of bus delay in Sydney whereas the web application provides a convenient and interactive means for the target audience to access the results. In all, our project is more detailed and multi-faceted than many existing studies for Sydney bus system and may provide useful insights for further studies.

REFERENCE

Back, K. (2002). *Test Driven Development: By Example.* Boston, Massachusetts, United States: Addison-Wesley Professional.

Bates, J., Polak, J., Jones, P., & Cook, A. (2001). THE VALUATION OF RELIABILITY FOR PERSONAL TRAVEL. Transportation Research Part E: Logistics and Transportation Review, 37 (2), pp. 191-229

Ceder, A. (2007). Public Transit Planning and Operation Theory, modelling and practice.

Crispin, L. (2008). *Agile Testing: A Practical Guide for Testers and Agile Teams.* London, United Kingdom: Pearson Education.

Davis, I. (2008). *What Are the Benefits of MVC?* Retrieved from Internet Alchemy: http://blog.iandavis.com/2008/12/what-are-the-benefits-of-mvc/

Zamudio, S.A., Santaolaya, R., & Fragoso, O.G. (2012). Restructuring Object-Oriented Frameworks to Model-View-Adapter Architecture. *IEEE Latin America Transactions, 10*, 2010-2016.

Evershed, N. (2017, May 23). Factcheck: just how bad are buses in Sydney's inner west? Retrieved 8 21, 2018, from The Guardian: https://www.theguardian.com/news/datablog/2017/may/23/factcheck-just-how-bad-are-buses-in-sydneys-inner-west

European Communities (2009, Sep 30). COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS. Retrieved 8 24, 2018, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52009DC0490&from=EN

Firew, T. (2016, Nov 28). Analysis of Service Reliability of Public Transportation in the Helsinki Capital Region: The Case of Bus Line 550.

Ma*, Z., Ferreira, L., & Mesbah, M. (2013, Oct). A Framework for the Development of Bus Service Reliability Measures. Brisbane, QLD 4072, Australia.

Nakanishi, Y. (2014). Bus Performance Indicators: On-Time Performance and Service Regularity. Transportation Research Record, 1571 (1).

NSW, T. f. (2013, Dec). SYDNEY'S BUS FUTURE. Retrieved 8 21, 2018, from Transport for NSW: https://www.transport.nsw.gov.au/sites/default/files/media/documents/2017/sydney-bus-future-final-web_0.pdf

O'Rourke, J. (2016, Sep 12). Bus commuters contact transport hotline to make thousands of complaints about 50 inner west bus routes. Retrieved 8 21, 2018, from Daily Telegraph: https://www.dailytelegraph.com.au/newslocal/inner-west/bus-commuters-contact-transport-hotline-to-make-thousands-of-complaints-about-50-inner-west-bus-routes/news-story/ebc8068422146c9ce7d41f8cf3cf4488

Polus, A. (1978, Aug). Modeling and measurements of bus service reliability. Transportation Research, 12 (4), pp. 253-256.

Skogen, E. D. (2014, Jun). On Implementations of Bus Travel Time Prediction Utilizing Methods in Artificial Intelligence. Retrieved 8 25, 2018, from https://brage.bibsys.no/xmlui/bitstream/handle/11250/253824/751710_FULLTEXT01.pdf?sequence=2&isAllowed=y

Sorratini, J. A., Liu, R., & Sinha, S. (2008, Jun 20). Assessing Bus Transport Reliability Using Micro-Simulation. Transportation Planning and Technology, 31 (3).

Leff, A., & Rayfield, J. T. (2001). Web-Application Development Using the Model/View/Controller Design Pattern. *IEEE Enterprise Distributed Object Computing Conference*, 118-127.

Marks, R. J. (1991). *Introduction to Shannon Sampling and Interpolation Theory* . New York: Springer-Verlag.

Transport for NSW. (2017). *General Transit Feed Specification - Timetable and Realtime Feed for NSW Buses Fileset Consumer Guide.* Sydney: Transport for NSW.