COSC2408

Programming Project 1

Semester 2, 2020

Capstone Project – Learning Analytics Visualisation

Name(s): Duncan Do, Labiba Islam, Fahim Tahmeed, Joel Jacob

Student Number(s): s3718718, s3694372, s3680881, s3660851

Contents Page

Introduction

In this Learning Analytics Visualisation project, an investigation was conducted on a dataset of parking events in the city of Melbourne (Victoria, Australia). This analysis was performed to garner insight into the trends of Melbourne's parking records. Such insights would then be used to answer a list of several business questions. These business questions directly relate a subsequent list of recommendations we will propose to the city of Melbourne. Such questions include:

- Which parking locations in Melbourne result in the most infringements?
- Which parking locations in Melbourne are most popular to park in?
- When is the rush hour for the parking locations in Melbourne?

The corresponding recommendations include:

- Recommend a place to park in relation to time
- Recommend a parking location to increase ticketer presence

These questions are what drove our investigations in the directions they went in. To answer these questions and subsequent recommendations, we had to investigate the past trends in the data as well as predict the future trends in the data. The former was done by analysing the existing parking records in the dataset and formulate visual representations of the parking trends in relation to the business questions (Violations, locations of parking events, times of parking events). This encapsulates the *Data Visualisation* segment of our project. The latter point was done by running the parking information we already possessed through multiple machine learning algorithms that would be able to learn from the trends of the data and predict the outcomes of future data (Outcomes being: whether a parking event was in violation, the location of a parking event, the time of a parking event). This segment of the project was dubbed, *Data Modelling*. The findings from both sections culminated into the recommendations to the city of Melbourne.

Methodology

Software Tools – A List of the software tools used throughout the project.

Language: Python. One of the most popular languages for data analysis. Its pre-built libraries do an excellent job in doing any sort of data analysis

IDE: Jupyter notebook. Running in the ipython environment, Jupyter notebook is the IDE of choice for all data analysts who use Python.

Libraries used:

- Data Wrangling: We used pandas and NumPy for data cleaning and wrangling
- Data Visualisation: We used mostly seaborn for our visualisations but some of the visualisations had matplotlib as well. We used seaborn styling on the matplotlib visualisations as well.
- Data Modelling: For data modelling we used the python sklearn library.

Version Control: For version control we used GitHub.

Data Cleaning – A breakdown of the methods used to prepare the data for analysis.

Loading the data

To make sure the data got loaded in correctly we used to function. Firstly, we used the data.head function to make sure all the columns loaded in. After that we used the data.tail function to make sure all the rows loaded in.

Data Types

To make sure all the data types were right we used the data.dtypes function. It shows the data types of each column.

Typos, Whitespaces, Upper/Lower Case

To check for typos, whitespaces, and cases we used the value_counts() function on every column with categorical values. The function groups together and counts all the same responses and displays it, making it easy to find any typos or case issues.

Whitespaces

To remove any extra blank spaces in the data we used the str.strip function for each column.

Upper/lower case

Once we found the values that had a case issue, we converted them all to uppercase using the str.upper function.

Sanity Checks (Outliers)

To check for outliers, we made a box plot, the visualization made it very easy to spot the outliers.

Missing Values

To find missing values we used the “isnull” function. It showed that there were no missing values present.

Data Cleaning – A breakdown of the important executive decisions made about preparing the data.

Changing arrival and departure times to date/time data type

This was the only decision made in relation to the datatypes of the fields. It was paramount that we changed these fields into date/time objects. This is because we plan to investigate the data in these fields using time-based functions. These include any descriptive or predictive analytics that care about the time of each parking record (E.g. Using the parking records to determine rush hours).

Neglecting the need for seconds in arrival and departure time

When converting the arrival and departure time fields into a date/time object; the converted data was stripped down to just the hours and minutes. The previously included seconds value had been removed. This however is an acceptable as we can view the duration of each parking record in seconds, as seen in the "Duration of Parking" field. Because of this field, we feel it is justified to have the arrival and departure times in just hours and minutes if it means we can utilize it as a date/time object.

Reason for putting all to uppercase

Majority of the string data entries were already all capitalized, except for a few. Instead of the cumbersome process of transforming the strings into proper capitalization, we felt it more suited to convert those few into all upper case to match the rest the data. All for the sake of uniformity of our string data.

Outliers

To deduce outliers in fields such as "Duration of Parking", a box plot was constructed to view said outliers. While we did find several entries deemed outliers, we cannot say for certain these are outliers. In a realistic scenario for fields like "Duration of Parking", it is not inconceivable for an individual to re-purchase a parking spot or even neglect the parking limit out of their own choice. Thus, unless outliers are some egregious value such as "parking duration for this record is 200 hours", we cannot deem these entries as outliers without further investigation. All the "outliers" found were too close to the main output that such "egregious" outliers do not exist.

Deduction that there are no typos

Throughout the data cleaning process, it was necessary to check the different inputs into each field (value counts function), in doing so for the string fields implicitly allows us to check for typos in the entries. If any string fields were yet to be checked by performing the other data cleaning techniques, they were actively checked after the fact. In doing all this and viewing all the inputs for all the string fields, we determined none were in error in relation to spelling or other typos.

Data Visualisation – An overview of the process of constructing the visualisations

General Process:

Each set of graphs constructed in *Data Visualization* each have their own personalised methodology in relation to the nature of the data they were presenting. Thus, specific breakdowns of the steps taken to produce said visualisations will be explored in the discussion aspect.

As a general walkthrough:

Extracting fields of importance from the dataset

To streamline the workflow when constructing the visualisations, the fields that are used in the construction are extracted into a separate data frame.

Manipulating the data frame

Transformations to the data frame are made to highlight the field relationship that is to be visualised later. This transformation could include:

- Refactoring the order of the records
- Eliminating records that do not fit the business question being visualised
- Etc.

Creating the visualisation

The prepared data frame is then piped into one of the graphs from the Matplotlib Python library. Where the graph parameters are tuned to create an accessible and legible visualisation.

Data Modelling – An overview of the universal steps taken when modelling the data

Classification Models used

All three business questions are classification type questions, thus multiple classification type models were used to see if we could predict the category/class of a new parking record (E.g. Class = in violation or not, Class = location of parking event). In total, 4 different classification models from the Scikit-Learn Python library; the specifics regarding each model will be explored in their respective discussion sections.

Important Note

Due to the nature of machine learning algorithms, the time to runs such algorithms on a large dataset grows exponentially. To avoid excessively (Weeks) long run times, a randomised slice of the dataset was used for each data modelling scenario. This was an executive decision made to allow for feasible run times while not compromising the integrity of the results (By using a reasonable sized slice of the dataset).

Each modelling process went through a series of steps to obtain the output:

Step 1: Identifying the target

Each classification model will take a series of fields as training data to predict some outcome. That outcome will be the value of another field in the dataset, dubbed the target field. That field needs to be isolated into its own variable.

Step 2: Identifying the features

The target field's value is predicted based on the models training using other fields as training data. At this point, any other field in the dataset that is not a unique identifier can be used as a feature to train the model.

Step 1a, 2a: Manipulating feature fields into acceptable forms

For both the target and feature fields; a classification model can only function with numerical values. Thus any field that has non-numeric data needs to be transformed. This can be done for all the string fields by looping through all the values in the field (value_counts()), and replacing every unique value with a unique integer (E.g. Every "Queens Street" becomes 0

and every “Spring Street” becomes 1). This maintains the integrity of the data while making it usable in the machine learning algorithm.
For this project, the only other field type was date-time, which can be easily transformed into numeric data by extracting each element of the date-time (E.g. hour) and passing that as a separate feature field.

Step 3: Parameter Tuning # 1

Each model has a collection of parameters, each with a range of values each can take. The model must be tweaked to use the optimal combination of parameter values to elicit the highest classification accuracy possible on the given dataset.
Grid Search from the Scikit-Learn Python library was used to loop through combinations of parameter values to find the most optimal combination on the dataset.

Step 4: Hill Climbing

Up until now, the models had been running on every viable feature field from the dataset, however not all those fields will have impact on the target field. A technique known as *Hill Climbing* is used to test the accuracy of the model with different combinations and amounts of feature fields used. The result of this technique will output the feature fields that have an impact on the target field, eliminating the redundant features.

Step 5: Parameter Tuning #2

With the set of important feature fields, the model’s parameters can be tuned once again for further accuracy. This is because now we can tune the model with only the important features rather than any and every feature obtainable.

Step 6: Training and testing the model

Now with both a concentrated set of features as well as a hyper tuned model, we are finally able to begin using the classification model. With only a finite amount of data to use, a subsection of the records is dedicated to training the model to learn what characteristics/values of a record result in the target categories/class.
The other subset of record is used to test the accuracy of the model. To see if the model can use its prior training to accurately predict the class of the record.

Step 7: Output results

Depending on the class type and other factors surrounding the business question, different results can be extracted from the model's performance. Universal ones would be accuracy ratings, confusion matrixes etc. (More details on output for each model explored in *discussion*).

Results

Data Visualisation – Streets with the highest amount (Percentage) of violations in each time (every 4 Hours)

(Author: Duncan Do)

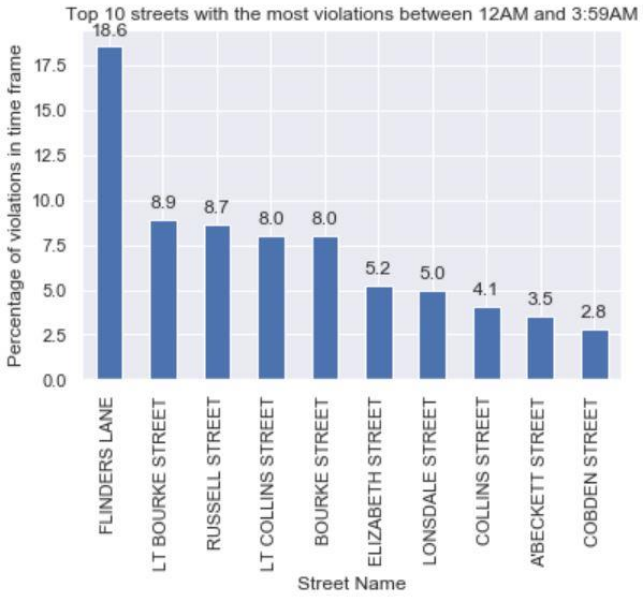


Figure [1] – Example visualisation of parking records in violation in a certain time slot

Multiple graphs were produced like figure 1 but of a different 4-hour time slot, relevant graphs will be placed in appendices and referred to in discussion

Data Visualisation – Most Popular Parking Street

(Author: Labiba Islam)

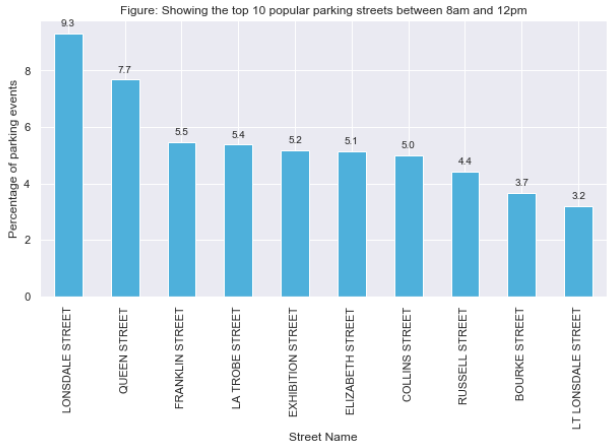


Figure [2] – Example visualisation of most popular parking street in a certain time interval

Multiple graphs were produced like figure 2 but of a different 4-hour time slot, relevant graphs will be placed in appendices and referred to in discussion

Data Visualisation – Most Popular Parking Area

(Author: Labiba Islam)

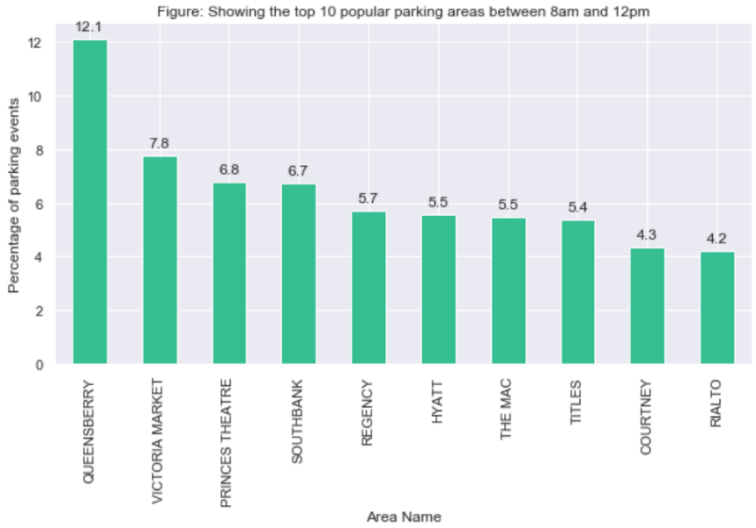


Figure [3] – Example visualisation of most popular parking street in a certain time interval

Multiple graphs were produced like figure 2 but of a different 4-hour time slot, relevant graphs will be placed in appendices and referred to in discussion

Data Visualisation – Frequency of Infringement for each month

(Author: Fahim Tahmeed)

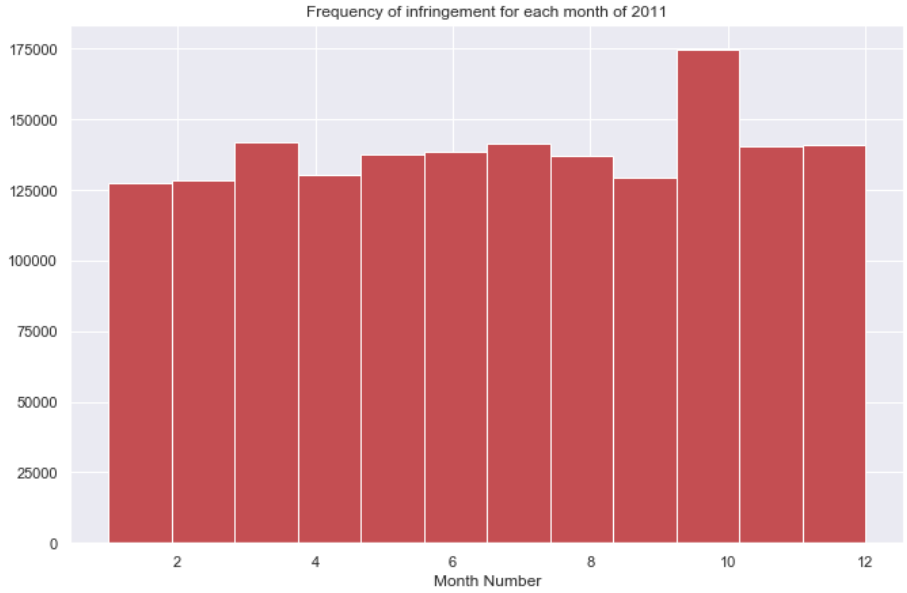


Figure [4] – Frequency of Infringement of each month

Data Visualisation – Percentage of infringement based on P based parking type

(Author: Fahim Tahmeed)

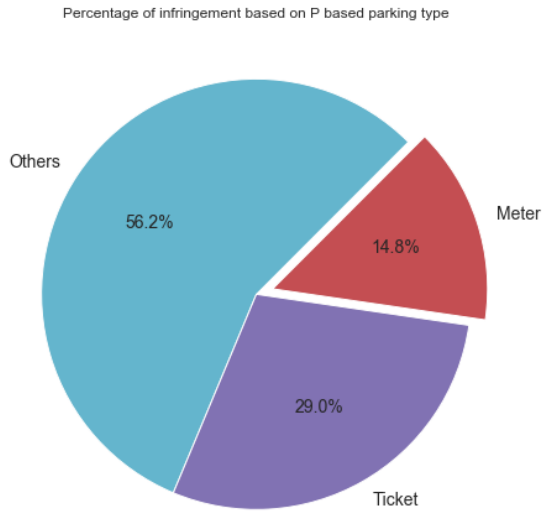


Figure [5] – Percentage of infringement based on P based parking type

Data Visualisation – Which is the busiest hour for each area

(Author: Joel Jacobs)

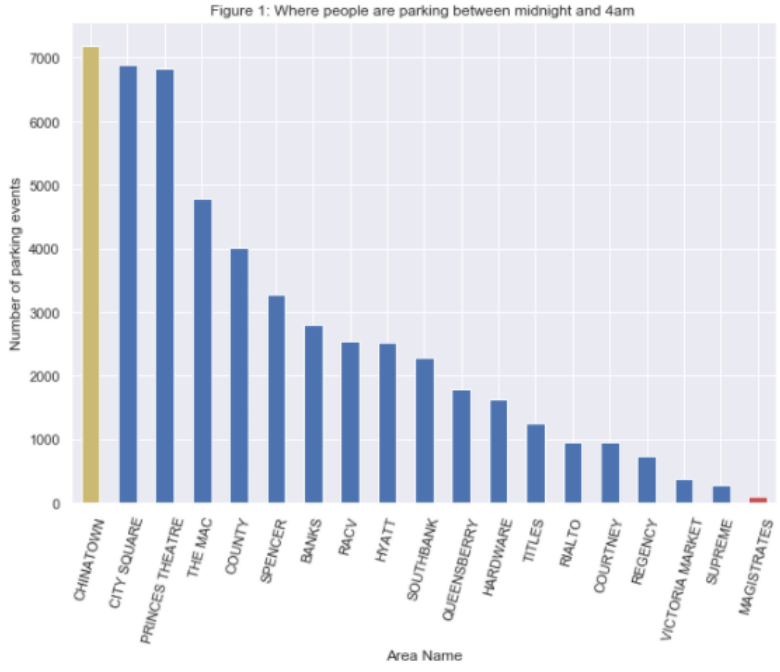


Figure [6]

Multiple graphs were produced like figure 5 but of a different 4-hour time slot, relevant graphs will be placed in appendices and referred to in discussion

Data Modelling – Predict a parking records violation status (Yes | No) – K-Nearest Neighbour classification model

(Author: Duncan Do)

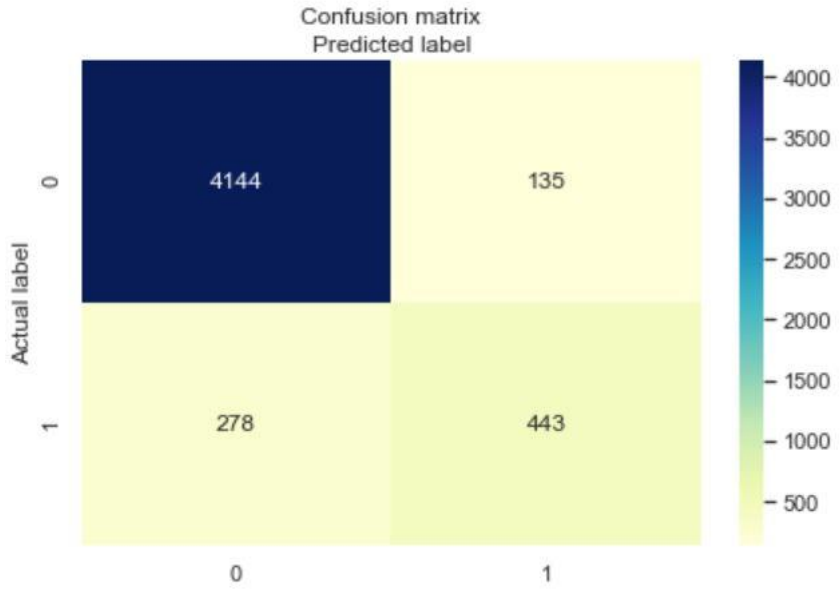


Figure [7] – Confusion Matrix of (Business Question: 1, Model: KNN)

Metric	**Score**
Accuracy Score	91.74%
Miscalculation Rate	8.26%
True-Positive Rate	61.44%
True-Negative Rate	96.85%
Precision	76.64%
F1-Score	68.21%
False-Positive Rate	3.16%
K-Folds Accuracy Score (5 Folds)	92.46%

Figure [8] – Calculated output data extracted from Confusion Matrix

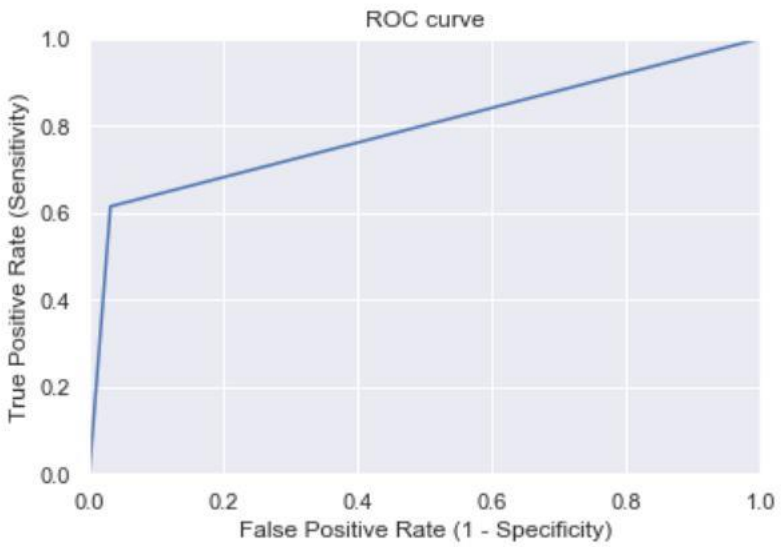


Figure [9] – ROC Curve for (Business Question 1, Model: KNN)

Data Modelling – Predict a parking records violation status (Yes | No) – Decision Tree classification model

(Author: Labiba Islam)

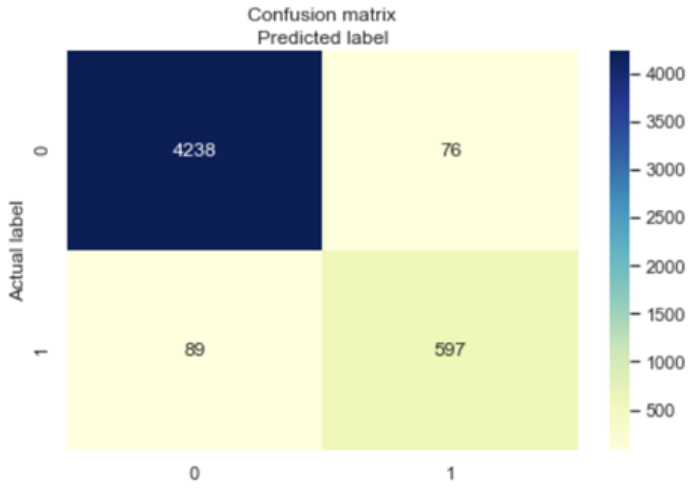


Figure [10] – Confusion Matrix of (Business Question: 1, Model: DT)

Metric	Score
Accuracy Score	96.7%
Miscalculation Rate	3.3%
True-Positive Rate	87.026%
True-Negative Rate	98.238%
Precision	88.707%
F1-Score	87.859%
False-Positive Rate	1.762%
K-Folds Accuracy Score (5 Folds)	86.97%

Figure [11] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records violation status (Yes | No) – Random Forrest classification model

(Author: Fahim Tahmeed)

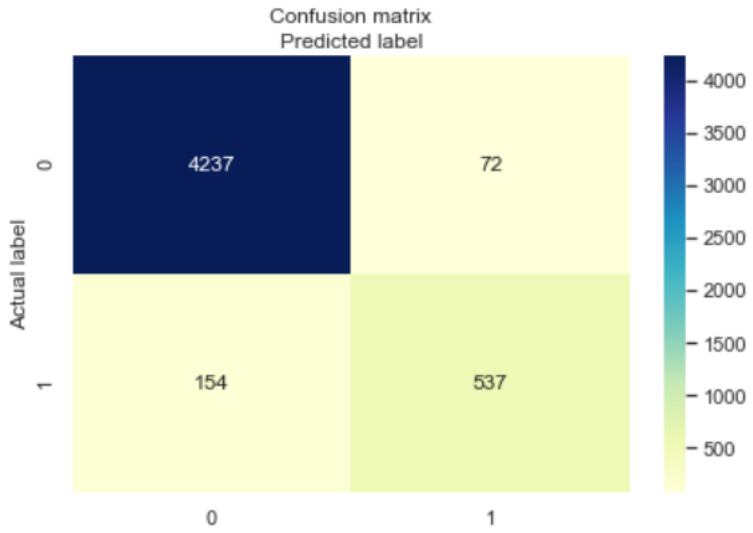


Figure [12] – Confusion Matrix of (Business Question: 1, Model: RF)

Metric	Score
Accuracy Score	95.48%
Miscalculation Rate	4.52%
True-Positive Rate	77.71%
True-Negative Rate	98.33%
Precision	88.18%
F1-Score	92.62%
False-Positive Rate	82.62%
K-Folds Accuracy Score (5 Folds)	1.67%

Figure [13] – Calculated output data extracted from Confusion Matrix

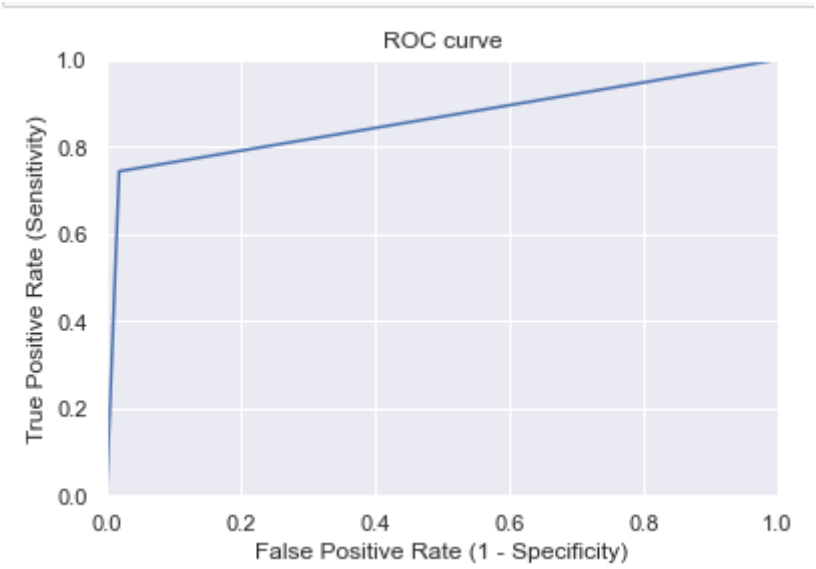


Figure [14] – ROC Curve for (Business Question 1, Model: RF)

Data Modelling – Predict a parking records violation status (Yes | No) – Support Vector Machine classification model

(Author: Joel Jacob)

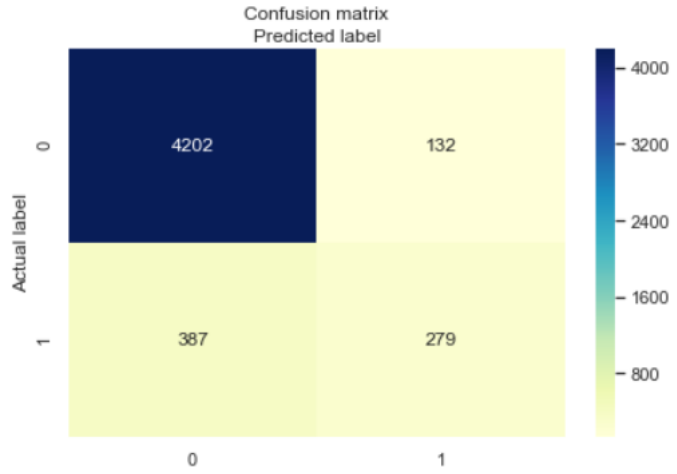


Figure [15] – Confusion Matrix of (Business Question: 1, Model: SVM)

Metric	**Score**
Accuracy Score	89.62%
Miscalculation Rate	10.38%
True-Positive Rate	41.89%
True-Negative Rate	96.95%
Precision	67.88%
F1-Score	51.81%
False-Positive Rate	3.1%
K-Folds Accuracy Score (5 Folds)	90.24%

Figure [16] – Calculated output data extracted from Confusion Matrix

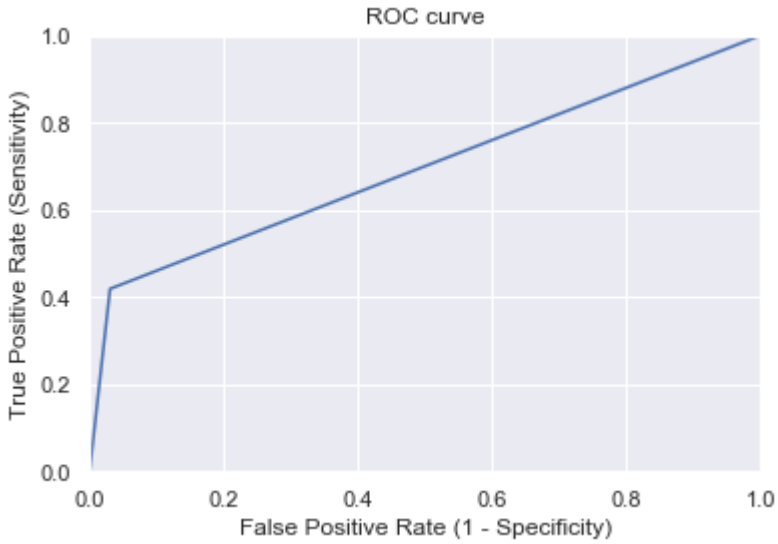


Figure [17] – ROC Curve for (Business Question 1, Model: SVM)

Data Modelling – Predict a parking records location (Street Name) – K-Nearest Neighbour classification model

(Author: Duncan Do)

```
Confusion Matrix
[[168   0   0 ...   0   0   0]
 [  0  25   0 ...   0   0   0]
 [  0   0 112 ...   0   0   0]
 ...
 [  0   0   0 ...   3   0   0]
 [  0   0   0 ...   0   2   0]
 [  0   0   0 ...   0   0   1]]

[Train/test split] score: 0.99120
Classification Report
```

Figure [18] – Confusion Matrix of (Business Question: 2, Model: KNN) – Visualisation of the Confusion Matrix is not legible, in appendices

Metric	**Score**
Accuracy Score	99.12%
Miscalculation Rate	0.88%
Precision (Weighted average)	99%
Recall (Weighted average)	99%
F1-Score (Weighted average)	99%
K-Folds Accuracy Score (5 Folds)	99.61%

Figure [19] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records location (Street Name) – Decision Tree classification model

(Author: Labiba Islam)

```
[[ 25   0   0 ...   0   0   0]
 [  0 106   0 ...   0   0   0]
 [  0   0  33 ...   0   0   0]
 ...
 [  0   0   0 ...   2   0   0]
 [  0   0   0 ...   0   0   1]
 [  0   0   0 ...   0   0   0]]
```

Figure [20] – Confusion Matrix of (Business Question: 2, Model: DT) – Visualisation of the Confusion Matrix is not legible, in appendices

Metric	**Score**
Accuracy Score	99.85%
Miscalculation Rate	0.15%
Precision (Weighted average)	100%
Recall (Weighted average)	100%
F1-Score (Weighted average)	100%
K-Folds Accuracy Score (5 Folds)	99.89%

Figure [21] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records location (Street Name) – Random Forrest classification model

(Author: Fahim Tahmeed)

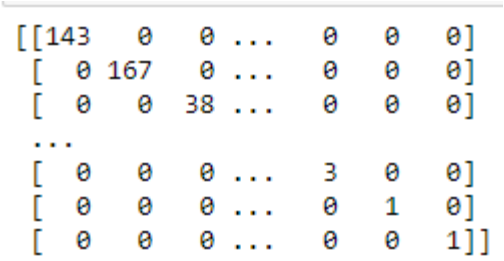


Figure [22] – Confusion Matrix of (Business Question: 2, Model: LR) – Visualisation of the Confusion Matrix is not legible, in appendices

Metric	**Score**
Accuracy Score	100%
Miscalculation Rate	0%
Precision (Weighted average)	100%
Recall (Weighted average)	100%
F1-Score (Weighted average)	100%
K-Folds Accuracy Score (5 Folds)	99.99%

Figure [23] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records location (Street Name) – Support Vector Machine

(Author: Joel Jacob)

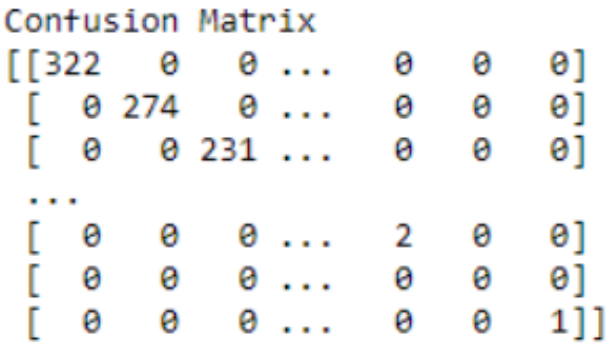


Figure [24] – Confusion Matrix of (Business Question: 2, Model: SVM) – Visualisation of the Confusion Matrix is not legible, in appendices

Metric	Score
Accuracy Score	99.86%
Miscalculation Rate	0.14%
Precision (Weighted average)	100%
Recall (Weighted average)	100%
F1-Score (Weighted average)	0%
K-Folds Accuracy Score (5 Folds)	99.95%

Figure [25] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records time (Arrival Hour) – K-Nearest Neighbour classification model

(Author: Duncan Do)

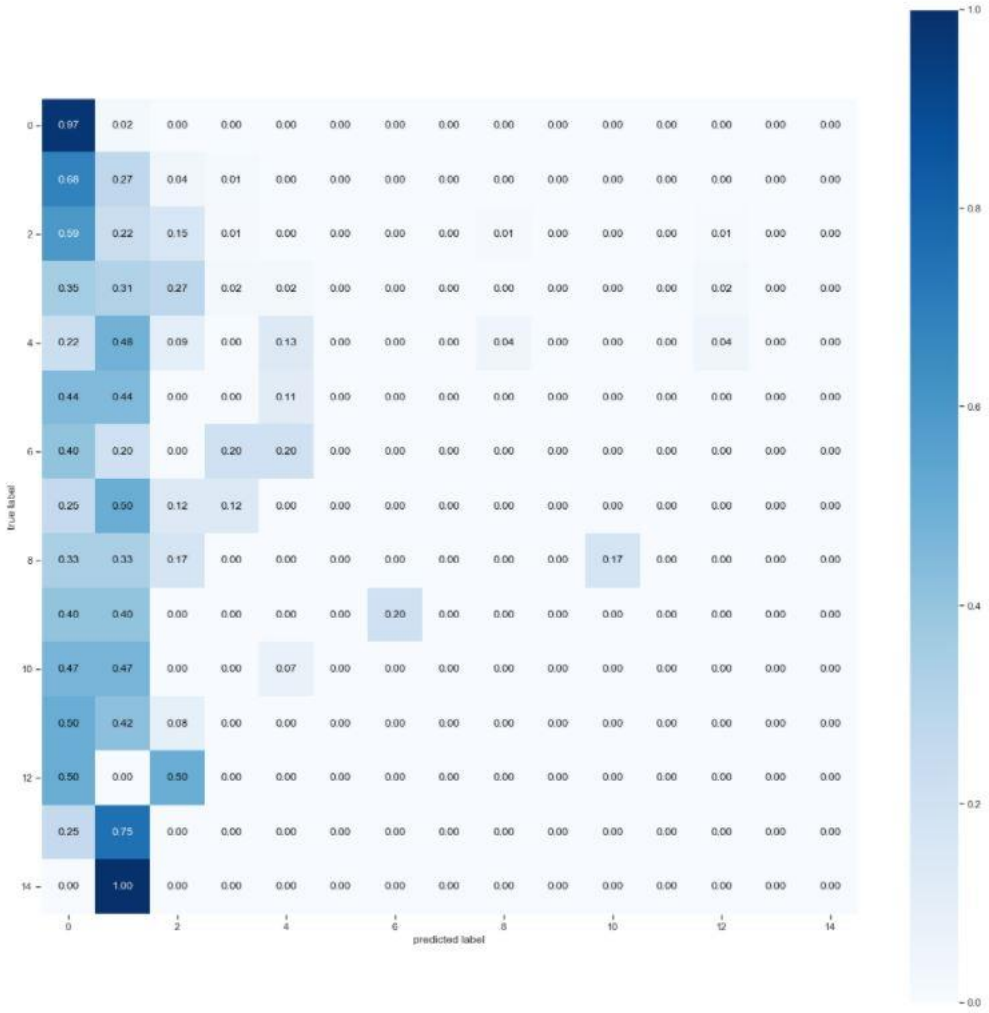


Figure [26] – Confusion Matrix of (Business Question: 3, Model: KNN)

Metric	**Score**
Accuracy Score	81.96%
Miscalculation Rate	18.04%
Precision (Weighted average)	77%
Recall (Weighted average)	82%
F1-Score (Weighted average)	79%
K-Folds Accuracy Score (5 Folds)	82.88%

Figure [27] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records time (Arrival Hour) – Decision Tree classification model

(Author: Labiba Islam)

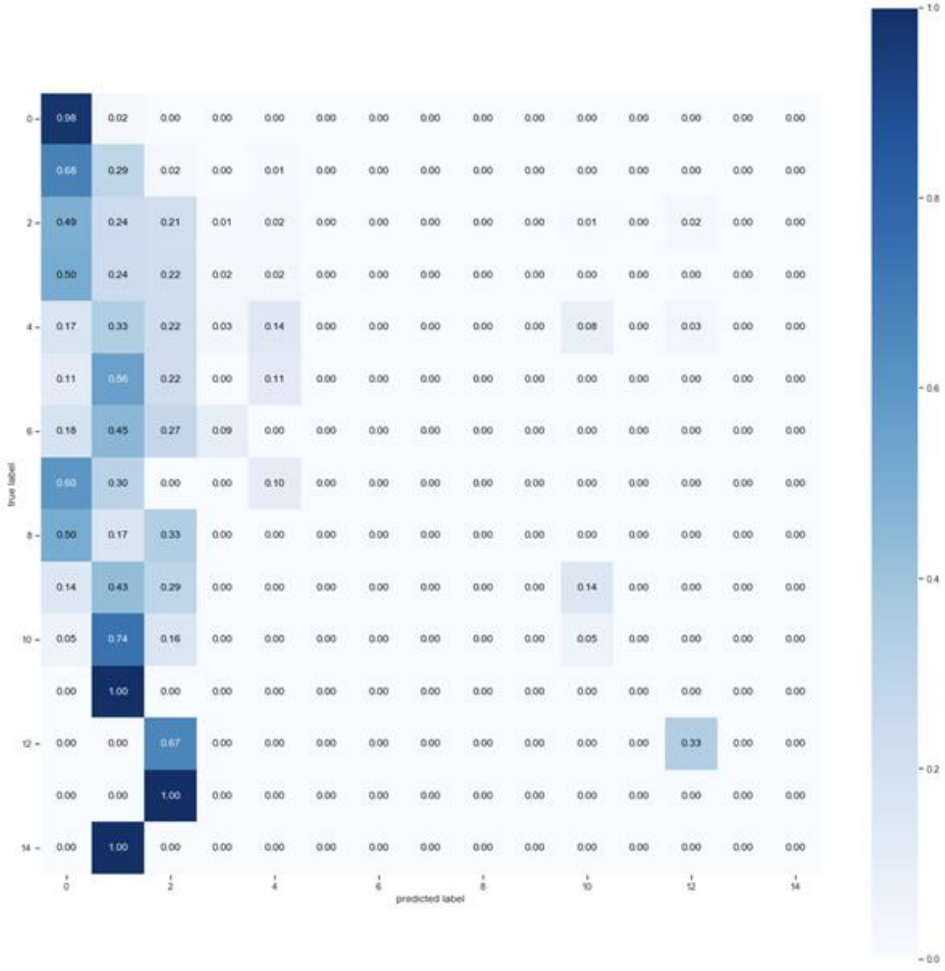


Figure [28] – Confusion Matrix of (Business Question: 3, Model: DT)

Metric	**Score**
Accuracy Score	82.36%
Miscalculation Rate	17.64%
Precision (Weighted average)	78%
Recall (Weighted average)	82%
F1-Score (Weighted average)	97%

Figure [29] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records time (Arrival Hour) – Random Forrest

(Author: Fahim Tahmeed)

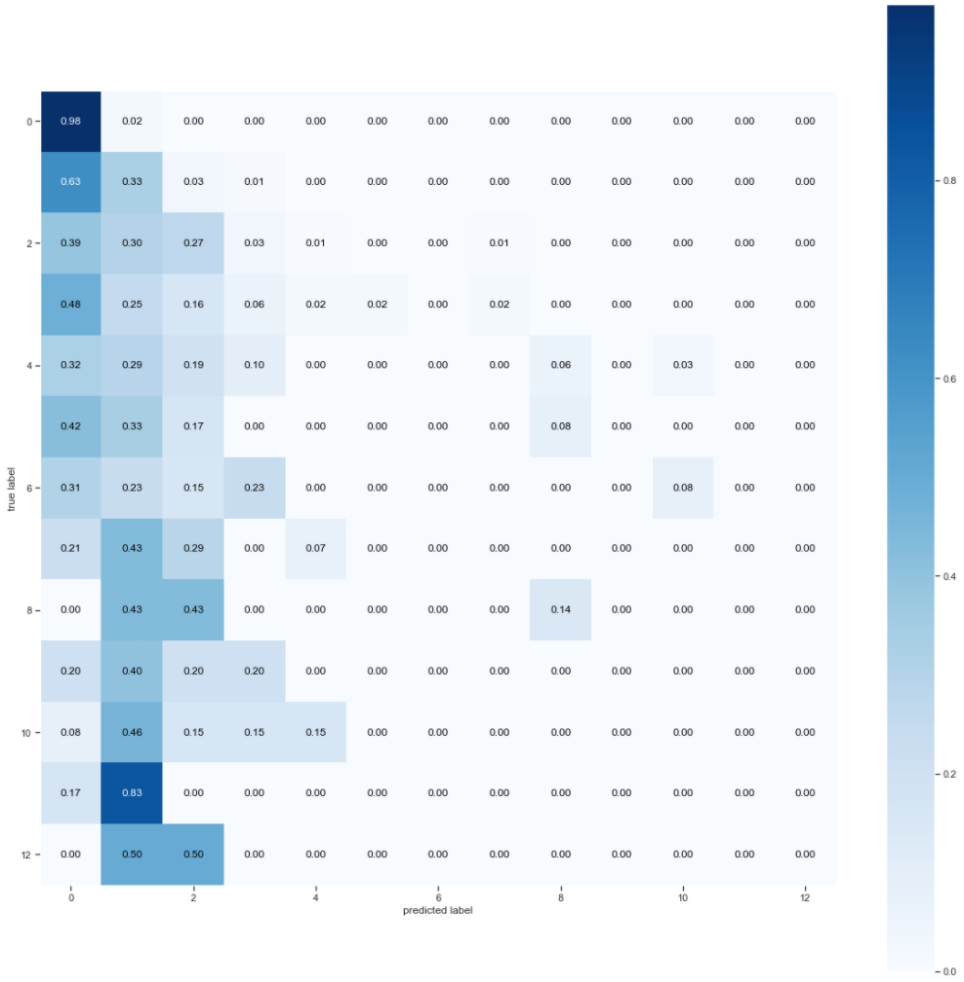


Figure [30] – Confusion Matrix of (Business Question: 3, Model: RF)

Metric	**Score**
Accuracy Score	83.7%
Miscalculation Rate	79%
Precision (Weighted average)	84%
Recall (Weighted average)	81%
F1-Score (Weighted average)	83.9%
K-Folds Accuracy Score (5 Folds)	83.8%

Figure [31] – Calculated output data extracted from Confusion Matrix

Data Modelling – Predict a parking records time (Arrival Hour) – Support Vector Machine

(Author: Joel Jacob)

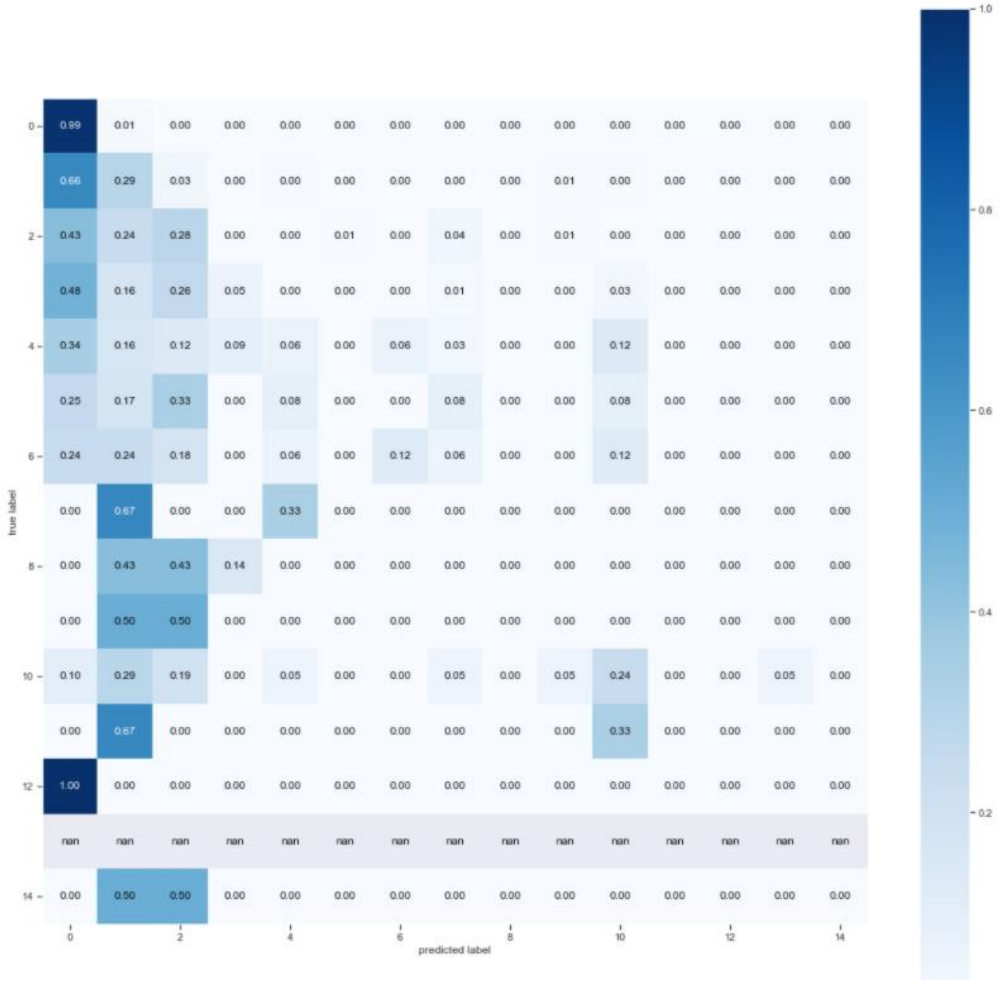


Figure [32] – Confusion Matrix of (Business Question: 3, Model: SVM)

Metric	**Score**
Accuracy Score	89.62%
Miscalculation Rate	10.38%
Precision (Weighted average)	87.9%
Recall (Weighted average)	41.89%
F1-Score (Weighted average)	96.95%
K-Folds Accuracy Score (5 Folds)	89.38%

Figure [33] – Calculated output data extracted from Confusion Matrix

Discussion

Data Visualisation – Streets with the highest amount (Percentage) of violations in each time (every 4 Hours)

(Author: Duncan Do)

The graphs for these visualisations were constructed by extracting a data frame from the initial dataset populated only by the parking records which have a value of 1 in their violation field (Meaning this parking record is in violation). The values that appeared in the *Street Name* field the most were ordered and the top 10 streets with the most violations were projected onto a bar graph. This was split to produce multiple graphs that each display the top 10 streets with the most violations within a 4-hour timespan. This was we can gauge if there are any changes to the "most violated" throughout the day.

In **figure [1]** in the **Results** section, Flinders Street owns a large percentage of the parking record violations for the first 4 hours of the day. This notion is then pushed aside as in further graphs in **Appendix [A]** show Flinders Street dropping from the top 10 in favour of streets such as Lonsdale Street, Exhibition Street, and Queens Street for most of the day. Flinders Street only regains a position in the top 10 (Top 1) by the concluding 4 hours of the day, as exhibited by the data on the last graph in **Appendix [A]**.

This behaviour could be attributed to the general location of these streets. As the streets that held most of the violations throughout the day are streets intertwine with the main grid of the City of Melbourne, the CBD. The times where these streets (E.g. Lonsdale Street) hold the most violations could be considered the "business hours" of the day. Thus, many citizens would enter these areas to work at businesses or to partake in businesses (E.g. Shop). As these CBD streets hold much of Melbourne's work and lifestyle capacities, it makes sense that many would park in or near this central point of the city during these times. Hence, the dominance in the violation graphs.

Data Visualisation – Most Popular Parking Street

(Author: Labiba Islam)

In reference to figure [2] and Appendix [C].

One of the interesting visualisations we decided to do is to find out 10 most popular streets where cars are being parked, particularly in the busy time of 8am to 12pm which is the time most people come to the city for office and other purposes. We decided to use a bar chart to visualise the data. From the bar chart we can see that Lonsdale street is the most popular street and LT Lonsdale street is the least popular. It is understandable as Lonsdale street has many street parking and a lot of offices are located around the street while LT Lonsdale street has limited parking spaces.

Data Visualisation – Most Popular Parking Area

(Author: Labiba Islam)

In reference to figure [3] and Appendix [C].

The figure above shows the top 10 busiest parking areas from 8am to 12pm in the morning. We can see that Queensberry is the busiest area. This could be because in this area street parking is available for 4 which is cheaper than the surrounding parking areas. Therefore, this area experienced a huge influx of parking events. Rialto has offices and hence it also had many parking events and stands 10th on the topmost busiest areas between 8am and 12pm.

Data Visualisation – Frequency of Infringement for each month

(Author: Fahim Tahmeed)

In reference to figure [4]

We decided to explore the frequency of violations each month for the year 2011. For that we chose a histogram diagram which will show the number of violations each month. As we can see from the diagram, the month where most infringement was given was October. The month that had the least number of infringements was January. Almost all the months had similar number of infringements and except the month of October, the trend was almost the same.

Data Visualisation – Percentage of infringement based on P based parking type

(Author: Fahim Tahmeed)

In reference to figure [5]

In the dataset, most of the infringements happened in P-based parking types. So we decided to explore a bit further and find out what type of P-based parking was causing the greatest number of infringements. So we separated the P-based parking infringements from the rest of the parking types. From the P-based parking types, most of the data were either Meter based, ticket based or other parking types which include clearways, no stop and disabled parking. We chose to use a pie chart to visualise the P-based parking infringements and we found that clearways, no stops and disabled parking were causing the most number of infringements, ticket based parking took the second place and meter based had the least number of infringements.

Data Visualisation – Which is the busiest hour for each area

(Author: Joel Jacobs)

In reference to figure [6]

To find the busiest area for the given time interval we explored the relationship between the 'Arrival time' and 'Area name' column. We used the 'datetime' function to isolate parking events that occurred between the desired interval. Once we did that we used 'value count' function on the Area name column to see how many parking events occurred for each area and used that data to plot a bar graphs.

In reference to Appendix [E]

Midnight to 4 am

From the graph we can see that for this time slot China town is the busiest area to park your car for the time slot followed by City square and Princes Theatre. The Magistrates has the least amount of people parking there for this time slot.

4am to 8am

From the graph we can see that for this time slot Queensberry is the busiest area to park your car for the time slot followed by Southbank and Victoria. The Magistrates has the least amount of people parking there for this time slot.

8am to 12pm

From the graph we can see that for this time slot Queensberry is the busiest area to park your car for the time slot followed by Victoria market and Princes theatre. The West Melbourne parking spot has the least amount of people parking there for this time slot.

12pm to 4pm

From the graph we can see that for this time slot Queensberry is the busiest area to park your car for the time slot followed by Princes theatre and Victoria market. The West Melbourne parking spot has the least amount of people parking there for this time slot.

4pm to 8pm

From the graph we can see that for this time slot Princes theatre is the busiest area to park your car for the time slot followed by Regency and The Mac. The West Melbourne parking spot has the least amount of people parking there for this time slot.

8pm to Midnight

From the graph we can see that for this time slot The Mac is the busiest area to park your car for the time slot followed by Regency and Princes theatre. The Tavistock parking spot has the least amount of people parking there for this time slot.

Most popular spot (any given time)

From the graph we can see that Queensberry is the most popular parking spot by a very big margin and would be very hard to find a parking spot there.

Data Modelling – Predict a parking records violation status (Yes | No) – K-Nearest Neighbour classification model

(Author: Duncan Do)

When using the K-Nearest Neighbour (KNN) classification model to predict whether a parking record is in violation or not, the model returned an accuracy rating of 91.74% (**figure [8]**). Meaning over 9 times out of 10, the model was able to correctly predict whether a parking record, based on its other features, was in violation or not. The ROC Curve (**figure [9]**) displays that the results of this KNN model are well above the standards of a random classifier (A straight diagonal line on the x = y) Elevated shape of the graph indicates that the predictions of the model were well founded in prior knowledge provided by the training data and the model tuning, rather than being random guesses that got lucky.

True Positive Score and True Negative Score

It was found that when the outputted class was expected to be "In Violation, the KNN model was correct 61.44% of the time (**figure [8]**). This is strikingly low given the overall accuracy. This is reflected in the confusion matrix, **figure [7].** In this heatmap, the bottom right square, which represents the amount of times the model correctly predicted a parking record was "In Violation" (True Positive), is noticeably low compared to the top left square, which represents the amount of times the model correctly predicted "NOT In Violation"(True Negative). This may be due the lack of representation of "In Violation" parking records in the sample slice of the dataset used for this model. Regardless, the model received enough training for "NOT in Violation" to produce a well above average accuracy rating.

Precision

While in comparison to the True Negative Score, the True Positive Score is lacking. When looking at True Positive in isolation; out of the total amount of predicted positive (In Violation = yes) observations, the KNN model was able to have a correct ratio of 76.64% (**figure [8]**). While still not as impressive as the True Negative score of 96.85 (**figure [8]**), it still proves that the model was able to precisely predict the "In Violation" outcome most of the time.

K-Folds Accuracy Boost

These results analysed thus far have been from rigid training structure of ALL TRAINING → ALL TESTING. To further boost the potential accuracy score of the KNN model, K-Folds cross validation was implemented. This runs the training-testing procedure multiple times

with the testing segment occurring at different points during the training data. These "folds" where the testing data is displaced are each their own environment condition for the model to gain experience and learn from. This has proven marginally successful, with the K-Folds method increasing the accuracy score by 0.72% (92.46% - **figure [8]**). In the scope of machine learning algorithms, this is a noticeable increase given only 5 folds were implemented, meaning only 5 different testing environments were used. With further investigation to additional folds and repetition type testing, the accuracy of KNN on this business question can improve.

Data Modelling – Predict a parking records violation status (Yes | No) – Decision Tree classification model

(Author: Labiba Islam)

The decision tree classifier to predict whether a parking event is in violation or not has an accuracy of 96.7%. The true positive rate is around 87% and the true negative rate is around 98%. The true negative rate is 98% which means the model predicted correctly 98% of the time when an infringement occurred. The precision, which is the percentage of true positives over actual results, is 88.708%. The false-positive rate is low, which makes the model for prediction a good one. The F1 score is around 0.88 which is also high. The greater the F1 score, the better. The results analysed above are outcomes after the training and testing with 5-folds cross validation whose mean accuracy score is around 87%.

Data Modelling – Predict a parking records violation status (Yes | No) – Random Forrest

(Author: Fahim Tahmeed)

Random Forest performed particularly well regarding predicting whether a particular event will result in an infringement or not. For the question, the model received an accuracy score of 95.4% and we were able to increase it using k-folds slightly to 95.8%. That is a very good score for a predictive model. The model also has a high true-positive score of 77.7% and true-negative score of 98.3% meaning the model predicted correctly when an infringement occurred 77.7% of time and it did not occur 98.3% of time. Naturally, the false-positive score was only 1.6%. The precision and f-score were also quite high at 88.1% and 82.6% respectively. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations and F1 Score is the weighted average of Precision and

Recall. Therefore, this score takes both false positives and false negatives into account. The higher the precision and f1-score, the better. As we have a high accuracy, precision, and f1-score for this question we can say that Random Forest performed well with regards to tackling this question. The reason that we could think of why the random forest performed so well is that random forest is supposed to work well with imbalanced data. The sample that we worked with for the fitting of the Random Forest model was imbalanced as it had way more non violations than violations. As a result the random forest performed so well in predicting violations.

Data Modelling – Predict a parking records violation status (Yes | No) –Support Vector Machine

(Author: Joel Jacob)

support-vector machines performed well in predicting if an event is an infringement or not. For this question, the SVM classifier got a score of 89.62% which we were able to slightly increase using the k folds technique to a 90.24% which is a pretty good score for the predictive model. The classifier has a very high True negative rate which means that the model was very good at predicting when it was not an infringement meaning that model was highly specific. However, the classifier did not do too well in True positive rate only receiving a 41.892% meaning that it was not too good at predicting when it was not an infringement. The f-score and precision are 51.81% and 67.83%. which Is not ideal as it is better to have a higher f-score and precision for accuracy, SVM did not really perform too well in this case. SVM generally doesn't perform too well with imbalanced data as with an increase in data imbalance, the ratio between positive and negative support vectors become more imbalanced; therefore, samples at the boundary of hyper place are more likely to be classified as negative. The classifier still did not do too bad, but it could have been more accurate.

Data Modelling – Predict a parking records location (Street name) – K-Nearest Neighbour classification model

(Author: Duncan Do)

When using the K-Nearest Neighbour (KNN) classification model to predict what street the parking record is on, the model returned an accuracy rating of 99.12%. Meaning almost 10 times out of 10, the model was able to correctly predict the location (street) of a parking record, based on its other features.

Confusion Matrix Visualisation → Classification Report

This business question does not have a heatmap visualisation for the confusion matrix presented. This is due to the high number of classes that make that visualisation illegible. The accuracy score and subsequent classification report extracted from the confusion matrix provide enough insight into the confusion matrix's behaviour that this is not a detriment to the analysis. (**appendix [C]** holds the original matrix)

Classification Report: Precision, Recall and F1 Score

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observation to all the observations in the class in question. Both of which are at 99% (**figure [19]**] for KNN with this business question. This means that regardless whether you look at the scope of correct predictions in positive observations or all observations, the model was able to be trained well enough to predict correctly near 100% of the time. The F1-Score (**figure [19]**) is the weighted average of the 2 metrics and is debatable a more useful metric for imbalanced data. From what can be seen from **figure [18]**, there is a clear weighting to the first few classes in this model than the last few, that kind of trend would bring light to why the F1-Score is as high as the accuracy score.

K-Folds Accuracy Boost

These results analysed thus far have been from rigid training structure of ALL TRAINING → ALL TESTING. To further boost the potential accuracy score of the KNN model, K-Folds cross validation was implemented. This runs the training-testing procedure multiple times with the testing segment occurring at different points during the training data. These "folds" where the testing data is displaced are each their own environment condition for the model to gain experience and learn from. This has proven marginally successful, with the K-Folds method increasing the accuracy score by 0.49% (99.61% - **figure [19]**). In the scope of machine learning algorithms, this is a noticeable increase given only 5 folds were

implemented, meaning only 5 different testing environments were used. With further investigation to additional folds and repetition type testing, the accuracy of KNN on this business question can improve.

Data Modelling – Predict a parking records location (Street Name) – Decision Tree classification model

(Author: Labiba Islam)

To predict the street name of a parking event the decision tree classifier has an accuracy of 99.85%, which is very high. The precision, recall and f1-score are all 100%, meaning that the model can predict the location accurately almost in all cases. The data was trained and tested with 5-folds cross validation and the score was 99.89%.

Data Modelling – Predict a parking records location (Street name) – Random Forrest

(Author: Fahim Tahmeed)

We also tried to predict the location of a particular parking event using Random Forest. This is the question where random forest had a near perfect accuracy of 100%. However, after running the k-folds cross validation the accuracy did lessen a bit to 99.9% which is still amazing when it comes to tackling this question. We again chose weighted average score for precision, recall and F1-Score and we got 100% score for those as well. Again the classes were imbalanced, and the random forest did well in predicting the location.

Data Modelling – Predict a parking records location (Street Name) – Support Vector Model

(Author: Joel Jacob)

Sample Text

Data Modelling – Predict a parking records time (Arrival Hour) – K-Nearest Neighbour classification model

(Author: Duncan Do)

When using the K-Nearest Neighbour (KNN) classification model to predict what street the parking record is on, the model returned an accuracy rating of 81.96%. Meaning 8 times out of 10, the model was able to correctly predict the time (arrival hour) of a parking record, based on its other features.

Data Imbalance

Like *Business Question: 2, Model: KNN*, the dataset slice used for this instance is unbalanced. As seen on the confusion matrix (**figure [26]**) The earlier classes have actual results on their True Positive Scores and array of False Positive Scores. While beyond class 4 all the way to class 14, there are 0 True Positive Scores. However, this is not due to the model's inability to classify those outcomes. Its due to the dataset slice not having data of those tail end classes in the model testing. Because of this imbalance, when obtaining Classification Report (**figure [27])** data such as Precision, Recall, and F1-Score; the weighting average was used over the macro average. This is because the weighting average identifies the 0s in the matrix as non-entries rather than fails.

Once we obtained those weighted averages, we still resulted in fairy high results regarding the True Positive's of the KNN Model. ((**figure [27]**) Precision = 77%, Recall = 82%). We can extrapolate from these metrics that despite not being properly tested-trained on a large collection of the target's classes, the model was able to maintain above average accuracy.

K-Folds Accuracy Boost

These results analysed thus far have been from rigid training structure of ALL TRAINING → ALL TESTING. To further boost the potential accuracy score of the KNN model, K-Folds cross validation was implemented. This runs the training-testing procedure multiple times with the testing segment occurring at different points during the training data. These "folds" where the testing data is displaced are each their own environment condition for the model to gain experience and learn from. This has proven marginally successful, with the K-Folds method increasing the accuracy score by 0.92% (82.88% - **figure [27]**). In the scope of machine learning algorithms, this is a noticeable increase given only 5 folds were implemented, meaning only 5 different testing environments were used. With further investigation to additional folds and repetition type testing, the accuracy of KNN on this business question can improve.

Data Modelling – Predict a parking records location (Street name) – Decision Tree classification model

(Author: Labiba Islam)

The accuracy of a decision tree classifier to predict the duration of a parking event is 82.36%. The classification report was generated from sklearn was used to find out precision, recall and F1-score. The precision is 78%, recall 82% and F1 score is 0.79. These values are particularly not great as they are all below 90%. The mean score of K-folds cross validation with 5 folds is around 97% which is high enough. However, the other metrics for the model imply that the model is not good enough to predict the duration of a parking event with high accuracy.

Data Modelling – Predict a parking records location (Street name) – Random Forrest classification model

(Author: Fahim Tahmeed)

Regarding predicting the duration of a car parking, Random Forest again did well but as good as predicting whether an event resulted in violation. It does have a good accuracy score at 83.7%. And we increased it slightly to 83.9% after running k-folds cross validation on it. We chose to use the classification report generated by the sklearn.classification_report function and took the weighted average score of precision, recall and F1-Score. The reason we took a weighted average score was because a lot of our classes for this question had 0.0 on the confusion matrix. However that is not an accurate description of our data. Weighted average score considers the null values that the classes give out and returns a score based on that as well. From the classification report, we can see that the precision, recall and F1-score is 79%, 84% and 81% respectively which are all very good scores for this model.

Data Modelling – Predict a parking records location (Street name) – Support Vector Model classification model

(Author: Joel Jacob)

Sample Text

Conclusion

Based on the results and discussion we have come up with the comparison below of our models and we also came up with some prescriptive analysis. The comparison table is given below:

Research Question	Model name	Accuracy score
Q1	K nearest neighbour	91.75%
	Decision-tree	96.7%
	Support Vector Machine	89.62%
	Random Forest	95.4%
Q2	K nearest neighbour	99.12%
	Decision-tree	99.85%
	Support Vector Machine	99.86
	Random Forest	100%
Q3	K nearest neighbour	81.96%
	Decision-tree	82.36%
	Support Vector Machine	84.1%
	Random Forest	83.7%

From the above comparison, we can see that when predicting Q1 or whether a particular event will result in an infringement or not, Decision-tree performed the best and Support Vector Machine was the worst performing model. To predict the location of a parking event, all the models performed incredibly well. But the standout was Random Forest with 100% accuracy achieved in predicting the location. But all the models had over 99% accuracy. For

the final question which was to predict the duration of a particular parking event, none of the models got a very high score of 90% and most of it was early 80%. Support Vector Machine has the highest score at 84.1% while K-nearest neighbour had the lowest score at 81.96%. So, even though some models performed better than others ever so slightly, overall all our models performed very well regarding the different research questions that we set out at the beginning of the project.

Data Recommendation – Recommend a place to park in relation to time

(Author: Joel Jacob)

In relation to Appendix [E]

To recommend place to park for a particular time of the day we used the sum of the total number of parking events for each area for time and compared it with each other. We grouped the times in 4hrs intervals. The recommendations are

Midnight-4am: After analysing figure 6 we would recommend parking at magistrates parking spot for this time frame as it has significantly lower amount of people parking there compared to the other parking spots for this time.

4am-8pm: After analysing figure 2-5 in appendix E we would recommend parking at the West Melbourne parking spot for this time. West Melbourne has a very low amount of parking events for this time.

8pm-Midnight: After analysing figure 6 in appendix E we would recommend parking at Tavisstock parking space as it has the least

When the parking events did not come up for a certain time, we assumed that the parking spot was closed for the time. West Melbourne came up as the most common recommendation as it was the least popular parking space overall as we can observe in figure 7 in appendix E.

Data Recommendation – Recommend a parking location to increase ticketer presence

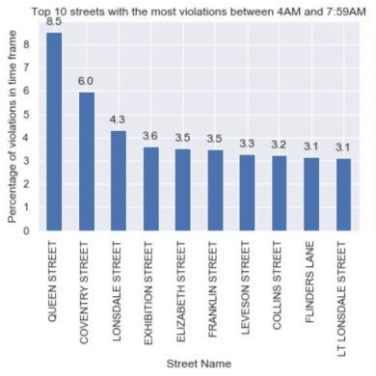
(Author: Labiba Islam)

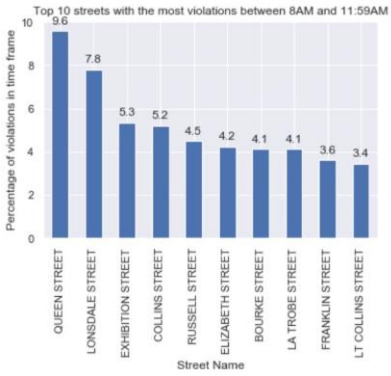
From all the visualisations in Appendix B, Queensberry, Southbank, Spencer, Victoria Market, Rialto are area names that come in the top 10 within the four intervals. In 3 out of 6 visualisations, Queensberry comes within the top 2 of the areas with the greatest number of infringements followed by Southbank. Therefore, Queensberry and Southbank are areas that need more ticket officer presence.

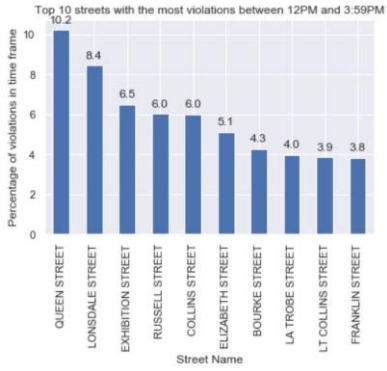
From the visualisations in Appendix A, the street names that appear the most are Lt Collins Street, Bourke Street, Lonsdale street and Collins Street. Some more of these streets are Flinders lane, Bourke street, Elizabeth street, Franklin street and Queen street. These streets have high numbers of events with violations. Hence, these streets need more ticket officer presence.

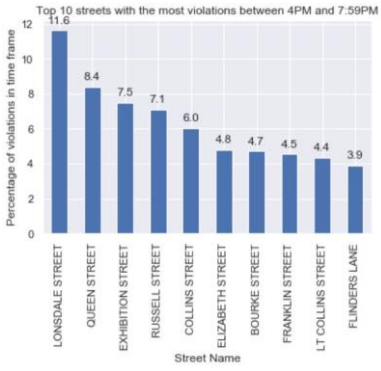
Appendices

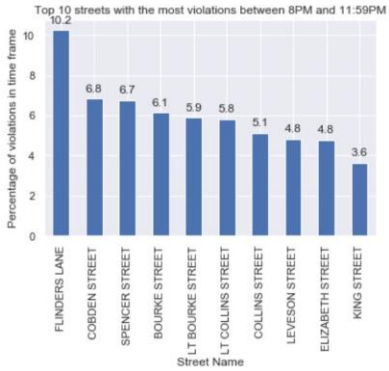
- Appendix [A] – Remaining graphs for *Streets with the highest amount (Percentage) of violations in each time (every 4 Hours)*



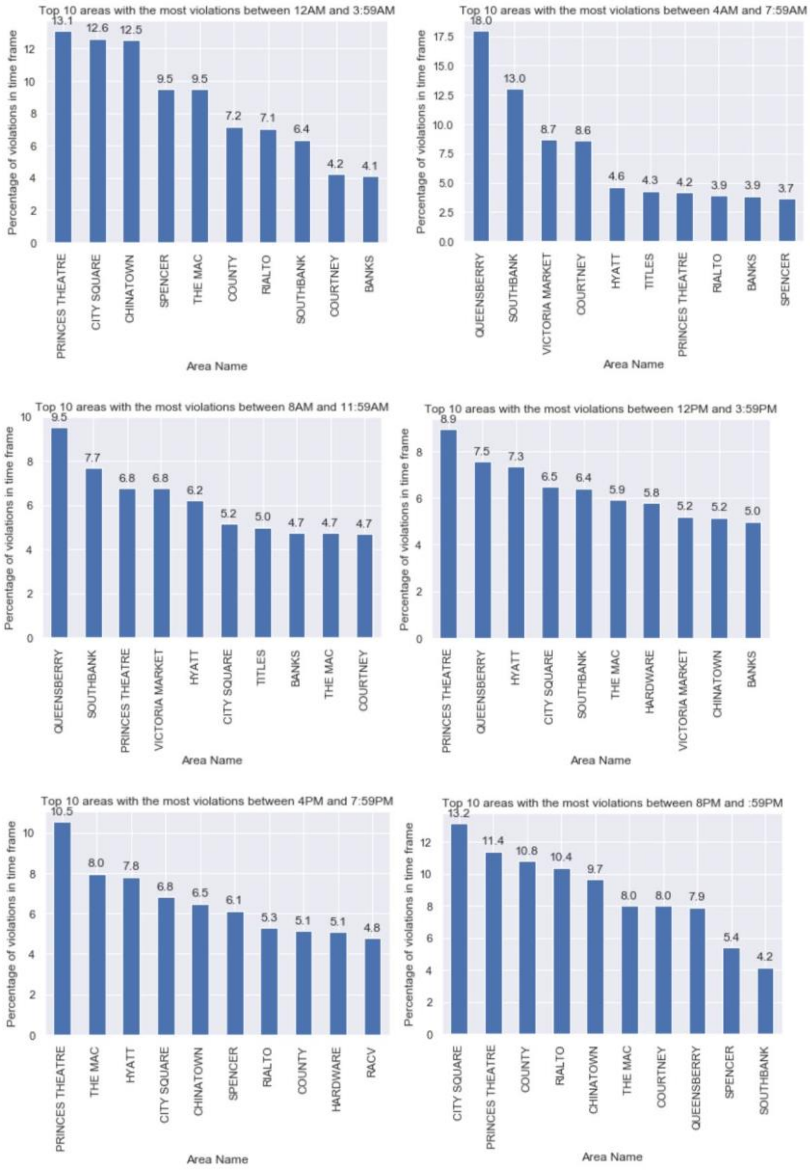




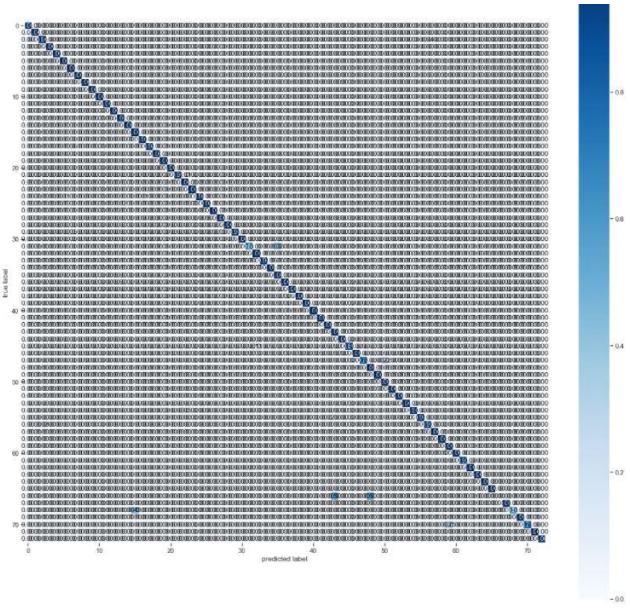




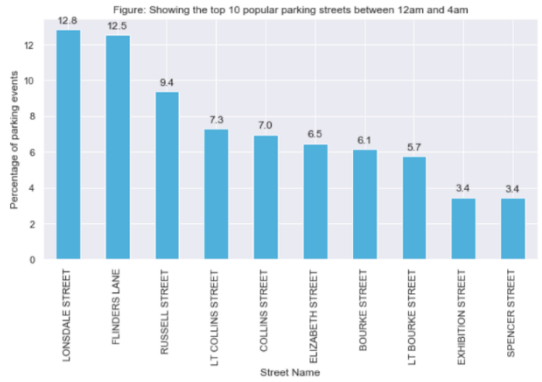
- Appendix [B] – Other graphs for ***AREAS*** *with the highest amount (Percentage) of violations in each time (every 4 Hours)*

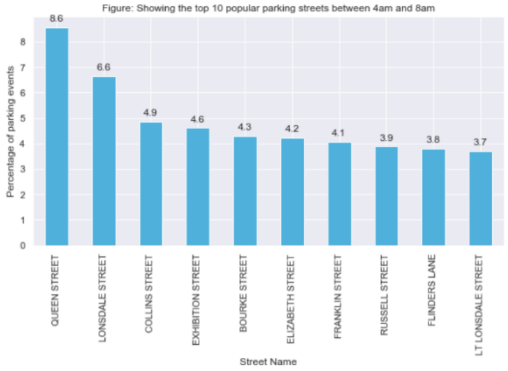


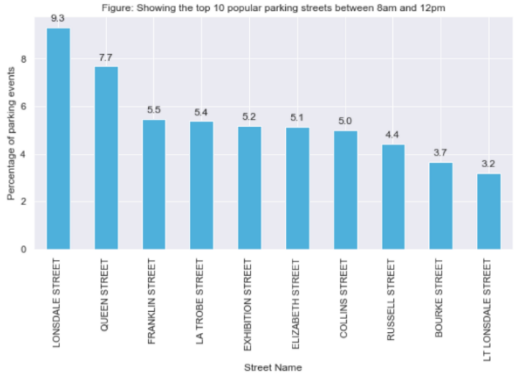
- Appendix [C] – Business Question: 2, Model: KNN – Confusion Matrix Heatmap

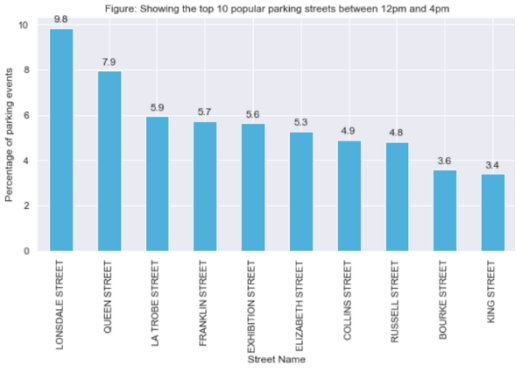


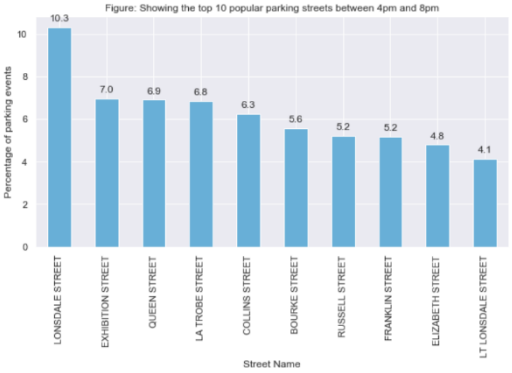
- Appendix [D] - Remaining graphs for *Most Popular Parking Spots*

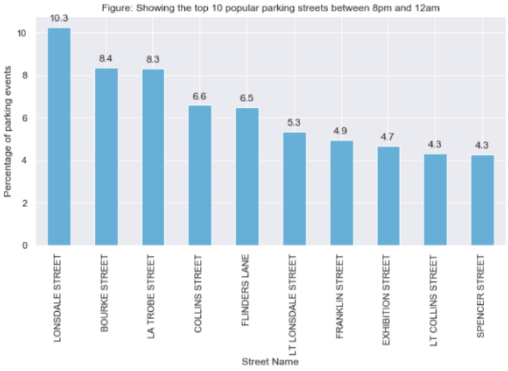


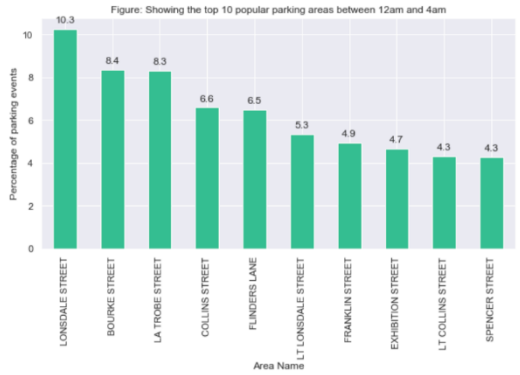


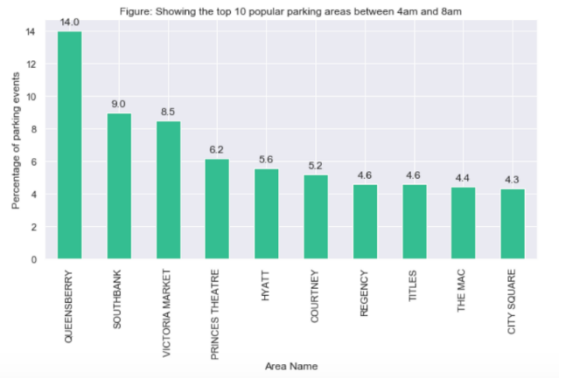


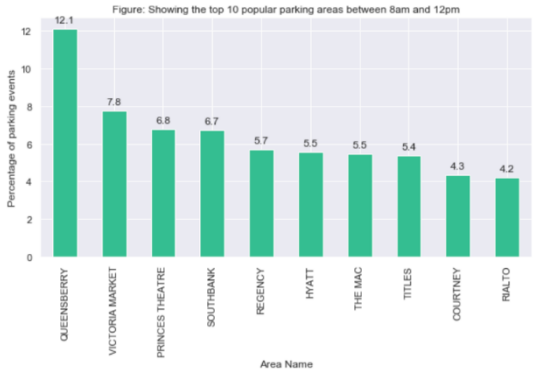


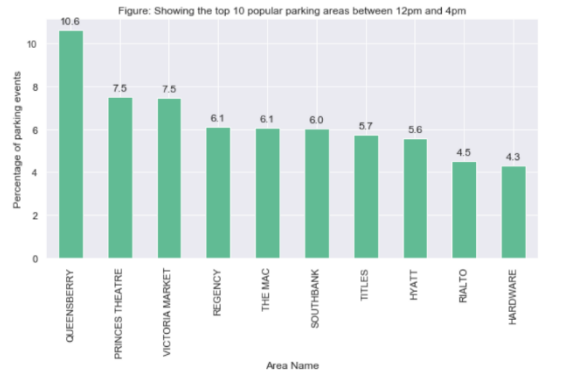


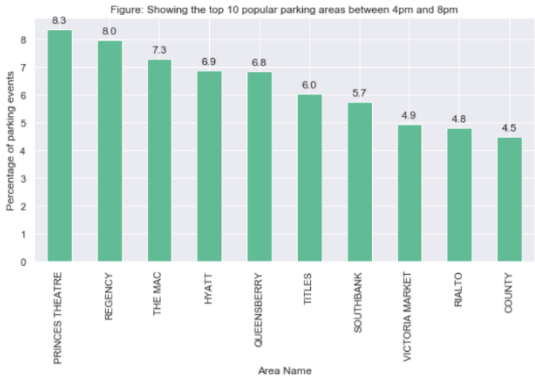




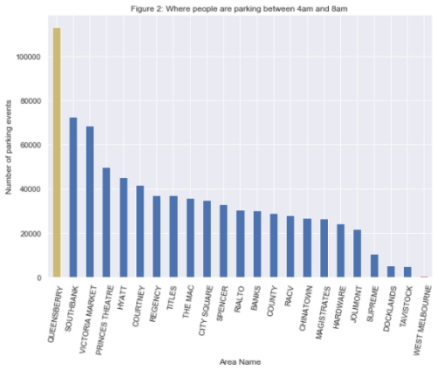


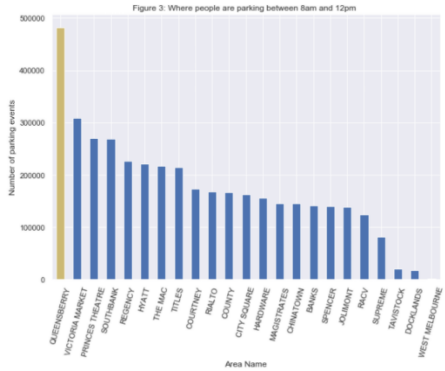






- Appendix [E] - Remaining graphs for *Which is the busiest hour for each area*





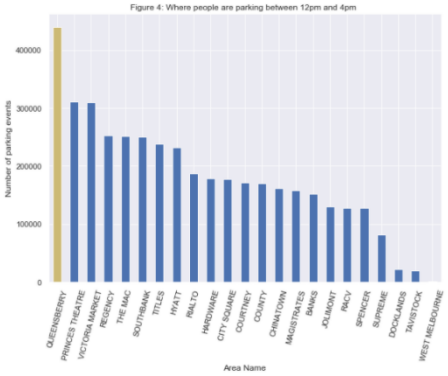


Figure 4: Where people are parking between 12pm and 4pm
Number of parking events
0
100000
200000
300000
400000
QUEENSBERRY
PRINCES THEATRE
VICTORIA MARKET
REGENCY
THE MAC
SOUTHBANK
TITLES
HYATT
RALTO
HARDWARE
CITY SQUARE
COURTNEY
COUNTY
CHINATOWN
MAGISTRATES
BANKS
JOLIMONT
RACV
SPENCER
SUPREME
DOCKLANDS
TAVISTOCK
WEST MELBOURNE
Area Name

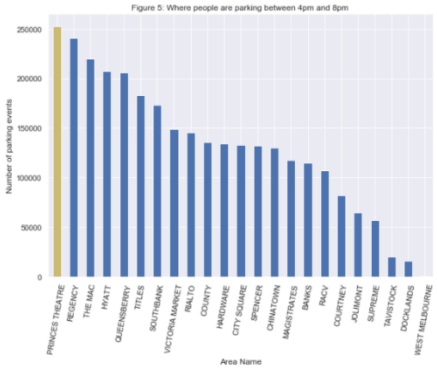


Figure 5: Where people are parking between 4pm and 8pm
Number of parking events
0
50000
100000
150000
200000
250000
PRINCES THEATRE
REGENCY
THE MAC
HYATT
QUEENSBERRY
TITLES
SOUTHBANK
VICTORIA MARKET
RALTO
COUNTY
HARDWARE
CITY SQUARE
SPENCER
CHINATOWN
MAGISTRATES
BANKS
RACV
COURTNEY
JOLIMONT
SUPREME
TAVISTOCK
DOCKLANDS
WEST MELBOURNE
Area Name

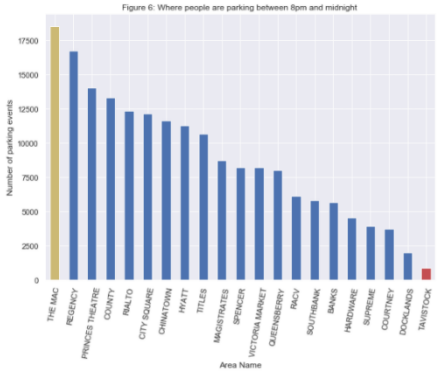


Figure 6: Where people are parking between 8pm and midnight
Number of parking events
0
2500
5000
7500
10000
12500
15000
17500
THE MAC
REGENCY
PRINCES THEATRE
COUNTY
RALTO
CITY SQUARE
CHINATOWN
HYATT
TITLES
MAGISTRATES
SPENCER
VICTORIA MARKET
QUEENSBERRY
RACV
SOUTHBANK
BANKS
HARDWARE
SUPREME
COURTNEY
DOCKLANDS
TAVISTOCK
Area Name

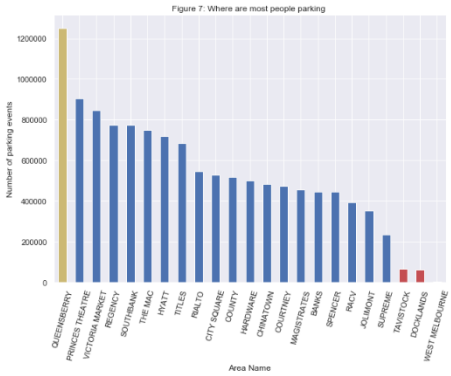


Figure 7: Where are most people parking
Number of parking events
0
200000
400000
600000
800000
1000000
1200000
QUEENSBERRY
PRINCES THEATRE
VICTORIA MARKET
REGENCY
SOUTHBANK
THE MAC
HYATT
TITLES
RALTO
CITY SQUARE
COUNTY
HARDWARE
CHINATOWN
COURTNEY
MAGISTRATES
BANKS
SPENCER
RACV
JOLIMONT
SUPREME
TAVISTOCK
DOCKLANDS
WEST MELBOURNE
Area Name

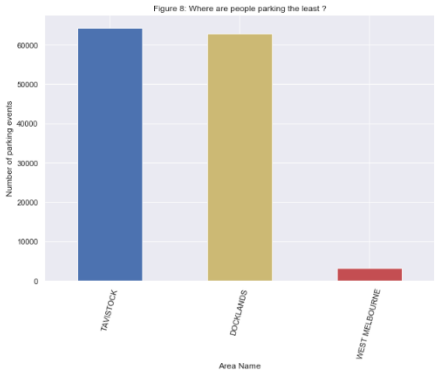


Figure 8: Where are people parking the least ?
Number of parking events
0
10000
20000
30000
40000
50000
60000
TAVISTOCK
DOCKLANDS
WEST MELBOURNE
Area Name

References

https://www.researchgate.net/post/How_to_know_that_our_dataset_is_imbalance

https://dev.to/bmor2552/binary-classification-problem-random-forest-onehot-encoder-34cg

https://intellipaat.com/blog/roc-curve-in-machine-learning/

https://datascience.stackexchange.com/questions/64441/how-to-interpret-classification-report-of-scikit-learn

https://www.ritchieng.com/machine-learning-evaluate-classification-model/

https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

https://machinelearningmastery.com/types-of-classification-in-machine-learning/?fbclid=IwAR2vpoaJBIxffwa9QBwsEcKct_Gsc2cxIpZvJn0KsCWTMqL13EtfA499Fgg

https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/

https://www.python-course.eu/confusion_matrix.php