SEPTEMBER 1, 2023

Analysis of QFES Incident Data (2011 – 2022)

Compiled Project Assessment Reports: 1A, 1B, 2A & 2B

JACK GOODRICH
UNIVERSITY OF ADELAIDE
Submitted as part of COMPSCI7319OL in the Master of Data Science (Online)

Table of Contents

Assessment 1A: Project 1 - Big Data Analysis

Jack Goodrich #1843707

Part 1: Problem Description

Modern society is dependent upon emergency services. Understanding the factors that affect how emergency services respond to individual incidents could help them anticipate the kind of response necessary to such incidents, and thereby respond faster and more effectively when they do occur. The faster organisations respond to incidents, and the sooner they complete their work at a given incident, the sooner they are freed up and able to respond to other incidents, thereby minimising harm or damage to assets or property. Understanding these factors in context is vitally important to each individual emergency service. So, what are these factors, and how exactly do they affect the way emergency services respond to incidents? These are the two overarching questions that this investigation will attempt to answer using the limited public data available.

Part 2: Dataset Description

In Australia, multiple states and territories provide data through open data portals (Australian Government, 2023, Government of South Australia, 2023, Queensland Government, 2023). In the context of emergency services, this allows access to a plethora of information that is recorded for historical individual incidents. Whereas the police and ambulance services often do not or are unable to publish much of the data that they collect, as it can be legally or medically sensitive, the fire and rescue services have more recently done their best to publish what they can, only redacting select sensitive personal information in order to deidentify individuals and incidents. For this reason, this investigation will be focusing on the fire and rescue services and will begin by looking at Queensland Fire and Emergency Services (QFES) incident data (QFES, 2023b).

The Queensland Government Open Data Portal (Queensland Government, 2023) currently provides 12 years of data from QFES (2011 – 2022). These are available as 12 separate CSV files (by year) with roughly 75,000 incidents recorded every year, totalling almost 900,000 incidents over the whole 12-year period (QFES, 2023b). Where features are not numerical measurements, the data is generally numerically encoded where strings or text data is categorical. The meta data describing all features (especially these encoded features) is provided and included here in the appendix (QFES, 2023a). In order to effectively analyse this, we may have to bring in demographic, meteorologic and other data to help elucidate trends or correlations that may exist in the datasets. After this, we can bring in data from other states and territories should we find it necessary, but for now, this data is sufficient.

Part 3: Initial Data Processing

The datasets are arranged and separated by year, but they share the same 98 features across all 12 years. The data will be imported with Python (Van Rossum and Drake, 2009) into a Jupyter Notebook (Kluyver et al., 2016) and combined into one single Pandas DataFrame (McKinney and al., 2010). After removing all the features that contain only NaNs, we are left with a single dataset totalling 864,997 rows or incidents, with 58 features containing various numbers of NaNs (see Figure 1 in Appendix). These NaNs may pose a problem when building one or more predictive models, depending on the kinds of models we choose to build. We will have to either not use those features containing NaNs in our model or replace the NaNs with appropriate values (such as 0 or mean values), depending on the context of each individual feature.

After data cleaning and pre-processing, the investigation will begin with feature analysis, attempting to find identifiable clusters and/or the strongest correlations in the data for incident duration, and then will attempt to build a predictive model with all the correlative features, or at least the more important features influencing incident duration. The kind of predictive model we chose to build will be very much dependent upon the results from our initial feature analysis. It is difficult to identify with certainty which we will use at this point, but it will most likely be a decision-tree based model, such as Random Forest Regression model

(Louppe, 2014), or XGBoost (Chen and Guestrin, 2016) (which is also an ensemble learning model), as these kinds of models suit large datasets with a large number of numerically encoded features.

Part 4: Refined Problem and Plan

Now that we have narrowed in on the QFES datasets and scope of investigation, we can refine our questions to be more specific in the context of this exact set of data:

1. Which are the most influential features affecting incident duration ('A27_A28_Elapsed_Time')?
2. In what manner do these features appear to influence incident duration?
3. Can these features (along with other potential external factors) be used to build a predictive model that can forecast the duration of incidents?

Should we not be able to proceed with the above questions for one reason or other, we will instead attempt a broader feature analysis, look for correlations and clusters between any two features, and then let the discovery of other correlations between features lead us to build a different predictive model. Failing that, we would consider emergency services data from other states and territories, revisiting our original questions, but with different data.

Appendix

Figures

Figure 1. Summary of NaN count in QFES Incident Dataset after removal of columns containing all NaNs.

Check NaN counts

```
In [96]: qidall.isna().sum()
```

```
Out[96]: A3_Station_ID                    0
         A4_Incident_No                   0
         A5_Exposure_No                   0
         A6_Alarm_Date                    0
         A2_Authority_Type                9
         A7_Day_of_Week                   0
         Day_Number                       0
         Year_Number                      0
         A8_Alarm_Time                    0
         A9_Method_of_Notification        0
         A10_A_P_Raising_Alarm            0
         A17_Suburb                    1391
         A18_Postcode                  1118
         A19_Complex_Type                 0
         A20_Fixed_Property_Use       71352
         A21_Type_of_Owner            72401
         A22_Type_of_Occupant         72398
         A23_Type_of_Incident             0
         A24_Action_Taken                 0
         A25_Control_or_Stop_Date    305488
         A26_Control_or_Stop_Time    305488
         A27_Duties_Completed_Date      423
         A28_Duties_Completed_Time      426
         A29_Peak_Personnel             418
         A35_Mutual_Aid               14644
         A36_Weather                  71686
         A37_Delayed_Arrival          72077
         A38_Shift_on_Duty            50834
         A39_CABA_Used                  544
         A42_Problems_Encountered     56206
         A56_Electricity             620568
         A57_Gas                     620405
         A58_Water                   620405
         A59_Police                  621505
         A60_Ambulance               621747
         A61_SES                     620405
         A62_Other_Fire_Service      620405
         A63_EPA                     620405
         A64_Vol_Rescue_Service      620405
         A65_Charities               620405
         A66_Govt_Welfare_Agencies   620405
         A67_Other                   858778
         A69_Fire_Name               856274
         A25_A26_Elapsed_Time        305488
         A27_A28_Elapsed_Time           419
         Completed                        0
         Firecom                          0
         Fire_Levy                       30
         MPI                              0
         AlarmTime                        0
         ControlStopTime             305488
         DutiesCompletedTime            419
         LocationId                       0
         FiuId                            0
         MasterIncidentId                 0
         MasterIncidentNumber             0
         MainAId                          0
         firecom6                         0
         dtype: int64
```

Meta Data

Meta Data – See “maina-metadata.pdf” attached to this submission (QFES, 2023a).

Referencing Format

Referencing style: default ‘Harvard’ style in EndNote (The EndNote Team, 2013). Style guides for each style are viewable inside EndNote after downloading style from this link:
https://endnote.com/downloads/styles/

References

AUSTRALIAN GOVERNMENT. 2023. *Data.gov.au* [Online]. Canberra, ACT, Australia: Australian Government. Available: https://data.gov.au/ [Accessed].

CHEN, T. & GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016 2016. ACM, 785–794.

GOVERNMENT OF SOUTH AUSTRALIA. 2023. *South Australian Government Data Directory* [Online]. Adelaide, SA, Australia: Government of South Australia. Available: https://data.sa.gov.au/ [Accessed 11/7/2023 2023].

KLUYVER, T., RAGAN-KELLEY, B., PÉREZ, F., GRANGER, B., BUSSONNIER, M., FREDERIC, J., KELLEY, K., HAMRICK, J., GROUT, J., CORLAY, S., IVANOV, P., AVILA, D., ABDALLA, S. & WILLING, C. 2016. Jupyter Notebooks - a publishing format for reproducible computational workflows. *In:* SCHMIDT, F. L. A. B. (ed.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas.* IOS Press.

LOUPPE, G. 2014. *Understanding Random Forests: From Theory to Practice.* PhD, University of Liège.

MCKINNEY, W. & AL., E. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 2010 Austin, TX, USA. Austin, TX, USA, 51-56.

QFES 2023a. Block A Meta Data. *In:* (QFES), Q. F. A. E. S. (ed.) *QFES Incident Meta Data.* Brisbane, QLD, Australia: Queensland Fire and Emergency Services.

QFES. 2023b. *QFES Incident Data* [Online]. Brisbane, QLD, Australia: Queensland Government. Available: https://www.data.qld.gov.au/dataset/qfes-incident-data [Accessed 6/7/2023 2023].

QUEENSLAND GOVERNMENT. 2023. *Open Data Portal* [Online]. Brisbane, QLD, Australia: Queensland Government. Available: https://www.data.qld.gov.au/ [Accessed 11/07/2023 2023].

THE ENDNOTE TEAM 2013. EndNote. EndNote 20 ed. Philadelphia, PA, USA: Clarivate.

VAN ROSSUM, G. & DRAKE, F. L. 2009. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

Assessment 1B: Project 1 - Big Data Analysis

Jack Goodrich #1843707

Part 1: Data Description

For part 1B of the investigation, this project will attempt to elucidate the main factors that influence incident duration in the emergency services and the manner in which they do this. It will continue to focus on the analysis of the 12-year-long incident dataset (QFES, 2023b) from the Queensland Fire and Emergency Services (QFES) with the final objective of building a predictive model. As a result of data cleaning and exploratory data analysis (EDA), the array has been narrowed down to a subset of less than half the original features (see appendix, figure 9). With the support of visualisations and statistical analyses, the most evident relationships that have been elucidated thus far during this project will be discussed herein.

Part 2: Pattern Identification

According to the EDA, it appears that incident duration is potentially influenced by a large series of factors, despite simple correlation analyses yielding no obvious results (see appendix, figure 6). The data may be so right skewed that interactions are too complex for basic correlation analyses or there is simply insufficient data in much of the dataset across numerous features. The varied effects of a select number of these features have been visualised below with plots and, where possible, supported with appropriate statistical tests. Even though much of the incident data is not normally distributed (see figure 1, below), these tests, especially the F-test and ANOVA, have been shown to still be robust to non-normal distributions (Blanca et al., 2017). For this reason, a log scale has been used to visualise the data in all the included plots.

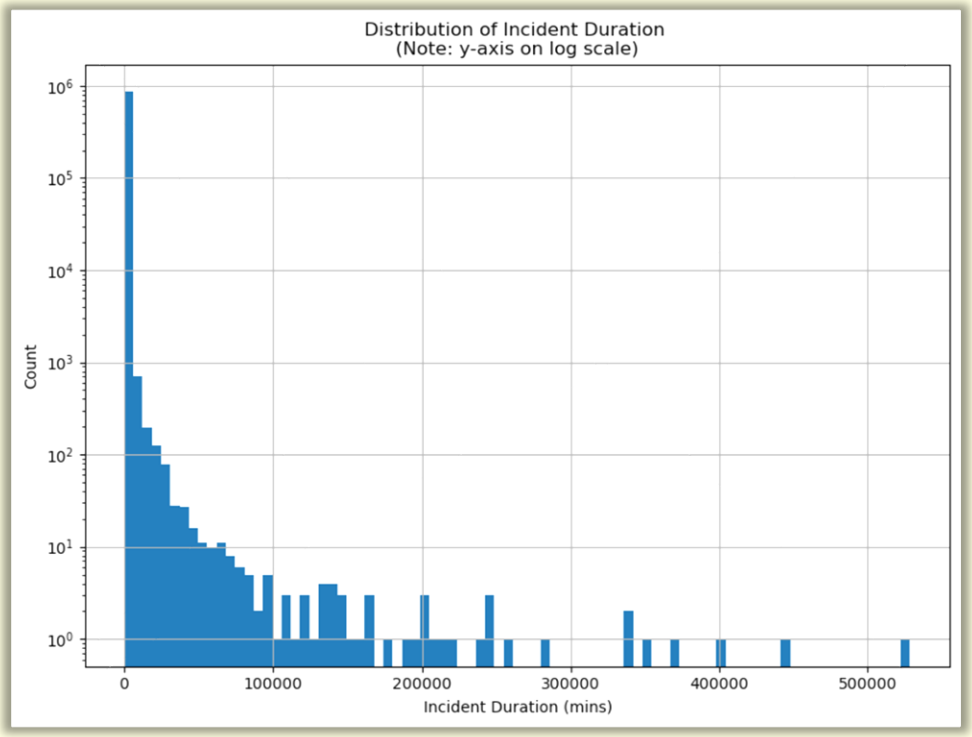

Figure 2. Univariate histogram of incident duration. This plot shows the extremely long right tail of the data, necessitating the use of the log scale to visualise the limited number of incidents occuring out to the right. As a proof, see the linear scale histogram in the appendix (figure 7).

Part 3: Visualisation

To help uncover relationships between incident duration and the potential predictor features, several new features were created using extracted information from the existing features in the dataset. Like with many of our dataset features, a number of these showed complex and/or significant relationships with incident duration (see plots below) and will likely be included in the predictive model for the next part of this investigation. We have included four example multivariate plots, below.

The first figure shows a scatter plot with two key interactions, the first between 'peak number of personnel at incident' and 'incident duration', with the second being the decision to charge a 'fire levy' (whether the alarm trigger/caller was at fault or not), which is also statistically significant (see figure 2), making it a good potential predictor of incident duration.

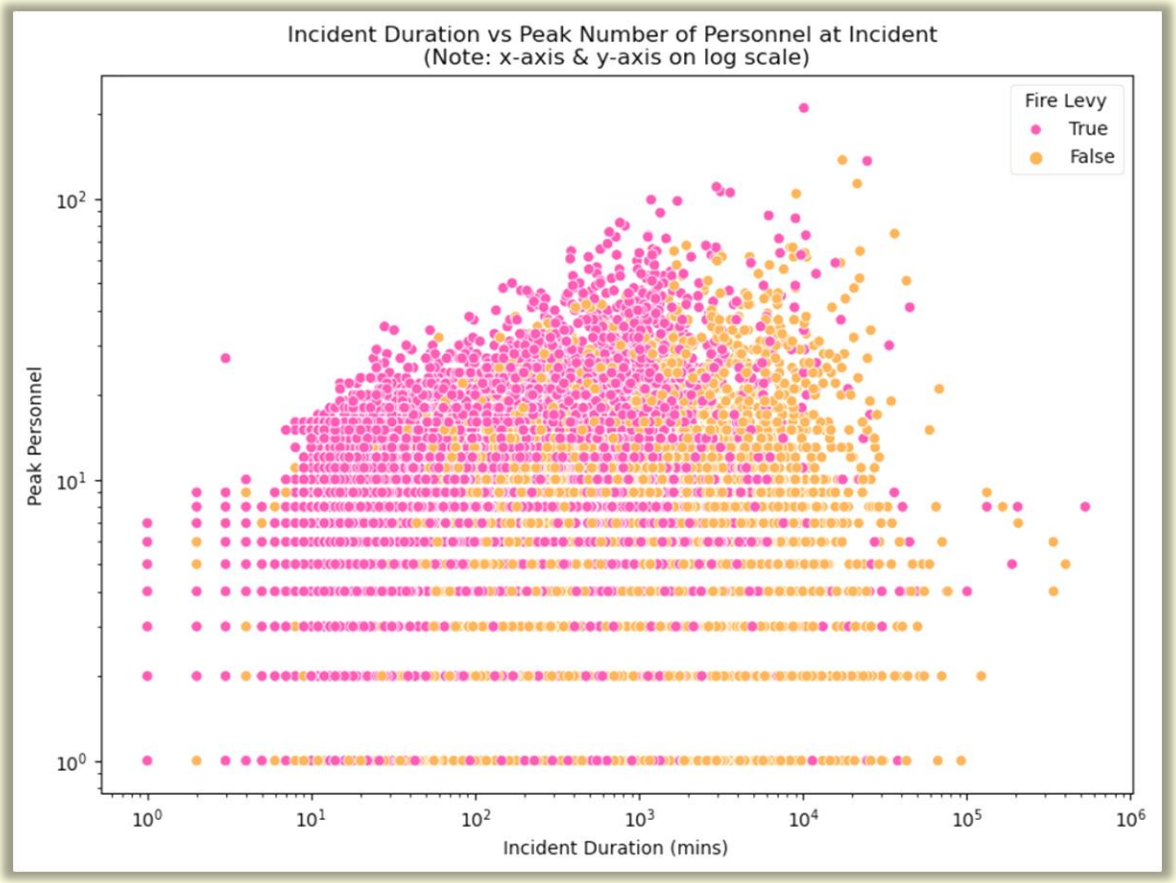

Figure 3. A scatter plot of 'incident duration' compared to the 'peak number of personnel at incident', with 'fire levy' in colour (1 = yes, 0 = no). The incidents where a fire levy has been charged appear to occur more frequently on the left of the plot, at lower incident durations, while those that have lasted for long periods of time appear less liekly to attract a fire levy on the right. This is supported by an F-test ($p < 0.001$, $F = 3374.31$) and a t-test ($p < 0.001$, $t = 27.99$).

The next three figures are box and whisker plots of three more notable interactions with incident duration which, where appropriate, are backed up statistically with an F-test or ANOVA.

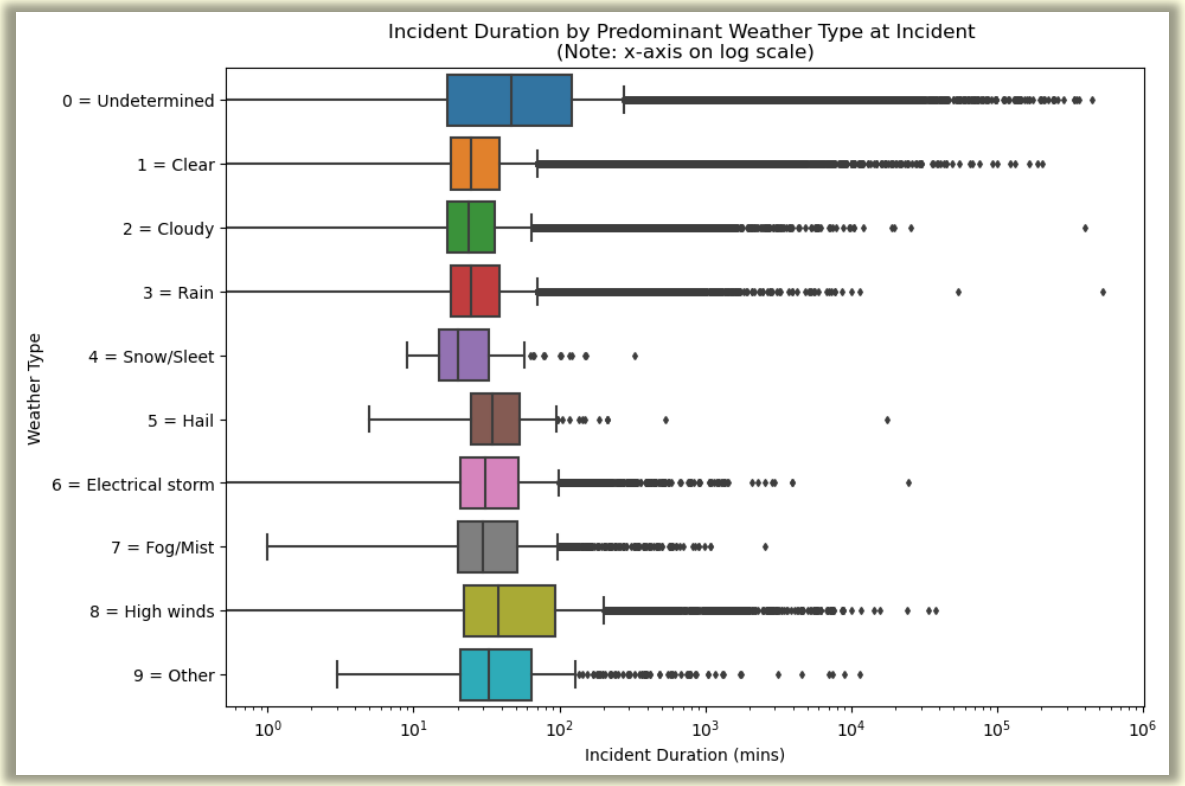

Figure 4. A box and whisker plot of incident duration for different categories of predominant weather type at incident. Verifying with an ANOVA, this difference is significant across types ($p < 0.001$, $F = 567.18$).

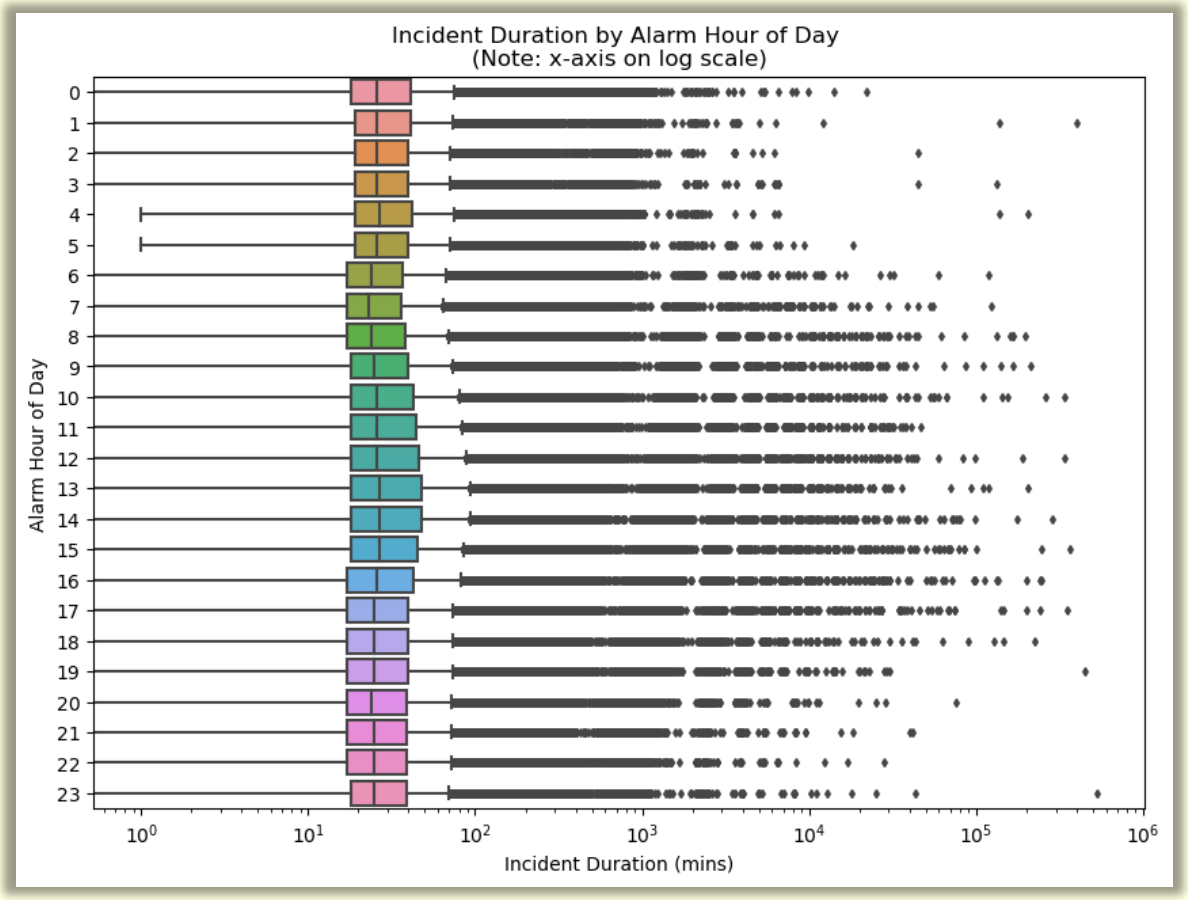

Figure 5. A box and whisker plot of incident duration across the day, separated by hour of alarm (hour in which alarm was raised/000 call was made). There is a clear ballooning out of incident duration across the middle of the day, where the mean and IQR can be seen to extend out between 0900 and 1700hrs. This is backed up by the same pattern that can also be seen in the outliers across the middle of the day.

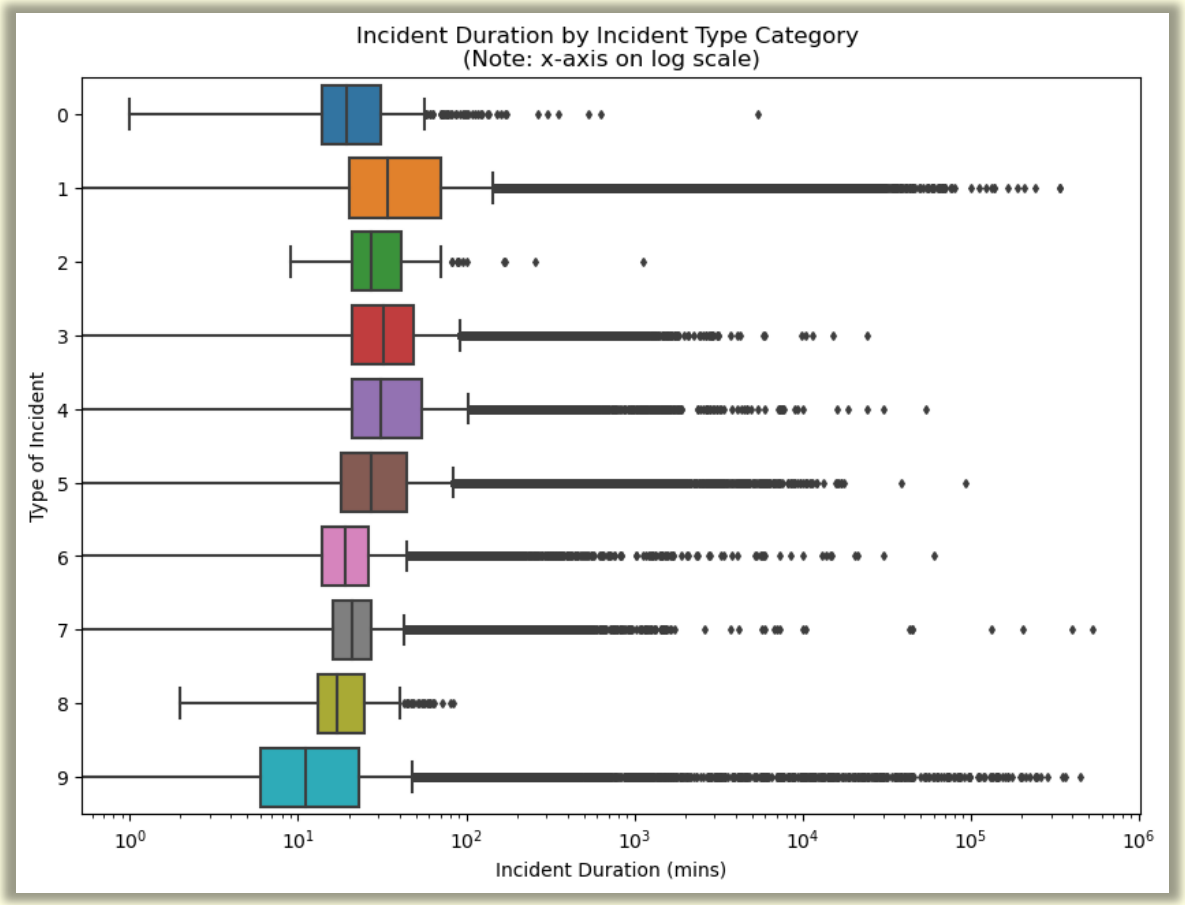

Figure 6. A box and whisker plot of incident duration across the overarching incident type categories. An ANOVA shows the difference across all categories is significant ($p < 0.001$, $F = 285.27$).

In summary, these four multivariate plots represent the kinds of interactions occurring between incident duration and many of the other features, which cannot simply be found with correlation or pair plot arrays.

Part 4: Problem Refinement

In completing EDA, we can now direct our focus to the overarching problem: using the complex interactions of these features to build a predictive model for incident duration. In the next part of this investigation, it may be beneficial to add other data into the analysis, as part of the model building, as this may help explain certain patterns of incidents (e.g. using population by postcode to explain higher concentrations of incidents in certain areas). The model will most likely be a decision-tree-based model for a number of reasons, but primarily because this has been shown to work well with incident data in the past (Rahmat-Ullah et al., 2021) and is generally far more interpretable than a neural network, even if there is a slight accuracy trade-off (Valenti et al., 2010). The interpretability of the model is crucial, especially the feature analysis, as the findings of this investigation will be reported to stakeholders at QFES in the hope of being able to enhance policy around incident response based on the findings of this investigation.

Appendix

Supplementary Figures

```
In [146]: qid_id.sort_values(ascending=False)

Out[146]: A27_A28_Elapsed_Time          1.000000
          A2_Authority_Type             0.081665
          LocationId                    0.044248
          A3_Station_ID                 0.039201
          A18_Postcode                  0.023024
          A39_CABA_Used                 0.014274
          Day_Number                    0.009316
          A29_Peak_Personnel            0.008730
          Firecom                       0.007268
          A7_Day_of_Week               -0.001381
          A42_Problems_Encountered     -0.002563
          A24_Action_Taken             -0.006914
          A10_A_P_Raising_Alarm        -0.006955
          A4_Incident_No               -0.007080
          MasterIncidentId             -0.007460
          A37_Delayed_Arrival          -0.008697
          Year_Number                  -0.009312
          A9_Method_of_Notification    -0.011400
          A23_Type_of_Incident         -0.015323
          A22_Type_of_Occupant         -0.025537
          A36_Weather                  -0.025558
          A35_Mutual_Aid               -0.030868
          A21_Type_of_Owner            -0.032999
          A20_Fixed_Property_Use       -0.035329
          A19_Complex_Type             -0.035705
          Name: A27_A28_Elapsed_Time, dtype: float64
```

Figure 7. Extracted Series in Jupyter Notebooks (Kluyver et al., 2016) showing all correlations of numerical or numerically encoded features with incident duration ('A27_A28_Elapsed_Time'). There are 25 out of all 37 remaining features. Some of these may be irrelevant as they are strictly categorical, but many are ordinal or numerical in some form and so a correlation analysis is still relevant, despite unfortunately yielding no notable correlations. (Note: the prefix 'A__' codes in the feature names have been retained for ease of reference to the supporting meta data document (QFES, 2023a). Some features include dozens of categories and subcategories which are known as divisions and subdivisions in the meta data, often necessitating the need to regularly refer back to the meta data.)

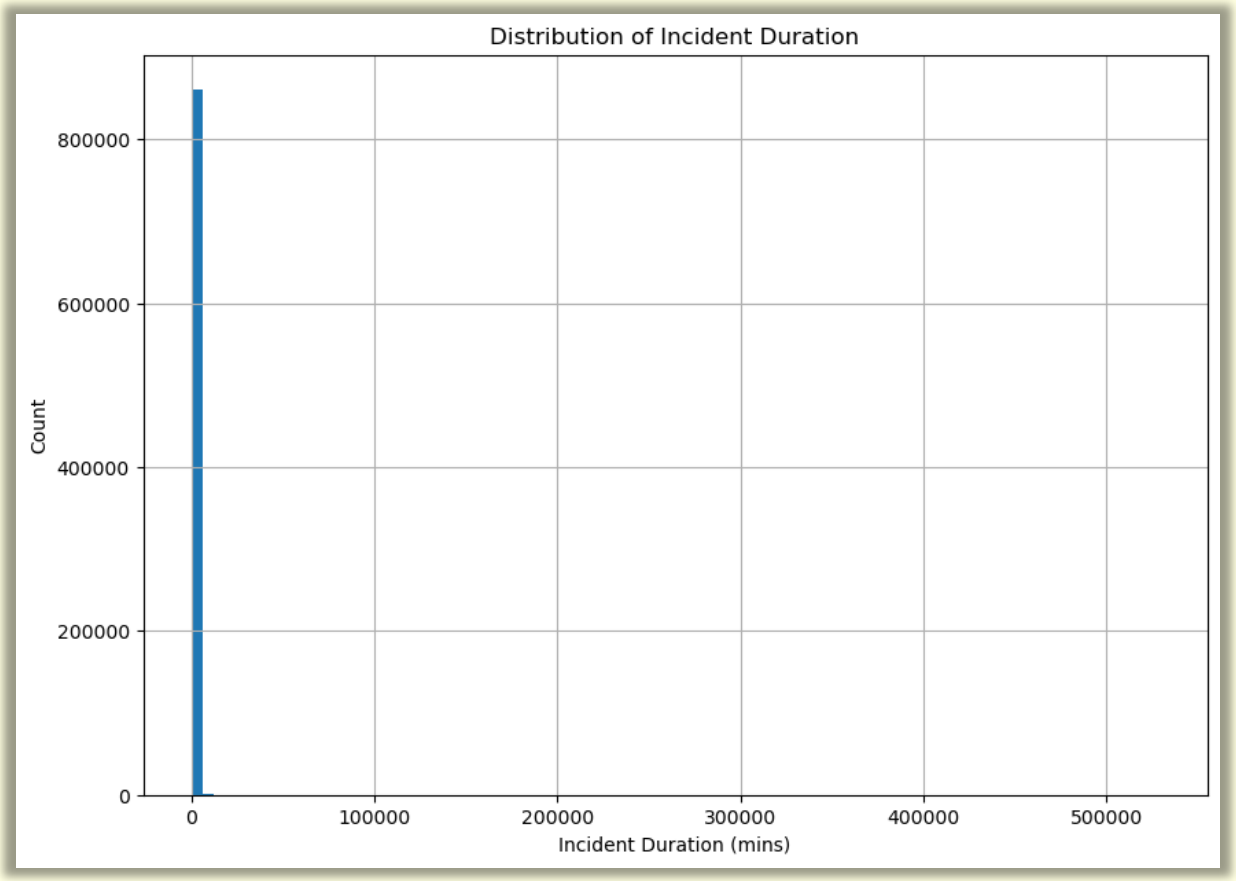

Figure 8. Histogram of incident duration (on a linear scale) for all incidents (bins=85). Extreme right tail very evident from distribution, hence more appropriate to have y-axis on log scale as in body of report.

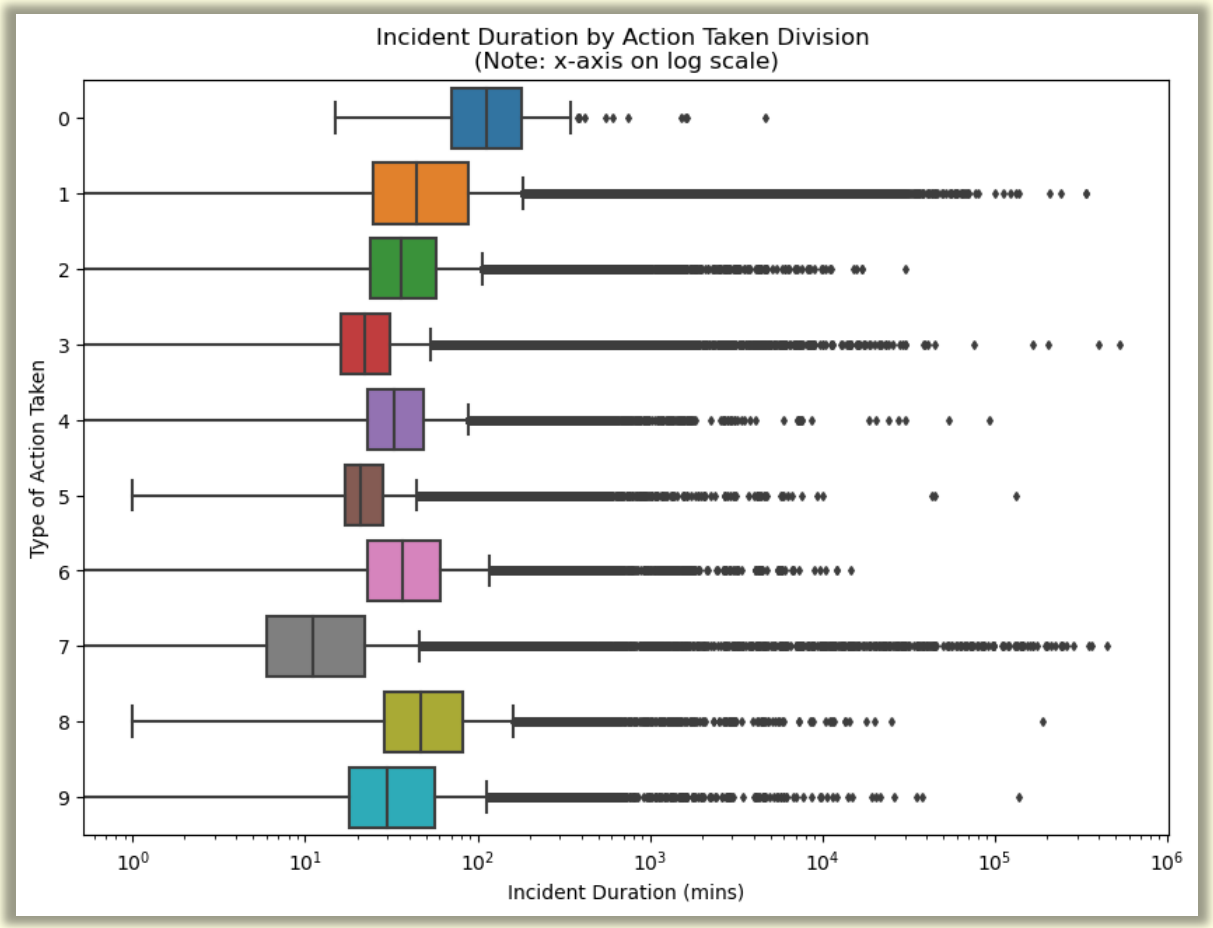

Figure 9. A box and whisker plot of incident duration across the different overarching categories of action taken. The x-axis is on a log scale to enable visualisation of the IQR and mean. The mean and IQR between groups appears significantly different, and a one-way ANOVA across all categories backs this up ($p < 0.001$, $F = 321.99$).

```
In [231]: qidall.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 861187 entries, 0 to 864996
Data columns (total 34 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   A3_Station_ID               861187 non-null  int64
 1   A4_Incident_No              861187 non-null  int64
 2   A2_Authority_Type           861187 non-null  float64
 3   A7_Day_of_Week              861187 non-null  int64
 4   Day_Number                  861187 non-null  int64
 5   Year_Number                 861187 non-null  int64
 6   A9_Method_of_Notification   861187 non-null  int64
 7   A10_A_P_Raising_Alarm       861187 non-null  int64
 8   A18_Postcode                861187 non-null  category
 9   A19_Complex_Type            861187 non-null  int64
 10  A20_Fixed_Property_Use      861187 non-null  float64
 11  A21_Type_of_Owner           861187 non-null  float64
 12  A22_Type_of_Occupant        861187 non-null  float64
 13  A23_Type_of_Incident        861187 non-null  int64
 14  A24_Action_Taken            861187 non-null  int64
 15  A29_Peak_Personnel          861187 non-null  int32
 16  A35_Mutual_Aid              861187 non-null  float64
 17  A36_Weather                 861187 non-null  float64
 18  A37_Delayed_Arrival         861187 non-null  category
 19  A38_Shift_on_Duty           861187 non-null  object
 20  A39_CABA_Used               861187 non-null  category
 21  A42_Problems_Encountered    861187 non-null  float64
 22  A27_A28_Elapsed_Time        861187 non-null  float64
 23  Firecom                     861187 non-null  int64
 24  Fire_Levy                   861187 non-null  category
 25  MPI                         861187 non-null  object
 26  LocationId                  861187 non-null  int64
 27  FiuId                       861187 non-null  object
 28  MasterIncidentId            861187 non-null  int64
 29  MasterIncidentNumber        861187 non-null  object
 30  MainAId                     861187 non-null  object
 31  AlarmHourOfDay              861187 non-null  int32
 32  IncidentTypeDivision        861187 non-null  int32
 33  ActionTakenDivision         861187 non-null  int32
dtypes: category(4), float64(8), int32(4), int64(13), object(5)
memory usage: 194.7+ MB
```

Figure 10. Table of remaining features of interest (including the target feature of incident duration, 'A27_A28_Elapsed_Time') after data cleaning, pre-processing and EDA.

References

BLANCA, M. J., ALARCÓN, R., ARNAU, J., BONO, R. & BENDAYAN, R. 2017. Non-normal data: Is ANOVA still a valid option? *Psicothema,* 29**,** 552-557.

KLUYVER, T., RAGAN-KELLEY, B., PÉREZ, F., GRANGER, B., BUSSONNIER, M., FREDERIC, J., KELLEY, K., HAMRICK, J., GROUT, J., CORLAY, S., IVANOV, P., AVILA, D., ABDALLA, S. & WILLING, C. 2016. Jupyter Notebooks - a publishing format for reproducible computational workflows. *In:* SCHMIDT, F. L. A. B. (ed.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas.* IOS Press.

QFES 2023a. Block A Meta Data. *In:* (QFES), Q. F. A. E. S. (ed.) *QFES Incident Meta Data.* Brisbane, QLD, Australia: Queensland Fire and Emergency Services.

QFES. 2023b. *QFES Incident Data* [Online]. Brisbane, QLD, Australia: Queensland Government. Available: https://www.data.qld.gov.au/dataset/qfes-incident-data [Accessed 6/7/2023 2023].

RAHMAT-ULLAH, Z., ALSMADI, S. & HAMAD, K. Classifying and Forecasting Traffic Incident Duration Using Various Machine Learning Techniques. 14th International Conference on Developments in eSystems Engineering (DeSE), 2021 2021 Sharjah, United Arab Emirates. IEEE, 388-393.

VALENTI, G., LELLI, M. & CUCINA, D. 2010. A comparative study of models for the incident duration prediction. *European Transport Research Review,* 2**,** 103-111.

Assessment 2A: Project 2 – 'Image And Text Big Data Analysis'

Jack Goodrich #1843707

Part 1: Problem Description

As part of this ongoing investigation into the factors that influence incident duration in the emergency services and the manner in which they do this, the main objective of this second project was to build, train and evaluate one or more predictive models. In the first project, numerous features and their respective complex interactions with incident duration were identified and analysed (see Assessments 1A and 1B). Here those interactions are exploited for the construction and training of multiple predictive models for incident duration. The inputs come from 12 years of Queensland Fire and Emergency Service (QFES) incident data (QFES, 2023b) available on the Queensland Government Open Data Portal (Queensland Government, 2023). The data was imported into a Jupyter notebook (Kluyver et al., 2016) as 12 individual CSV files (one for each year) and concatenated into a single Pandas (McKinney and al., 2010) 'DataFrame' totalling 864,997 rows and 98 features.

After data cleaning and pre-processing, our output data contained less than half the number of original features but more than 99% of the incidents (see appendix, figure 1). All variables that had the same number of categories as the length of the dataset (unique ID variables, and the like) were dropped, as well as variables containing more NaN entries than could be feasibly dropped by row (e.g. NaN entries for a given variable making up more than 50% of rows). In other features, if there were negligible numbers of rows with missing data (e.g. less than 1%), then these rows were able to be dropped if there was not an extant 'undetermined' numerically encoded category listed in the meta data document that could be used to replace them, as in other features (QFES, 2023a). Finally, all remaining outliers were removed using a cut-off of three standard deviations from the mean for incident duration. Once clean and stored as the correct variable types, the data was used to tune and train the constructed predictive model.

Part 2: Methodology

Given the size of this dataset, there were few options in terms of machine learning techniques which were feasible to run on a standard desktop/laptop machine without deferring to cloud computing to get the model built and trained, especially when it came to the computationally expensive and onerous task of hyperparameter tuning. A Random Forest regression model would have been another suitable option (Louppe, 2014), but for the large memory requirements of one-hot encoding all the categorical variables. For this part of the investigation the most suitable technique chosen was a decision-tree-based ensemble learning model known as XGBoost (XGBoost Developers, 2022a). It uses a non-exhaustive approach that starts with any weak learner, and then iteratively improves using superior weights of subsequent trees (Chen and Guestrin, 2016). It was chosen for its deft balance between accuracy, speed and interpretability for this dataset (Shwartz-Ziv and Armon, 2021).

XGBoost has a number of advantages that this project required over other models: it is made to be used on large datasets given its non-exhaustive, iterative learning technique; it is decision-tree-based which makes it good at dealing with large numbers of variables, including categorical variables; it has useful elements of other machine learning techniques built-in, such as lasso and ridge regression (L1 and L2 regularisation); and it can one-hot encode categorical variables under-the-hood. It also has a neat scikit-learn wrapper/API that makes it very streamlined to use, but this is simply a bonus for speed and brevity of code during the process.

Once the dataset had been cleaned and pre-processed for analysis, the preliminary model was built with the default settings and the training sets were fit to the model. The model was later refined using hyperparameter tuning, followed by overfitting reduction (see records in appendix, figure 5). Prediction results for all stages were derived from both the testing and training sets separately using the '.predict()' function and the RMSE was calculated for both (see appendix, figure 3).

Part 3: Setup and Results

The experiment setup begun with clean and relevant data (using only variables which had exhibited statistically significant differences between groups, or clear patterns across groups within a given feature). 25% of the data was held out for model testing, post-training. The model was then constructed using tuned hyperparameters from a 3-fold cross-validation with the GridSearchCV function (see appendix, figure 2). Evaluation of the preliminary model showed that, although training RMSE was relatively low, the testing RMSE was too high compared to the dummy regression baseline calculated on the mean. This suggested that the model was overfitting.

Even with hyperparameter tuning, the model was very good at performing on the training data but failed repeatedly to predict well in the testing data. After the basic hyperparameters were identified and set, some had to be changed to help avoid the overfitting to the training data that was occurring. After reducing overfitting, the model performed adequately, but not considerably better than the dummy regressor calculated on the mean. At best, the model was performing around 15% better in testing than our dummy regressor (see appendix, figure 3).

Part 4: Problem Refinement

To attempt to prevent overfitting in these models a number of hyperparameters were tuned (XGBoost Developers, 2022c, XGBoost Developers, 2022b). Firstly, to reduce model complexity, the focus was placed on tuning 'max_depth', 'min_child_weight' and 'gamma'. Optimising these meant that the model was kept simple enough to avoid over complexifying, thereby preventing overfitting to the training data. However, this was insufficient to solve the problem, so the next step was to add noise using 'subsample' and 'colsample_bytree', thereby introducing another aspect of randomness, as well as adjusting the 'learning_rate'/'eta'. However, none of these managed to stop the overfitting that was occurring in the preliminary models built thus far (see appendix, figure 5), so L1 and L2 regularisation had to then be considered. It was only by increasing either of these values significantly that the model was able to improve testing set performance (see appendix, figure 5), with L1 regularisation being the most effective in the end. This meant that more features contributed to the final predictive model, i.e. fewer features' weights went to 0 (see appendix, figure 4), while keeping the model relatively uncomplex, which together helped to eliminate overfitting.

However, the results of the model predicting only 15% better than the mean suggests one of two things: either the model is not well tuned or constructed, or that there are extremely complex relationships at play making it quite difficult to predict incident duration given the data. It is possible a deep neural network may perform better than XGBoost for this kind of dataset, albeit at the expense of interpretability. This could be one tactic employed for future investigations if a predictive model proved truly desirable for the stakeholders at QFES. However, it seems that XGBoost is often still the model of choice to date when it comes to tabular data (Shwartz-Ziv and Armon, 2021), so further tuning on more hyperparameters is likely necessary in a more scientifically robust way, such as with nested k-fold cross-validation (Brownlee, 2020).

Appendix

Supplementary Figures

```
In [105]: qidall.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 859717 entries, 0 to 864996
Data columns (total 31 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   A3_Station_ID              859717 non-null  category
 1   A2_Authority_Type          859717 non-null  category
 2   A7_Day_of_Week             859717 non-null  category
 3   Day_Number                 859717 non-null  category
 4   Year_Number                859717 non-null  category
 5   A9_Method_of_Notification  859717 non-null  category
 6   A10_A_P_Raising_Alarm      859717 non-null  category
 7   A17_Suburb                 859717 non-null  category
 8   A18_Postcode               859717 non-null  category
 9   A19_Complex_Type           859717 non-null  category
 10  A20_Fixed_Property_Use     859717 non-null  category
 11  A21_Type_of_Owner          859717 non-null  category
 12  A22_Type_of_Occupant       859717 non-null  category
 13  A23_Type_of_Incident       859717 non-null  category
 14  A24_Action_Taken           859717 non-null  category
 15  A29_Peak_Personnel         859717 non-null  int32
 16  A35_Mutual_Aid             859717 non-null  category
 17  A36_Weather                859717 non-null  category
 18  A37_Delayed_Arrival        859717 non-null  category
 19  A38_Shift_on_Duty          859717 non-null  category
 20  A39_CABA_Used              859717 non-null  int32
 21  A42_Problems_Encountered   859717 non-null  category
 22  A27_A28_Elapsed_Time       859717 non-null  float64
 23  Firecom                    859717 non-null  category
 24  Fire_Levy                  859717 non-null  category
 25  MPI                        859717 non-null  category
 26  LocationId                 859717 non-null  category
 27  AlarmHourOfDay             859717 non-null  category
 28  FixedTypeDivision          859717 non-null  category
 29  IncidentTypeDivision       859717 non-null  category
 30  ActionTakenDivision        859717 non-null  category
dtypes: category(28), float64(1), int32(2)
memory usage: 56.7 MB

In [105]: 864997-859717

Out[105]: 5280

In [110]: (864997-859717)/864997

Out[110]: 0.006104067412950565

Dataset now contains 859717 incidents, which is a reduction of only 5280 (~0.6%) from the original 864997 lines of the total combined 12 years of data.
```

Figure 11. Screenshot showing information summary of final output dataset as well as calculation of total incidents/rows dropped from dataset after data cleaning and pre-processing was complete.

```
Train Model

In [133]: xgbr_model.fit(X_train, y_train) # default n_estimators is 100 trees in the forest

Out[133]: XGBRegressor(alpha=11000, base_score=None, booster=None, callbacks=None,
                       colsample_bylevel=None, colsample_bynode=None, colsample_bytree=1,
                       early_stopping_rounds=None, enable_categorical=True,
                       eval_metric=None, feature_types=None, gamma=0, gpu_id=None,
                       grow_policy=None, importance_type=None,
                       interaction_constraints=None, lambda=1, learning_rate=0.3,
                       max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None,
                       max_delta_step=None, max_depth=6, max_leaves=None,
                       min_child_weight=0.01, missing=nan, monotone_constraints=None,
                       n_estimators=30, n_jobs=None, num_parallel_tree=None, ...)
```

Figure 12. Screenshot showing list of hyperparameters and their values for the best fit XGBoost model. These align with the best performing model hyperparameter setup in figure 5, below.

XGBR Model RMSE

In [136]:
```
print('Prelinimary XGBR Model')
y_xgbr_train_predict = xgbr_model.predict(X_train) # Prediction in the training dataset
y_xgbr_test_predict = xgbr_model.predict(X_test) # Prediction in the testing dataset

RMSE_training = np.sqrt(mean_squared_error(y_train, y_xgbr_train_predict)) # use numpy .sqrt() function to get training RMSE
RMSE_test = np.sqrt(mean_squared_error(y_test, y_xgbr_test_predict)) # use numpy .sqrt() function to get test RMSE

print('The baseline RMSE is', rmse)
print('The model RMSE in training is', RMSE_training)
print('The model RMSE in testing is', RMSE_test)
```

```
Prelinimary XGBR Model
The baseline RMSE is 213.66003464237312
The model RMSE in training is 165.88451770890066
The model RMSE in testing is 179.89732619693675
```

Figure 13. Screenshot of RMSE results from testing and training predictions during evaluation. The model RMSE in testing can be seen to be more than 15% better than our baseline (dummy reagressor caluclated on mean), and the training RMSE is slightly lower again. This was the best testing RMSE achieved during hyperparameter tuning and overfitting reduction.

Feature Importance

In [134]:
```
pd.DataFrame({'Variable':X.columns,
              'Importance':xgbr_model.feature_importances_}).sort_values('Importance', ascending=False)
```

Out[134]:

	Variable	Importance
13	A23_Type_of_Incident	0.170255
1	A2_Authority_Type	0.164359
14	A24_Action_Taken	0.063073
0	A3_Station_ID	0.062199
23	Fire_Levy	0.059203
7	A17_Suburb	0.054239
8	A18_Postcode	0.048841
15	A29_Peak_Personnel	0.043512
3	Day_Number	0.038984
19	A38_Shift_on_Duty	0.037783
10	A20_Fixed_Property_Use	0.035091
17	A36_Weather	0.020952
27	FixedTypeDivision	0.020056
5	A9_Method_of_Notification	0.019822
29	ActionTakenDivision	0.017385
4	Year_Number	0.016206
18	A37_Delayed_Arrival	0.015873
21	A42_Problems_Encountered	0.015630
6	A10_A_P_Raising_Alarm	0.014633
25	LocationId	0.012540
20	A39_CABA_Used	0.010248
9	A19_Complex_Type	0.009968
11	A21_Type_of_Owner	0.009101
26	AlarmHourOfDay	0.008436
12	A22_Type_of_Occupant	0.007875
2	A7_Day_of_Week	0.007542
16	A35_Mutual_Aid	0.006849
28	IncidentTypeDivision	0.006414
24	MPI	0.002930
22	Firecom	0.000000

Figure 14. A list of features and their associated importance in our final XGBoost model. The feature importance can be seen to spread to all but one of the features, as L1 regularisation was used to help prevent overfitting, before this, a substantial number features' weights would go to 0 during training.

Tuning Results:

Tuning for Model Complexity (only adjusting 'max_depth', 'n_estimators', and 'min_child_weight'):

- Round 1:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 49.52948405021306
  - The model RMSE in testing is 196.64013692209673
  - parameters: {'alpha': 0, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 1, 'learning_rate': 0.3, 'max_depth': 8, 'min_child_weight': 0.01, 'n_estimators': 30, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}

Introduce Randomness (adjust 'subsample', 'colsample_bytree' and 'eta'/'learning_rate':

- Round 2:
  - Kernal was dying, unable to tune these parameters with this dataset

Wind back initial parameters to reduce overfitting and modify L1 and L2 regularisation:

- Round 3:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 179.17761626376344
  - The model RMSE in testing is 186.19624447987286
  - {'alpha': 0, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 8000, 'learning_rate': 0.3, 'max_depth': 4, 'min_child_weight': 0.01, 'n_estimators': 30, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}

Better hyperparameters without tuning; lambda and n_estimators were giving better testing set performance before this tuning round, so will have to modify parameters manually in the k-fold cv...

- Round 4:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 180.53114570208274
  - The model RMSE in testing is 186.31152183840106
  - {'alpha': 0, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 12000, 'learning_rate': 0.3, 'max_depth': 4, 'min_child_weight': 0.01, 'n_estimators': 32, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}

Must modify hyperparameters manually to counteract overfitting; let's try L1 regularisation instead of L2...

- Round 5:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 172.88666964631653
  - The model RMSE in testing is 180.55691961440948
  - {'alpha': 11000, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 1, 'learning_rate': 0.3, 'max_depth': 4, 'min_child_weight': 0.01, 'n_estimators': 30, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}
- Round 6:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 169.26565178355548
  - The model RMSE in testing is 180.53199894566256
  - {'alpha': 11000, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 1, 'learning_rate': 0.3, 'max_depth': 5, 'min_child_weight': 0.01, 'n_estimators': 30, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}
- Round 7:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 165.88451770890066
  - The model RMSE in testing is 179.89732619693675
  - {'alpha': 11000, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 1, 'learning_rate': 0.3, 'max_depth': 6, 'min_child_weight': 0.01, 'n_estimators': 30, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}

After setting 'max_depth' to >6 model becomes too complex again and begins to start overfitting; the above round seems like the best performance we can get for now as the results below begin to get worse for the testing set:

- Round 8:
  - The baseline RMSE is 213.66003464237312
  - The model RMSE in training is 162.0567213343404
  - The model RMSE in testing is 180.18034241660314
  - {'alpha': 11000, 'colsample_bytree': 1, 'enable_categorical': True, 'gamma': 0, 'lambda': 1, 'learning_rate': 0.3, 'max_depth': 7, 'min_child_weight': 0.01, 'n_estimators': 30, 'random_state': 77, 'subsample': 1, 'tree_method': 'gpu_hist'}

Figure 15. A record of all the hyperparameter values and evaluation results from hyperparameter tuning and overfitting reduction. Model complexity could not be successfully reduced to avoid overfitting with GridSearchCV function alone, as the function minimises RMSE across the whole dataset, meaning that our testing set always performed badly as the model was overfitting to the training data during each fit. The only way overfitting was avoided was by tuning the L1 and L2 regularisation hyperparameters, with L1 regularisation being the most effective in the end.

References

BROWNLEE, J. 2020. *Nested Cross-Validation for Machine Learning with Python* [Online]. Machine Learning Mastery. Available: https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/ [Accessed 6 August 2023].

CHEN, T. & GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016 2016. ACM, 785–794.

KLUYVER, T., RAGAN-KELLEY, B., PÉREZ, F., GRANGER, B., BUSSONNIER, M., FREDERIC, J., KELLEY, K., HAMRICK, J., GROUT, J., CORLAY, S., IVANOV, P., AVILA, D., ABDALLA, S. & WILLING, C. 2016. Jupyter Notebooks - a publishing format for reproducible computational workflows. *In:* SCHMIDT, F. L. A. B. (ed.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas.* IOS Press.

LOUPPE, G. 2014. *Understanding Random Forests: From Theory to Practice.* PhD, University of Liège.

MCKINNEY, W. & AL., E. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 2010 Austin, TX, USA. Austin, TX, USA, 51-56.

QFES 2023a. Block A Meta Data. *In:* (QFES), Q. F. A. E. S. (ed.) *QFES Incident Meta Data.* Brisbane, QLD, Australia: Queensland Fire and Emergency Services.

QFES. 2023b. *QFES Incident Data* [Online]. Brisbane, QLD, Australia: Queensland Government. Available: https://www.data.qld.gov.au/dataset/qfes-incident-data [Accessed 6/7/2023 2023].

QUEENSLAND GOVERNMENT. 2023. *Open Data Portal* [Online]. Brisbane, QLD, Australia: Queensland Government. Available: https://www.data.qld.gov.au/ [Accessed 11/07/2023 2023].

SHWARTZ-ZIV, R. & ARMON, A. 2021. Tabular Data: Deep Learning is Not All You Need. *arXiv pre-print server*.

XGBOOST DEVELOPERS. 2022a. *Introduction to Boosted Trees* [Online]. Available: https://xgboost.readthedocs.io/en/stable/tutorials/model.html [Accessed 1 August 2023].

XGBOOST DEVELOPERS. 2022b. *Notes on Parameter Tuning* [Online]. Available: https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.html [Accessed 1 August 2023].

XGBOOST DEVELOPERS. 2022c. *XGBoost Parameters* [Online]. Available: https://xgboost.readthedocs.io/en/stable/parameter.html [Accessed 1 August 2023].

Assessment 2B: Project 2 — Big Data Analysis

Submitted by Jack Goodrich #1843707

Part 1: Summary

The main objective of this four-part investigation was to create a predictive model for incident duration in the context of emergency services. For this project, specific data from the Queensland Fire and Emergency Services (QFES) was used from a 12-year period (2011 – 2022) (QFES, 2023b). The model created in the last part of this investigation was able to predict incident duration with around a 15% improved error rate than the mean baseline. The complexities of this dataset are evident with 859,717 incidents spanning 30 predictor variables, most of which were categorical features numerically encoded for analysis, sometimes with dozens of categories (QFES, 2023a). It was therefore imperative to choose a non-exhaustive model to analyse the set, otherwise the analysis would have been far too computationally expensive for a standard benchtop machine. The decision to use XGBoost allows vast room for improvement as it is a model that is extremely sensitive to hyperparameter settings given it uses sometimes dozens of hyperparameters, depending on the kind of model being built and trained. The areas of potential improvement will therefore be outlined and discussed below, and alternative solutions will be proposed.

Part 2: Analysis and Visualisation

The results of the XGBoost model show that, in testing, the incident duration was able to be predicted with 15.8% lower root mean square error (RMSE) than mean baseline (see figure 1, below). This was done through careful hyperparameter tuning, as noted in part 2A, and allowed the model to learn a solution over just 30 boosting rounds (see figure 2, below). The set of final features included in the model showed that almost all the remaining features contributed to a solution and that, as found in part 2A of this investigation, eliminating or minimising the influence of the less important features would lead to overfitting (see figure 3, below).

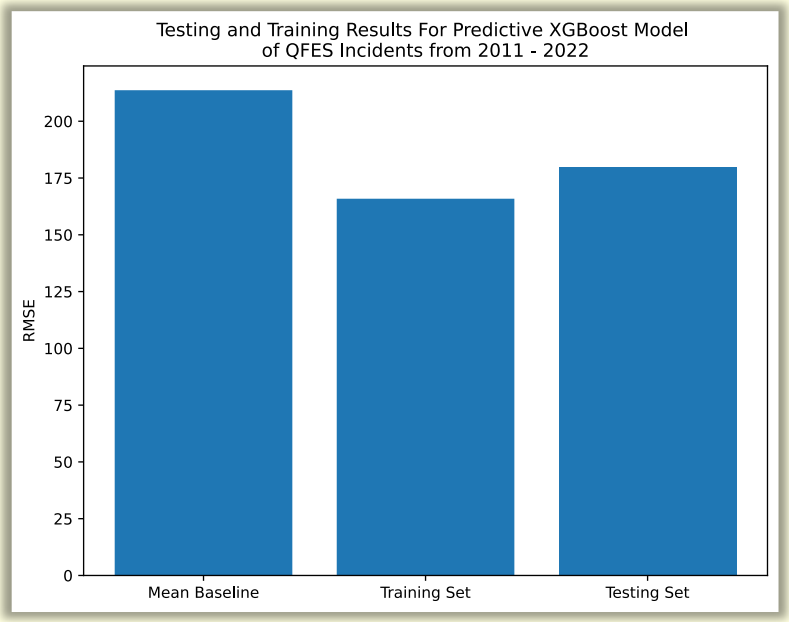

Figure 16. A bar plot showing comparison between baseline, training and testing results in RMSE. The testing RMSE can be seen to be only slightly higher (7.7%) than the training RMSE, but still markedly (15.8%) lower than baseline.

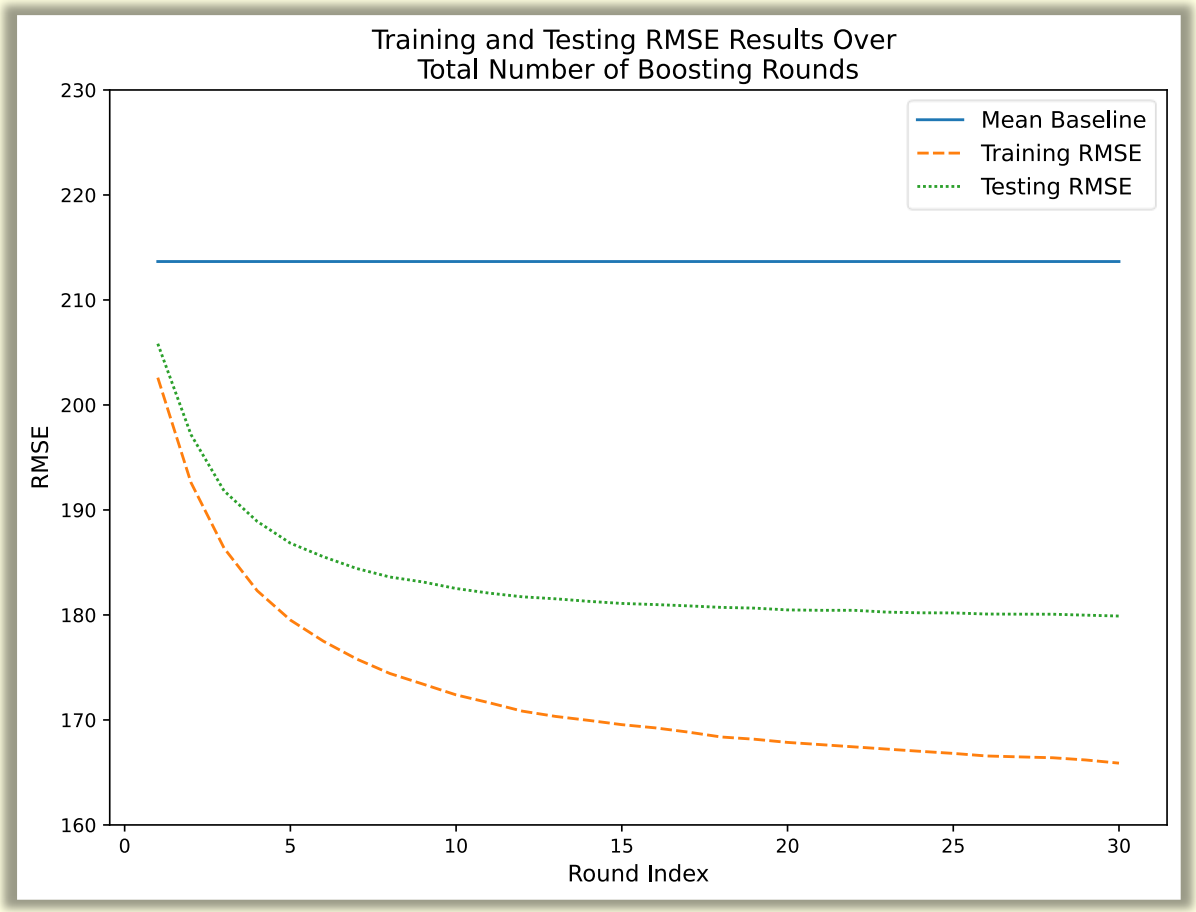

Figure 17. A line graph plotting individual boosting round results over the total number of boosting rounds for both training and testing in RMSE. The testing and training RMSE can be seen to start high, close to the mean baseline but after 30 rounds end up considerably below baseline, with testing only slightly higher (7.7%) than the training RMSE.

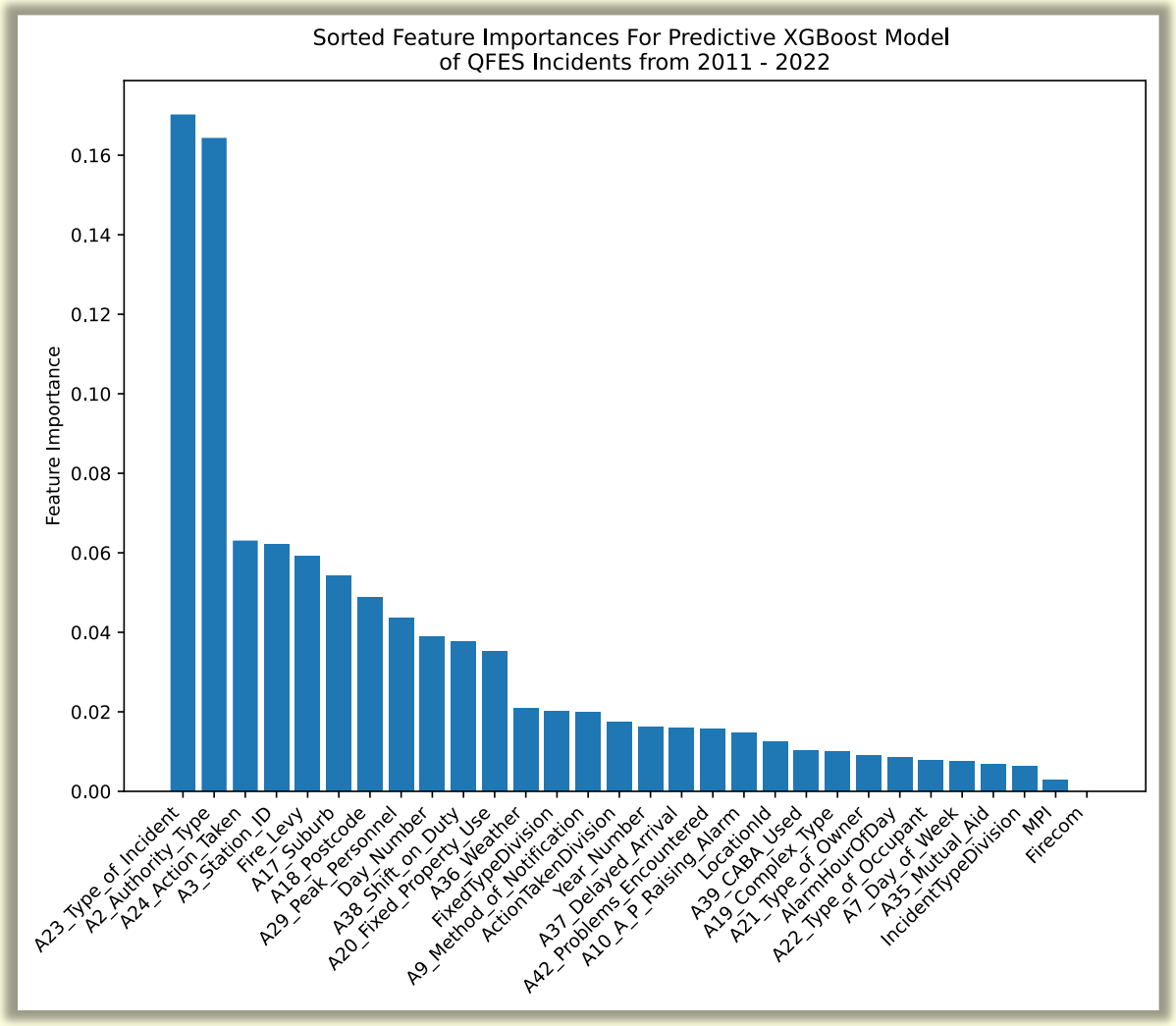

Figure 18. A bar plot of the list of features in the final model and their associated importance. The first two features on the left, the type of incident and type of responding authority, can be seen to be significantly more informative to the model than the others, but there is a noticeable second tier of features in the following nine, which sit above the remaining features on the right. From weather type onwards, however, the features appear to exert almost negligible influence over the model, but without these, the model was still overfitting, so they are clearly necessary to maintain given that the model overfitting was overcome by using L1 regularisation.

Part 3: Solution Improvement

The above figures show a sound result in any sense, as it is an obvious improvement on simply using a dummy regressor based on the mean, however, there are patent, real-world implications of using a model that still has such considerable error. It must be noted that if a model such as this was to be deployed in real time to predict incident duration and inform the decisions of QFES in how the resources are allocated, then there is significant room for error, making it an unreliable model for such applications. It could be used in a general sense to help stakeholders at QFES determine where resources may need to be allocated in advance, however, if the accuracy of the model could be improved upon first by reducing error, then it would be more useful for numerous applications across the board at QFES.

Several techniques could be used to improve on the solutions for building a predictive model for incident duration. As noted in 2A, using nested cross-validation and spending more time on tuning more hyperparameters would be the first and most obvious step in finding improved solution for the model (Brownlee, 2020b). Unfortunately, this is not feasible within the remaining time frame given the number of

hyperparameters that need to be tuned and the processing power required. Another way would be to bring in more data. Spatial data, environment data and further incident data, including traffic data, are some extremely important considerations which have proved useful for similar models in the past (Mukhopadhyay et al., 2022, Mukhopadhyay et al., 2020, Martin et al., 2021). A larger model with more nuanced information could become a far better prediction solution, albeit at the potential expense of making the model more complex.

Another way of making a more useful predictive model for the stakeholders at QFES would be to specifically create a timeseries forecasting model (Hyndman and Athanasopoulos, 2021), which could even be done using XGBoost (Brownlee, 2020a). Again, this would not be feasible to build in the remaining timeframe, but, like any well-constructed forecasting model, this would help stakeholders at QFES in making advanced decisions, months or years out, about where resources can best be allocated as the state grows in population making certain areas and times of day far busier than historically.

Part 4: Conclusion

The final solutions proposed above will improve upon the XGBoost model devised or create new models that will be able to answer certain questions or problems in more specific and appropriate ways. It is vital that future decisions always consider the accuracy and error in any of these models as stakeholders may use the models to inform organisational business investment in the best areas, for the most important equipment and in optimising allocation of funding across departments or sectors.

References

BROWNLEE, J. 2020a. *How to Use XGBoost for Time Series Forecasting* [Online]. Available: https://machinelearningmastery.com/xgboost-for-time-series-forecasting/ [Accessed 9 August 2023].

BROWNLEE, J. 2020b. *Nested Cross-Validation for Machine Learning with Python* [Online]. Machine Learning Mastery. Available: https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/ [Accessed 6 August 2023].

HYNDMAN, R. J. & ATHANASOPOULOS, G. 2021. Forecasting: Principles and Practice. 3rd ed. Melbourne, Australia: OTexts.com/fpp3.

MARTIN, R. J., MOUSAVI, R. & SAYDAM, C. 2021. Predicting emergency medical service call demand: A modern spatiotemporal machine learning approach. *Operations Research for Health Care,* 28**,** 100285.

MUKHOPADHYAY, A., PETTET, G., VAZIRIZADE, S. M., LU, D., JAIMES, A., SAID, S. E., BAROUD, H., VOROBEYCHIK, Y., KOCHENDERFER, M. & DUBEY, A. 2022. A Review of Incident Prediction, Resource Allocation, and Dispatch Models for Emergency Management. *Accident Analysis & Prevention,* 165**,** 106501.

MUKHOPADHYAY, A., PETTET, G., VAZIRIZADE, S. M., VOROBEYCHIK, Y., KOCHENDERFER, M. & DUBEY, A. 2020. A Review of Emergency Incident Prediction, Resource Allocation and Dispatch Models.

QFES 2023a. Block A Meta Data. *QFES Incident Meta Data.* Brisbane, QLD, Australia: Queensland Fire and Emergency Services.

QFES. 2023b. *QFES Incident Data* [Online]. Brisbane, QLD, Australia: Queensland Government. Available: https://www.data.qld.gov.au/dataset/qfes-incident-data [Accessed 6 July 2023].