|  |
| --- |
| OTAGO POLYTECHNIC AUCKLAND INTERNATIONAL CAMPUS |
| Project Implementation and Evaluation |
| Crime Prediction Using Predictive Analytic |
|  |
|  |
| **April Love Naviza**  **Vimitaben Mukeshchanadra Vaidya**  **Wisanu Boonrat** |

**5/25/2020**

|  |
| --- |
|  |

**Graduate Diploma in Information Technology**

# Executive Summary

Machine Learning is the process of training a computer system on how to make an accurate prediction using historical records. Our project is about predictive policing, where we focused on predicting the crime risk in New Zealand. This implementation document will show the process of how we developed a machine learning model using different algorithms.

There are 19 sections in this document. Just after the introduction, chapter 2 gives a summary of the literature review that discusses crime factors, data transformation, and algorithms. Those studies gave us ideas on what are the feature factors we have to consider and identify which algorithm performed best for this type of project.

Several project management tools have been used. We have utilized Jira to manage the sprints, identify the tasks, record timesheet, and embellish the stories. All codes have been uploaded in Bitbucket to make it accessible to all the team members. While Trello, Gantt chart, WhatsApp, and Teams are for communication and documentation storage. Our codes are written in Python programming language, wherein we have utilized different IDE that will be identified in sections 5. And in the setup workspace section, the programming tools and library that have been installed will be defined.

The implementation part is divided into five parts, and wherein each section will show how we have come up with the baseline score for all the dataset acquired from different countries. We have focused and spent time in data engineering and implemented 12 techniques before developing the model. All the details are in section 10.

After having multiple prototypes using the post-processed datasets, where able to develop a model would provide a more accurate result. To test and be confident with our final model, we have applied cross-validation, all the

In the final sprint, we were able to explore deep learning by utilizing the post-processed dataset to TensorFlow. This allows us to compare and support our machine learning models. At the end of this project, we build a simple user interface to show how the model performed and predict the crime risk for a specific location, day, and crime type.

# Table of Contents

[Executive Summary 2](#_Toc46605043)

[Table of Contents 3](#_Toc46605044)

[Table of Tables 5](#_Toc46605045)

[Table of Figures 6](#_Toc46605046)

[1 Introduction 8](#_Toc46605047)

[2 Literature Review 8](#_Toc46605048)

[2.1 Features and factors of crime trends 8](#_Toc46605049)

[2.2 Data Transformation 9](#_Toc46605050)

[2.3 Algorithms 10](#_Toc46605051)

[3 Project Management Tools 11](#_Toc46605052)

[3.1 Jira Board 11](#_Toc46605053)

[3.1.1 Sprint Management 11](#_Toc46605054)

[3.1.2 Story, Task, and Subtask Management 11](#_Toc46605055)

[3.1.3 Requirement Management 12](#_Toc46605056)

[3.1.4 Changes Management 13](#_Toc46605057)

[3.1.5 Time Tracking 14](#_Toc46605058)

[3.1.6 Task-Updated Notification 15](#_Toc46605059)

[3.2 Trello 16](#_Toc46605060)

[3.3 Microsoft Teams 16](#_Toc46605061)

[4 Development Tools & Programming Language 17](#_Toc46605062)

[5 Workspace Structure 17](#_Toc46605063)

[5.1 High-Level Project Structure 17](#_Toc46605064)

[5.2 New Zealand Related File Structure 18](#_Toc46605065)

[6 Setup Workspace 20](#_Toc46605066)

[6.1 Install Anaconda or Miniconda 20](#_Toc46605067)

[6.2 Create Conda Environment (env\_dev) 20](#_Toc46605068)

[6.3 Install Python Libraries 21](#_Toc46605069)

[6.4 Setup Google Colab 22](#_Toc46605070)

[6.5 Setup the Spyder Project 23](#_Toc46605071)

[6.6 Test the Workspace 24](#_Toc46605072)

[7 Implementation 26](#_Toc46605073)

[7.1 Crime Datasets 26](#_Toc46605074)

[7.2 Descriptive Analysis 27](#_Toc46605075)

[7.3 Data Visualization 27](#_Toc46605076)

[7.4 Baseline Feature and Target Variables 29](#_Toc46605077)

[7.5 Baseline Scores for All Datasets 29](#_Toc46605078)

[7.6 Prototyping with New Zealand (2) Dataset 29](#_Toc46605079)

[8 Feature Selection 30](#_Toc46605080)

[9 Detect Invalid Data 31](#_Toc46605081)

[10 Data Pre-Processing 32](#_Toc46605082)

[10.1 Rename Columns 32](#_Toc46605083)

[10.2 Drop Unused Columns 32](#_Toc46605084)

[10.3 Drop Invalid Area Unit 32](#_Toc46605085)

[10.4 Extract Month and Year 32](#_Toc46605086)

[10.5 Impute Day of Week 33](#_Toc46605087)

[10.6 Convert Day of Week to Day Index 33](#_Toc46605088)

[10.7 Impute Hour of Day 34](#_Toc46605089)

[10.8 Generate Crime Factors 35](#_Toc46605090)

[10.8.1 MONTH\_AREA\_CRIME\_COUNT 35](#_Toc46605091)

[10.8.2 DAY\_AREA\_CRIME\_COUNT 36](#_Toc46605092)

[10.8.3 3\_DAY\_AREA\_CRIME\_MEAN 36](#_Toc46605093)

[10.8.4 MONTH\_AREA\_CRIME\_TYPE\_COUNT 37](#_Toc46605094)

[10.8.5 Merge Crime Factors into the Dataframe 37](#_Toc46605095)

[10.8.6 DAY\_AREA\_CRIME\_RATIO 37](#_Toc46605096)

[10.8.7 MONTH\_AREA\_CRIME\_RATIO 37](#_Toc46605097)

[10.8.8 CRIME\_RATIO 38](#_Toc46605098)

[10.8.9 CRIME\_RANK 38](#_Toc46605099)

[10.8.10 RISK 38](#_Toc46605100)

[10.8.11 Unit Testing 38](#_Toc46605101)

[10.9 Group the weapon type. 40](#_Toc46605102)

[10.10 Convert Month to Quarter 41](#_Toc46605103)

[10.11 Convert Hour to Partition 42](#_Toc46605104)

[10.12 Export a new CSV file 43](#_Toc46605105)

[11 Predictive Modelling 44](#_Toc46605106)

[12 Cross-Validation 45](#_Toc46605107)

[13 Deep Learning: TensorFlow 46](#_Toc46605108)

[14 Results Evaluation 51](#_Toc46605109)

[14.1 Score Comparison 51](#_Toc46605110)

[14.1.1 Predict the Crime Type 51](#_Toc46605111)

[14.1.2 Predict the Risk of Top 3 Crime Types 52](#_Toc46605112)

[14.1.3 Predict the Risk of All Crime Types 53](#_Toc46605113)

[14.2 Results Visualisation 54](#_Toc46605114)

[14.2.1 Confusion Matrix 54](#_Toc46605115)

[14.2.2 Receiver Operating Characteristic Curve (ROC) 56](#_Toc46605116)

[14.2.3 Precision-Recall Curve (PR) 58](#_Toc46605117)

[14.3 Quantitative Evaluation 60](#_Toc46605118)

[15 Application User Interface 61](#_Toc46605119)

[15.1 Using Microsoft Power BI 61](#_Toc46605120)

[16 Conclusion 63](#_Toc46605121)

[17 Future Work 63](#_Toc46605122)

[18 Discussion 64](#_Toc46605123)

[18.1 Basic Data Science and Coding Skills 64](#_Toc46605124)

[18.2 Research & Believe in the Feasible Solution 64](#_Toc46605125)

[18.3 Plot Graphs 64](#_Toc46605126)

[19 Reference List 66](#_Toc46605127)

[20 Appendices 67](#_Toc46605128)

[20.1 Jira Board & Trello 67](#_Toc46605129)

[20.2 Shared Documents 67](#_Toc46605130)

[20.3 Source Code 67](#_Toc46605131)

[20.4 Web Resources & Tools 67](#_Toc46605132)

[20.5 New Zealand Dataset (Final version) 68](#_Toc46605133)

# Table of Tables

[Table 1. Tools & programming language. 17](#_Toc46605134)

[Table 2. Main folders of the project. 18](#_Toc46605135)

[Table 3. Folders and file structure for the crime prediction in New Zealand. 18](#_Toc46605136)

[Table 4. Test the workspace setup by executing the code. 24](#_Toc46605137)

[Table 5. Crime datasets and columns. 26](#_Toc46605138)

[Table 6. Crime datasets and features' details. 27](#_Toc46605139)

[Table 7. Define feature and target variables for the baseline. 29](#_Toc46605140)

[Table 8. Baseline scores for all datasets. 29](#_Toc46605141)

[Table 9. Feature selection with ExtraTrees. 30](#_Toc46605142)

[Table 10. Feature selection with SelectKBest. 30](#_Toc46605143)

[Table 11. Feature selection with Mutual Information. 30](#_Toc46605144)

[Table 12. Feature selection with Chi Contingency. 30](#_Toc46605145)

[Table 13. Invalid data in New Zealand dataset. 31](#_Toc46605146)

[Table 14. Scores by TensorFlow. 46](#_Toc46605147)

[Table 15. Baseline scores of New Zealand crime dataset when the target variable is the crime type. 51](#_Toc46605148)

[Table 16. Scores of New Zealand’s top 3 crime dataset when the target variable is the crime risk. 52](#_Toc46605149)

[Table 17. Scores of New Zealand crime dataset when the target variable is the crime risk (all crime types). 53](#_Toc46605150)

[Table 18. Scores of the New Zealand crime dataset when the target variable is the crime risk (all crime types without maz\_depth). 53](#_Toc46605151)

# Table of Figures

[Figure 1. Predictive Policing's Jira board. 11](#_Toc46605152)

[Figure 2. Jira board - Active sprint view. 12](#_Toc46605153)

[Figure 3. Jira's user story. 13](#_Toc46605154)

[Figure 4. Bitbucket's commits in the user story. 14](#_Toc46605155)

[Figure 5. Jira's time tracking. 14](#_Toc46605156)

[Figure 6. Story's work log. 15](#_Toc46605157)

[Figure 7. Jira's task notifications via email. 15](#_Toc46605158)

[Figure 8. Trello - Sprint Retrospective board. 16](#_Toc46605159)

[Figure 9. Shared documents in MS Teams. 16](#_Toc46605160)

[Figure 10. Workspace's structure. 18](#_Toc46605161)

[Figure 11. Anaconda installation. 20](#_Toc46605162)

[Figure 12. New file in Google Colab. 22](#_Toc46605163)

[Figure 13. Google Colab links to google drive. 22](#_Toc46605164)

[Figure 14. How to save the file in Google Colab. 23](#_Toc46605165)

[Figure 15. Google Colab in Github. 23](#_Toc46605166)

[Figure 16. New project in Spyder. 24](#_Toc46605167)

[Figure 17. Project view in Spyder IDE. 24](#_Toc46605168)

[Figure 18. Flask application is running. 25](#_Toc46605169)

[Figure 19. Visualize all variables in the dataset (New Zealand). 28](#_Toc46605170)

[Figure 20. Visualize the outlier in the dataset (New Zealand). 28](#_Toc46605171)

[Figure 21. Transform the UNKNOWN data into DAY\_OF\_WEEK. 33](#_Toc46605172)

[Figure 22. Transform the DAY\_OF\_WEEK into DAY. 34](#_Toc46605173)

[Figure 23. Transform the invalid hours into the valid hours. 34](#_Toc46605174)

[Figure 24. Total crime grouped by month and territorial authority. 35](#_Toc46605175)

[Figure 25. Total crime grouped by day and area unit. 36](#_Toc46605176)

[Figure 26. 3 days mean of the crime. 36](#_Toc46605177)

[Figure 27. Total crime grouped by crime type 37](#_Toc46605178)

[Figure 28. The total crime at each risk level. 38](#_Toc46605179)

[Figure 29. The equations of crime ratios. 39](#_Toc46605180)

[Figure 30. The total day crime ratio grouped by month and area\_1 should equal to 1. 39](#_Toc46605181)

[Figure 31. The total month crime ratio grouped by month and area\_1 should equal to 1. 40](#_Toc46605182)

[Figure 32. Group the weapon types. 41](#_Toc46605183)

[Figure 33. Group months into quarters. 41](#_Toc46605184)

[Figure 34. Group hours into hour partitions. 42](#_Toc46605185)

[Figure 35. The variables in the original New Zealand crime dataset. 43](#_Toc46605186)

[Figure 36. The variables after feature engineering in the New Zealand dataset. 43](#_Toc46605187)

[Figure 37. Cross-validation in sklearn. 46](#_Toc46605188)

[Figure 38. Visualize the model by TensorFlow Board. 50](#_Toc46605189)

[Figure 39. TensorFlow's scores by epochs. 50](#_Toc46605190)

[Figure 40. Confusion matrixes of all models. 55](#_Toc46605191)

[Figure 41. ROC curves of all models. 57](#_Toc46605192)

[Figure 42. PR Curves of all models. 59](#_Toc46605193)

[Figure 43. The prototype of an application user interface for the crime prediction in New Zealand. 61](#_Toc46605194)

[Figure 44. The predictive models’ integration with Power BI. 62](#_Toc46605195)

[Figure 45. ROC curves reveal the overfitting. 65](#_Toc46605196)

[Figure 46. The location type found only in the Unlawful Entry crime. 65](#_Toc46605197)

# Introduction

This project is all about crime risk prediction in New Zealand. All the requirements, tools, processes, and computation in training a Machine Learning model will be discussed in this document.

In the initials stage, there are 6 different crime datasets acquired from other counties such as Boston, Denver, London, San Francisco, New Zealand version 1, and version 2. Using these records, we have come up with a baseline score without applying any feature engineering. Among the datasets the New Zealand version 2 got the highest accuracy rate, however, the result using various algorithms is only ranging from 50-60% accuracy. This gave us the opportunity to train and improve our model.

In the next sprint, we focused on crime factors such as Day, Month, and Territorial Authority. And applied data transformation to get the risk level. All the steps, formula, and details are discussed in the data pre-processing section below. This technique resulted in a higher accuracy rate from 50-60% it improved to an 80-90% rate.

We have also looked at deep learning by exploring PyTorch and TensorFlow. Comparing the result to the Machine Learning model, it gave us almost the same. This shows that in future work if the new dataset is more complex, the Deep Learning model can be utilized in predicting crime risk.

For the result visualization, we have developed a simple user interface that allows users to select data from the list and it will present a map to show the crime risk level in a specific territorial area. Another way to visualize our result, we have to integrate our model to a commercial platform like Microsoft Power BI. Both applications predict the crime risk with 80-90% accuracy rate as what we aim to attain by the end of this project.

# Literature Review

## Features and factors of crime trends

Many studies have been published about predictive policing, especially to those countries where the crime rate is rapidly increasing every year. The common objective of the researchers is to develop a model that will predict the crime occurrence by using machine learning or deep learning technology (Woo Kang & Bong Kang. 2017).

Schneider (2015) mentioned in his research that there are two methods for predicting crime events, qualitative, and quantitative approach. Among the two, the quantitative method is commonly used to predict the crime rate trend through the time series model. This approach analyses the factors of the historical records such as Demographic (e.g., age, gender, and population) and Macro-economic features (e.g., unemployment rates) to crime prediction. These two variables have been the determining factor of crime rates (Deadman, 2002). Though there are still many other features that can also influence the future crime event, in another study conducted in New Zealand, it was tested that weather temperature and precipitation showed a significant effect on violent and property crimes (Horrocks & Menclova, 2011). On the other hand, Rumi et al. 2018 introduced dynamic features and highlighted its advantages compared to static or historical features. It recommends to utilizing social media and human mobility in crime event prediction, for example, the check-ins of Foursquare users and visiting records, this captures the potential crime event immediately based on their routine and activities. In conclusion, by combining the dynamic and static features, it can improve the prediction accuracy performance. (Rumi, Deng & Salim. 2018).

Traditionally historical crime dataset is being used to analyse the crime patterns. However, many authors considered adding crime factor features to the original dataset, such as population census, social-economic factors (like average income), and weather data that includes precipitation and snow levels, wind speed, maximum and minimum temperatures (Berrada & Martegiani). Another study that uses the Denver crime incident dataset also merged the Neighborhood Demographic details and focused on spatial and temporal criminal hotspots to produce a more accurate crime prediction (Almanie et al., 2015).

In deep learning, it was proposed to a feature-level data fusion method with an environmental context based on a deep neural network (DNN). The dataset contains a collection from various online databases of crime statistics, demographic and meteorological data, and images collected from Google Street View. The authors concluded that aside from conventional methods of using population, income, and education from the different datasets, it is substantial to consider the environmental context information to predict the crime occurrence accurately.

## Data Transformation

Feature selection and data transformation is the most critical process in model designing. It improves model accuracy and reduces overfitting (Kaushik, 2016).[[1]](#footnote-2) Many pre-processing data techniques were performed in different studies. One of the first steps in feature selection is handling the missing values, and these variables can be drop if the percentage of the missing value is more than the threshold that has been set (Sharma, 2018) [[2]](#footnote-3) or if the values are not key attributes (Almanie et al. 2015).

It is common in a crime dataset that there are numerous types of crimes. Researches like Berrada et al., 2016 reduced the crime type by and grouping them into violent crimes, property crimes, and other crimes, while for (Pradhan,2018), he extracted the information from the crime description column and grouped them to a new category. The other authors applied dimension reduction using attribute subset selection wherein only related and key attributes will be selected (Almanie et al. 2015), while others preferred to use only the top crime classes to have a balanced dataset (Yuki et al. 2019).

Date and time are also significant features, and they come in different formats. In order to make it simple and standard, the date has been split into “Day”, “Month”, “Year”, “Hour”, “Minute”, “Second” attributes (Yuki et al. 2019). Another option is to create a variable representing the number of crimes in each area by 24h, 7days, 1 month, and 4 months (Berrada & Martegiani). While for another study, Almanie et al. 2015 have adopted the military time system and mapped it to a 4hrs interval, which reduces the diversity of the crime time feature. Having this standard and making the column name unified becomes beneficial when dealing with multiple datasets.

## Algorithms

It is critical to identify the appropriate predictive modelling approach, depending on the dataset that has been acquired and pre-processed. In the study of (Almaw et al. 2018), they explored several data mining algorithms and summarized the techniques implemented in crime prediction from other researchers. In the table comparison, it was observed that among the common algorithms used, such as Naïve Bayes, Decision Tree, SVM, and Neural Network, the Naive Bayes demonstrated more accurate in prediction. The study concluded that through a combination of algorithms or by the integration of multiple models, better accuracy could be achieved.

Almanie et al. 2015 chose to use the algorithms, Apriori, Naïve Bayes, and Decision Tree in their study. Using the Denver local criminal records, the Naïve Bayes with 51% give a high accuracy rate compared to the Decision tree (42%) though both have the same performance in terms of running time. Decision Tree created a very complicated tree that cannot generalize the data. The strength of Apriori, that has been observed in this study is its readiness, and easiness of use and implementation. On the contrary, it takes a very long running time.

In one study, the Logistic Regression and Random Forest algorithm has been compared using the dataset that includes Boston local crimes, census, and weather data. Between the two, Logistic regression got the highest result wherein fitting a Random Forest model on the data yielded only 71% accuracy (Berrada & Martegiani, 2016).

Tree-based algorithms are commonly used in predictive analysis. Yuki et al. 2019 explored and compared the Random Forest, Decision Tree, Extra Tree, Bagging, and AdaBoost algorithms using the Chicago crime events from the Chicago Police Department’s CLEAR system. Overall, the tree-based algorithms showed a predictive result closer to the actual scores. While the Bagging method performed better using time and location feature, in contrast to AdaBoost.

# Project Management Tools

## Jira Board

Jira has provided a cloud service to manage a software project for free for 10 users. Jira board is suitable for an agile project, which is planned to be completed by sprints.

### Sprint Management

The advantage of managing the stories and tasks by a sprint is that the development team and Product Owner can manage an agreement of the output of each sprint. If there is an urgent requirement which is needed to be completed, the task can be created and be put into the current sprint or next sprint depending on team agreement.

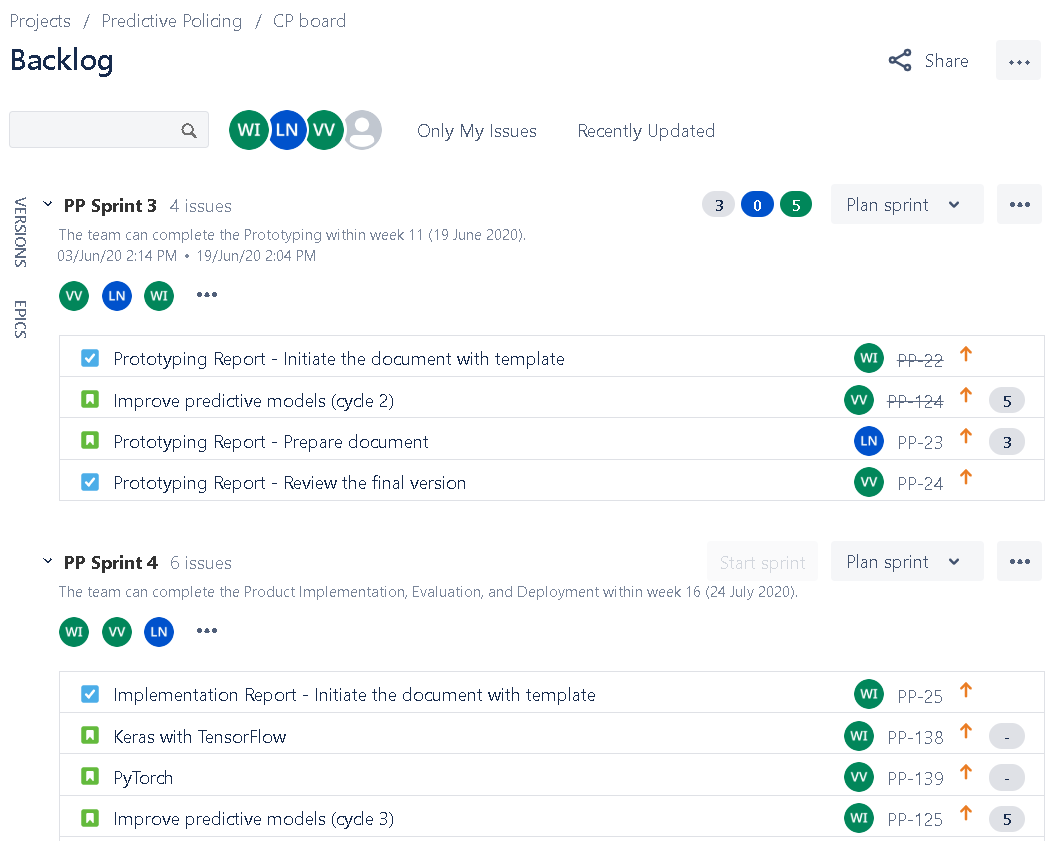


Figure 1. Predictive Policing's Jira board.

### Story, Task, and Subtask Management

The tasks’ assignees and statuses are visualized in the Jira board. The development team can change the status easily by moving the task’s card into the next status. When some tasks are stuck in the same status longer than the expected man-hour, it means that the development team needs to immediately take some action into the task to resolve the problem.

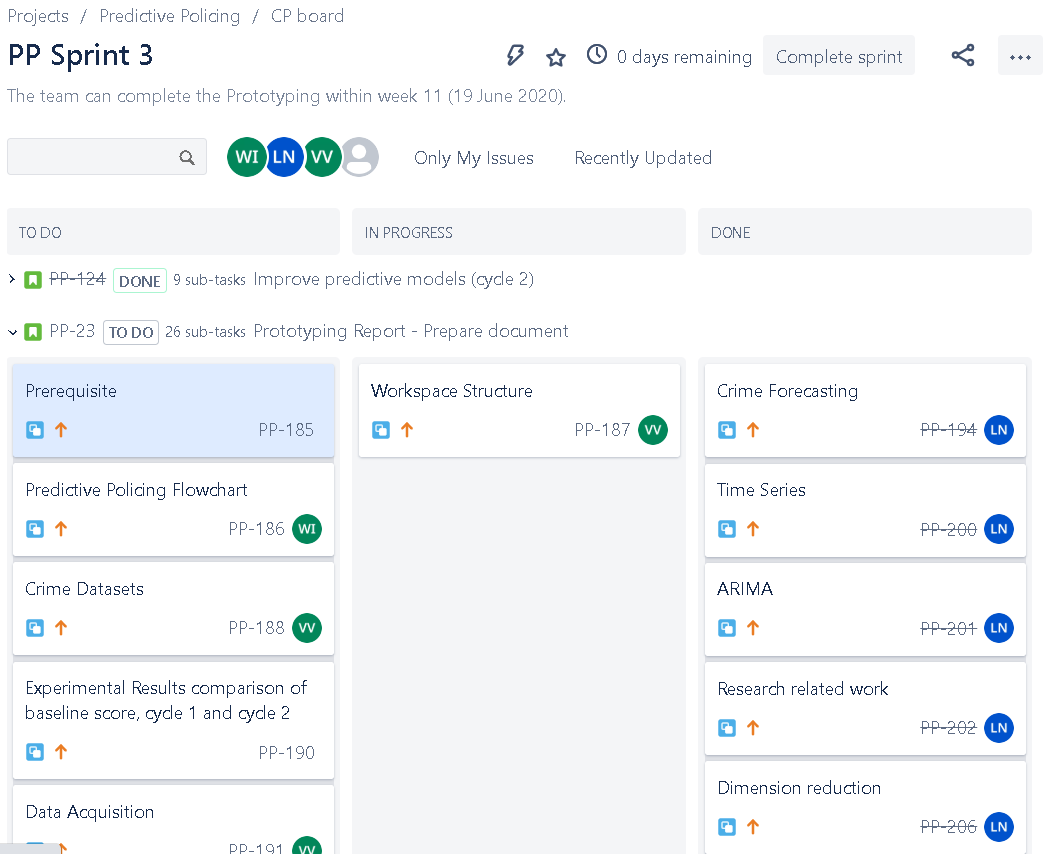


Figure 2. Jira board - Active sprint view.

### Requirement Management

In each task, the requirements can be described using wikitext formatter and the related documents can be attached. This feature encourages the agile team to have a common understanding of every task of the sprint.

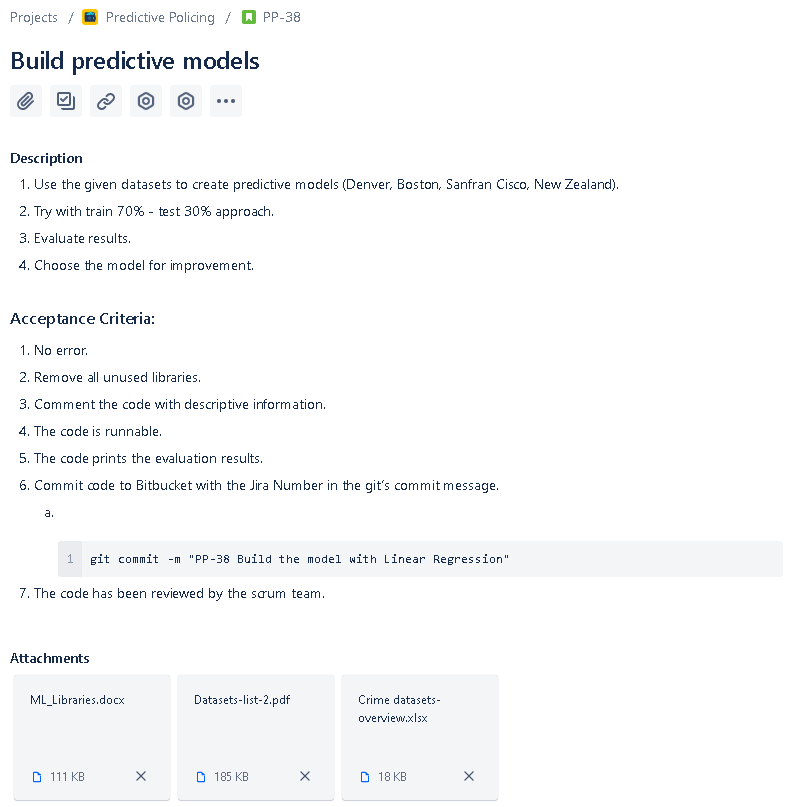


Figure 3. Jira's user story.

### Changes Management

The Jira board can be linked to the Bitbucket repository. By using a specific pattern of Git commit message, the code changes will link to the story and can be tracked.

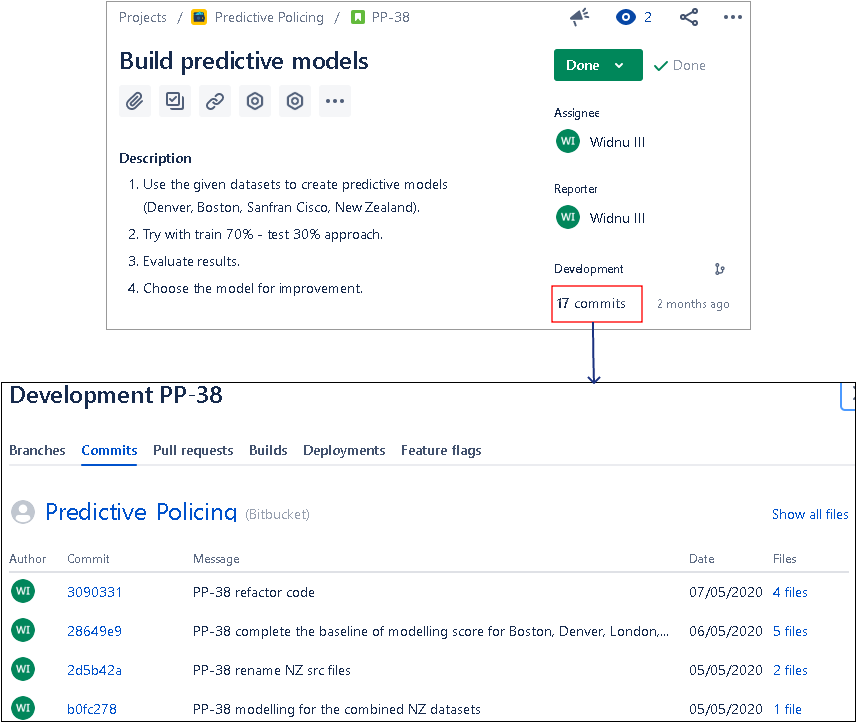


Figure 4. Bitbucket's commits in the user story.

### Time Tracking

In each task, the developers can log the time spent and there is a Work log button at the end of the web page to review the work logs.

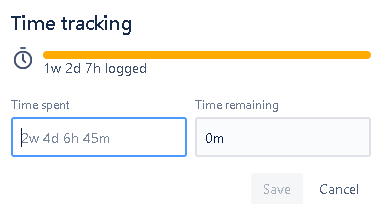


Figure 5. Jira's time tracking.

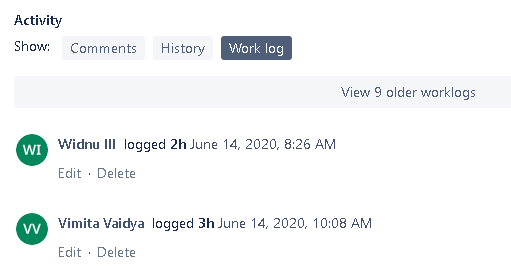


Figure 6. Story's work log.

### Task-Updated Notification

The development team can have an email notification when someone makes a change into the task.

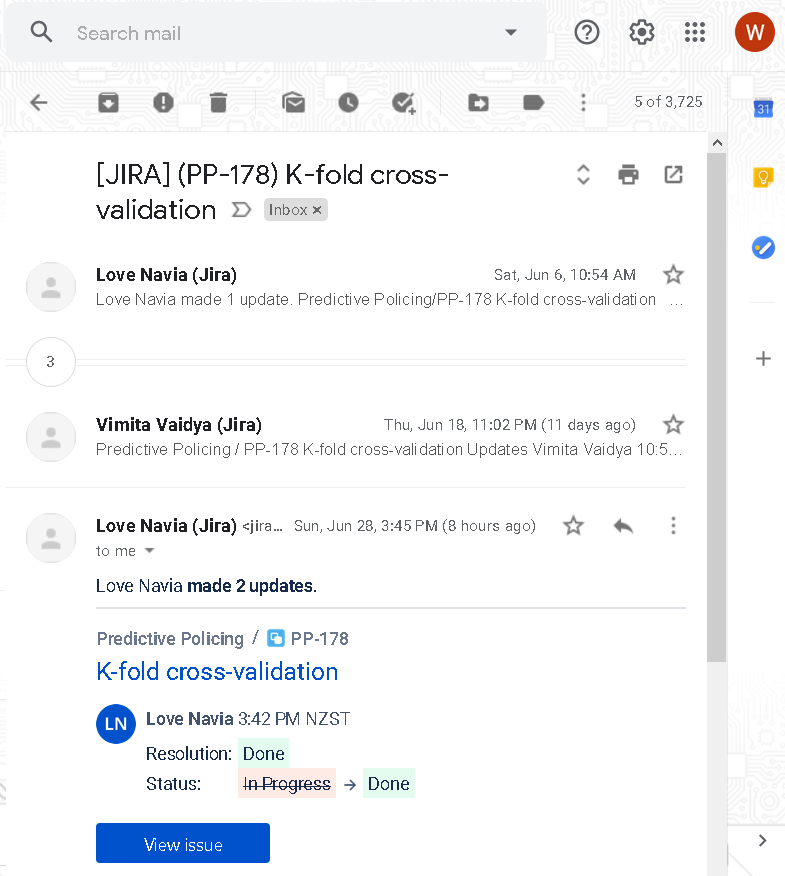


Figure 7. Jira's task notifications via email.

## Trello

Trello has provided an online-board application, in which the user can create cards for notes, tasks, and shared files. For this project, we use Trello to organize the Sprint Retrospective and Knowledge Sharing.

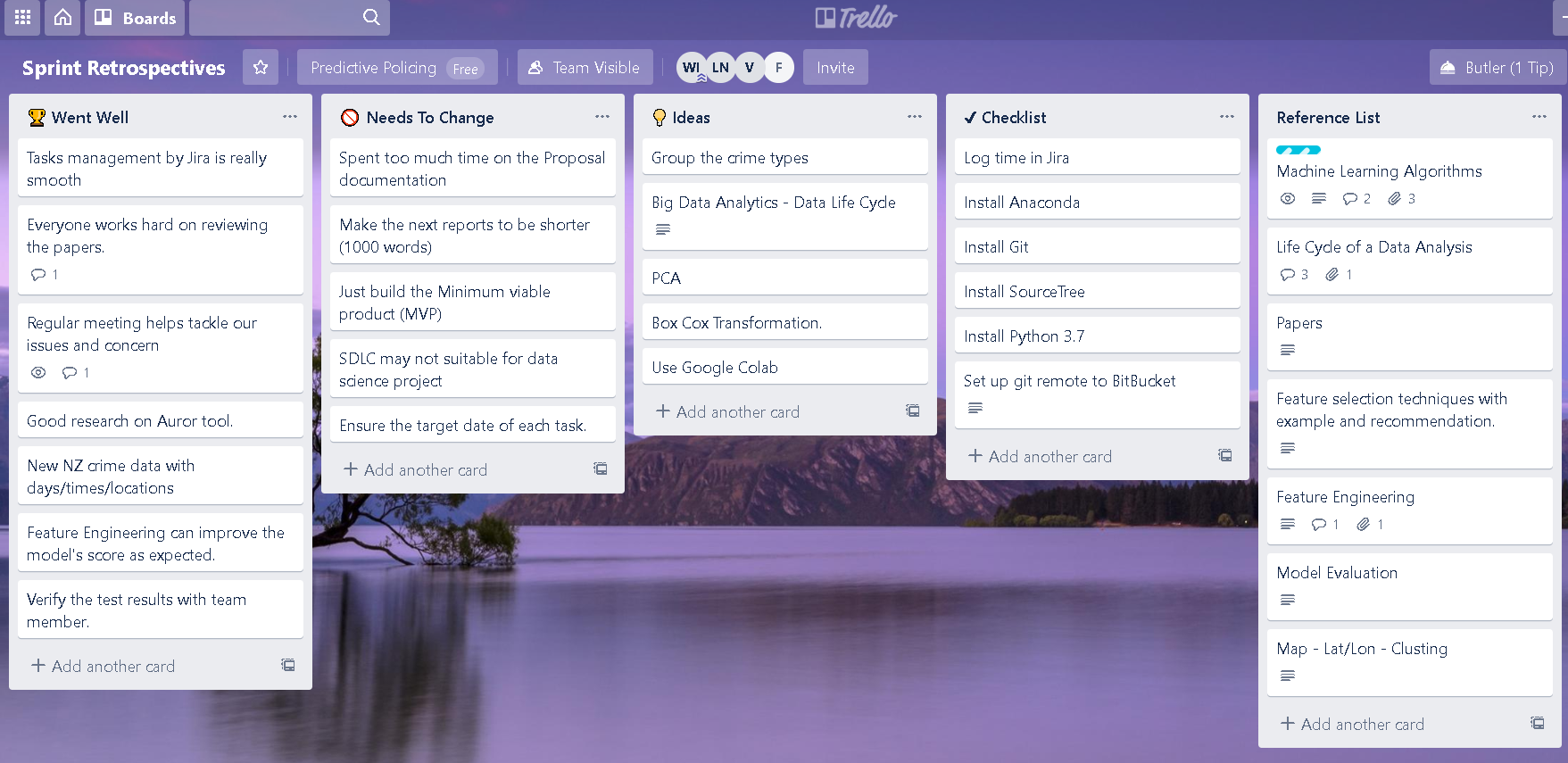


Figure 8. Trello - Sprint Retrospective board.

## Microsoft Teams

Microsoft Teams is a collaboration platform with many features such as chat rooms, video meetings, and document management. All documents in this project are uploaded and shared via this tool.

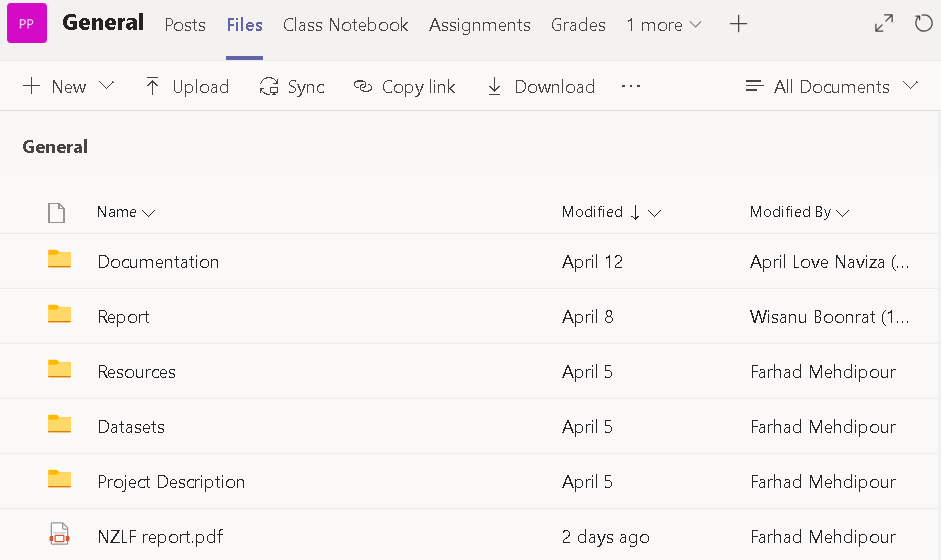


Figure 9. Shared documents in MS Teams.

# Development Tools & Programming Language

The development team uses Python programming language to implement this project rather than R because Python is easier to understand and learn. There are also many Python communities and shared packages of Machine Learning features to be installed for project development. To create a working environment, there is several prerequisite tools as shown in the following table.

Table 1. Tools & programming language.

|  |  |  |
| --- | --- | --- |
| Tools | Description | Links |
| Anaconda Individual Edition | Package and environment manager for Python/R data science project. It provides both the UI and command line to manage a project. | <https://docs.anaconda.com/anaconda/> |
| Miniconda | A minimal version of Anaconda. It provides only the command line to manage a project. | <https://docs.conda.io/en/latest/miniconda.html> |
| Spyder | An integrated development environment (IDE) for a data science project. It is recommended to install this tool via Anaconda. | <https://www.spyder-ide.org/> |
| PyCharm | PyCharm professional version is for a data science project. It is the foundation of JetBrains, and Anaconda also allow to install this tool. | <https://blog.jetbrains.com/pycharm/> |
| Python 3.6-3.8 | The specific version of 3.6-3.8 is required for Deep Learning development with TensorFlow2. | <https://www.python.org/downloads/> |
| Python Package | There is a set of packages needed to be installed to develop this project. This will be described in the Setup Workspace section. | <https://anaconda.org/anaconda/repo> |
| Bitbucket | A web-based version control repository, which can be plugged into the Jira board to track the code changes. This is the main repository for this project. | <https://bitbucket.org/product/> |
| Google Colab | A cloud-based workspace to develop Machine Learning and Deep Learning using Python libraries such as Keras, TensorFlow, PyTorch, and OpenCV. | <https://colab.research.google.com/notebooks/intro.ipynb> |
| GitHub | Another web-based version control repository. GitHub can be bundled with Google Colab to share and run the code on their cloud services. | <https://github.com/> |
| Git | A distributed version control system, which provides a command-line interface to manage versions of the source code. | <https://git-scm.com/downloads> |

# Workspace Structure

The workspace consists of 4 folders to store files, which are data, source code, configuration, and log files. Although this project has related to the crime prediction in different locations, the naming convention in file-level is applied by appending the location’s name into the file’s name such as nz\_crime\_dataset.csv, boston\_crime\_dataset.csv, and Denver\_crime\_dataset.csv.

## High-Level Project Structure

The project consists of data, source code, output, configuration, and log folders.

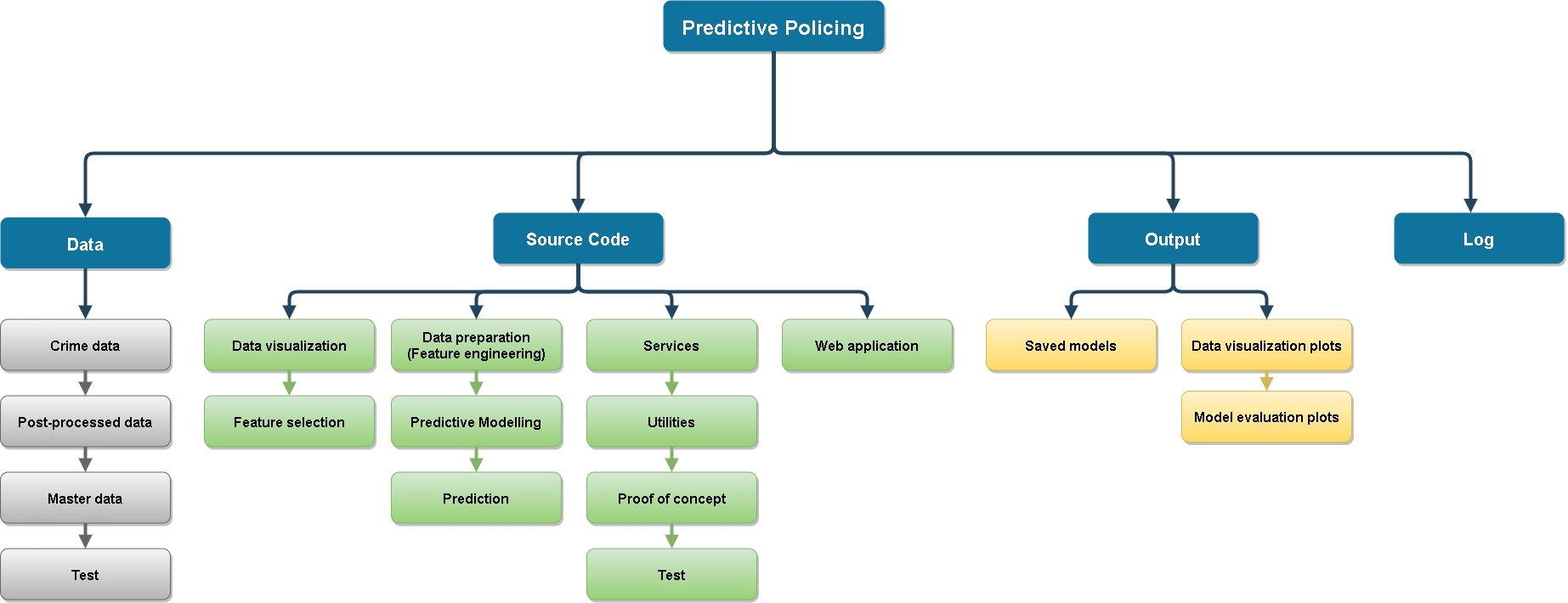


Figure 10. Workspace's structure.

Table 2. Main folders of the project.

|  |  |
| --- | --- |
| Folders | Descriptions |
| /predictive\_policing/data | Store all datasets (New Zealand, Boston, Denver, London, and San Francisco) |
| /predictive\_policing/src | Source code for this project |
| /predictive\_policing/output | Store the graphs of descriptive analysis, prediction results, and the saved Machine Learning models and encoders. |
| /predictive\_policing/conf | Python logger configuration |
| /predictive\_policing/log | Store application log messages. |

## New Zealand Related File Structure

To execute a prediction using the New Zealand crime dataset, there is a set of specific files and folders.

Table 3. Folders and file structure for the crime prediction in New Zealand.

|  |  |  |  |
| --- | --- | --- | --- |
| Modules | Paths | Files | Descriptions |
| Dataset | /predictive\_policing/data/NZ datasets/ | nz\_victim\_timeplace.csv | Original New Zealand crime dataset. |
| /predictive\_policing/data/post\_processed/ | nz\_crime\_dataset\_6\_crimes.csv  nz\_crime\_dataset\_6\_crimes\_with\_latlong.csv  nz\_area.csv | Cleaned the New Zealand crime dataset. The dataset with latitude and longitude will be used in the web application to plot the map. The nz\_area.csv has only the location data to be used in UI. |
| Data Visualization | /predictive\_policing/src/visualization/ | pp\_nz\_victim\_timeplace.py | Generate graphs to visualize data. |
| Feature Selection | /predictive\_policing/src/feature\_selection/ | nz\_chisquare.py  nz\_extratree.py  nz\_mutual\_info.py  nz\_selectkbest.py | Calculate scores for choosing feature variables. |
| Pre-Processing | /predictive\_policing/src/preparation/ | nz\_pre\_process.py | Run this file to clean data and generate a CSV file for predictive modelling (nz\_crime\_dataset.csv). |
| Predictive Modelling | /predictive\_policing/src/modeling/ | nz\_mod.py | Run this file to perform a prediction with a set of algorithms. |
| Prediction Testing | /predictive\_policing/src/prediction/ | nz\_load\_predict.py  nz\_load\_predict\_no\_pca.py  nz\_load\_predict\_no\_pca\_mock.py | Run these files to load the models and perform the prediction for testing. |
| Deep Learning: TensorFlow | /predictive\_policing/src/dl/ | nz\_dl\_mod.py | Run this file to perform a prediction with TensorFlow. But we prefer to run it via Google Colab. |
| Application Log Message | /predictive\_policing/log/ | application.log | The log messages will be appended into this file, which includes the classification report, confusion matrix, and accuracy scores of the predictive models. |
| Service Classes | /predictive\_policing/src/services/ | feature\_service.py  graph\_service.py  log\_service.py  model\_service.py | The service classes that contain functions to be called by python files. |
| Web Application | /predictive\_policing/src/web/ | nz\_flask\_app.py  /templates/index.html | The Flask web application to render a user interfaces for the New Zealand crime prediction. |
| Output | /predictive\_policing/output/ | /encoder/  /set\_2\_no\_pca/ | The saved models of OrdinalEncoder and the trained models. |

# Setup Workspace

This section describes how to set up the development environment.

## Install Anaconda or Miniconda

1. Choose the Conda environment to install. Miniconda is recommended if we want a lightweight environment.
   1. Anaconda Individual Edition: <https://www.anaconda.com/products/individual>.
   2. Miniconda: <https://docs.conda.io/en/latest/miniconda.html>
2. Verify results by open the Anaconda Navigator or Anaconda Prompt from the Start menu.

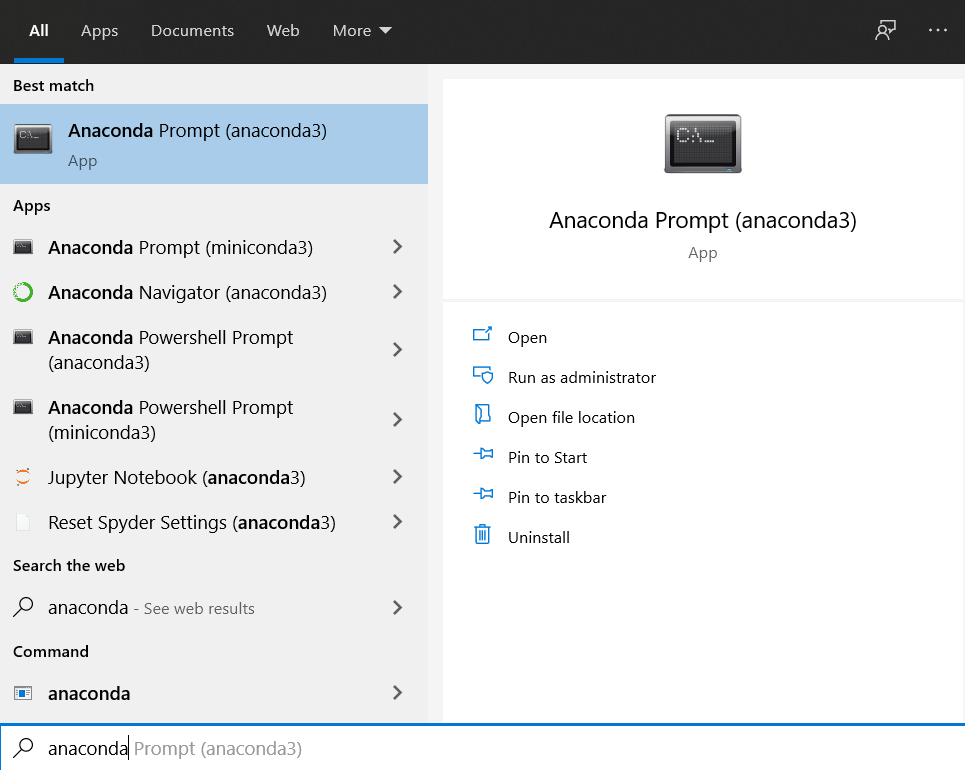


Figure 11. Anaconda installation.

## Create Conda Environment (env\_dev)

1. Open Anaconda Prompt.
2. The environment’s name is ‘env\_dev’. So, we will enter the command:

conda create -n env\_dev tensorflow

1. We can notice the keyword ‘tensorflow’ after the command. This keyword means to create the environment from the TensorFlow template provided by Anaconda. So, this environment will support TensorFlow2.
2. Activate the environment:

conda activate env\_dev

1. Verify the version of python of env\_dev environment. The minimum version is 3.6.

python --version

1. Deactivate the environment by using:

conda deactivate

## Install Python Libraries

If we have chosen to install Anaconda Individual Edition, we do not need to install Spyder IDE by command line because the IDE is already bundled with Anaconda. The manual installation of Spyder IDE is needed only when we choose to install Miniconda.

It is recommended to use ‘conda install’ rather than ‘pip install’ command because Conda can handle library dependencies and manage the packages which may contain software written in any language. The following steps describe how to install the required packages.

1. List the current libraries in the environment.

conda list

1. For Miniconda, install Spyder IDE.

conda install -y -c anaconda spyder

1. If the TensorFlow is not version 2, we need to upgrade it.

pip install -y --upgrade tensorflow

1. Run the following command in Anaconda Prompt to install the required packages for this project.

conda install -c conda-forge matplotlib

conda install -c anaconda numpy

conda install -c anaconda pandas

conda install -c anaconda scikit-learn

conda install -c anaconda seaborn

conda install -c conda-forge xgboost

conda install -c conda-forge imbalanced-learn

*conda install -c conda-forge folium*

*conda install -c conda-forge geopandas*

*conda install -c conda-forge Descartes*

*conda install -c anaconda flask*

1. If we want to update the library’s version, we can use the ‘conda update’ command.

conda update scikit-learn

## Setup Google Colab

The advantage of coding in Google Colab is that we do not need to set up the workspace and libraries because Google Colab has provided the environment that is ready for data science project development. We can also save and share the code via GitHub or Gist to team members for code reviewing and testing. There are a few steps to run the code on Google Colab.

1. Log in the google account, then go to <https://colab.research.google.com>.
   1. The page would be redirected to <https://colab.research.google.com/notebooks/intro.ipynb>.
2. Create a new python file from the menu.
   1. File > New notebook
   2. This file will be saved into the google drive in My Drive/Colab Notebooks/

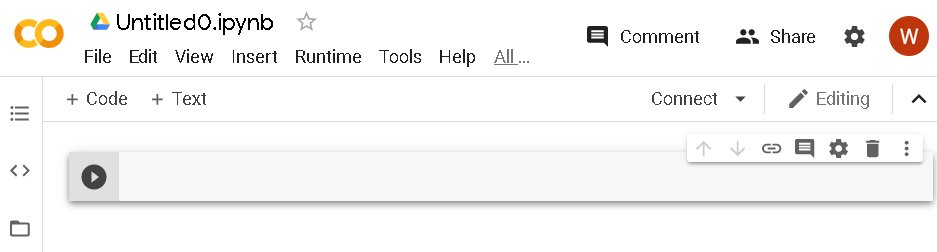


Figure 12. New file in Google Colab.

1. Upload the dataset 'nz\_crime\_dataset.csv' from this [link](https://otagopoly.sharepoint.com/sites/PredictivePolicing/Shared%20Documents/General/Datasets/NZ%20datasets/post_processed/nz_crime_dataset.csv) into your google drive with the following folder structure.
   1. My Drive/Colab Notebooks/dataset/nz\_crime\_dataset.csv

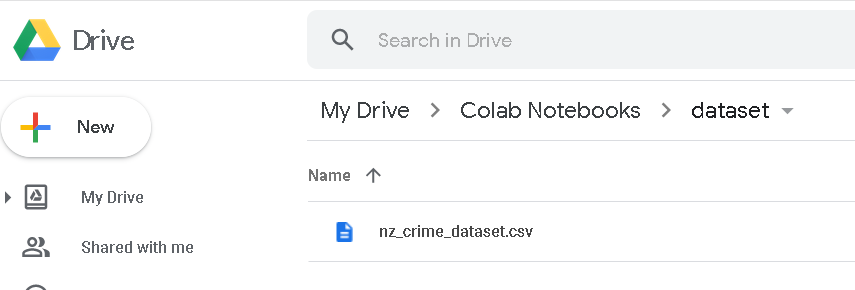


Figure 13. Google Colab links to google drive.

1. We can save the code into GitHub. Or share the code by saving it as a GitHub Gist.

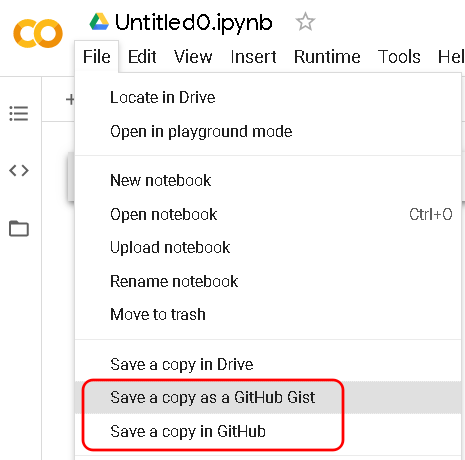


Figure 14. How to save the file in Google Colab.

1. Run the code from GitHub.



Figure 15. Google Colab in Github.

## Setup the Spyder Project

1. Install Git: <https://git-scm.com/downloads>
2. Create a new project folder on the local machine.
3. Right-click in the new project folder and select ‘Git Bash Here’.
4. Clone the source code from Bitbucket into the project folder by the following command.

git clone <https://wisanuboonrat@bitbucket.org/wisanuboonrat/predictive_policing.git>

1. Open Spyder IDE. Ensure to select the correct Conda environment (env\_dev).
2. In the menu bar, select Project > New Project…
3. Create the new project from an existing directory by choosing the location that we have cloned the source code from Bitbucket.

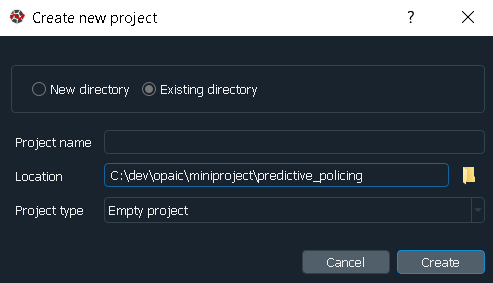


Figure 16. New project in Spyder.

1. The project’s folder structure should appear on the left side of Spyder IDE.

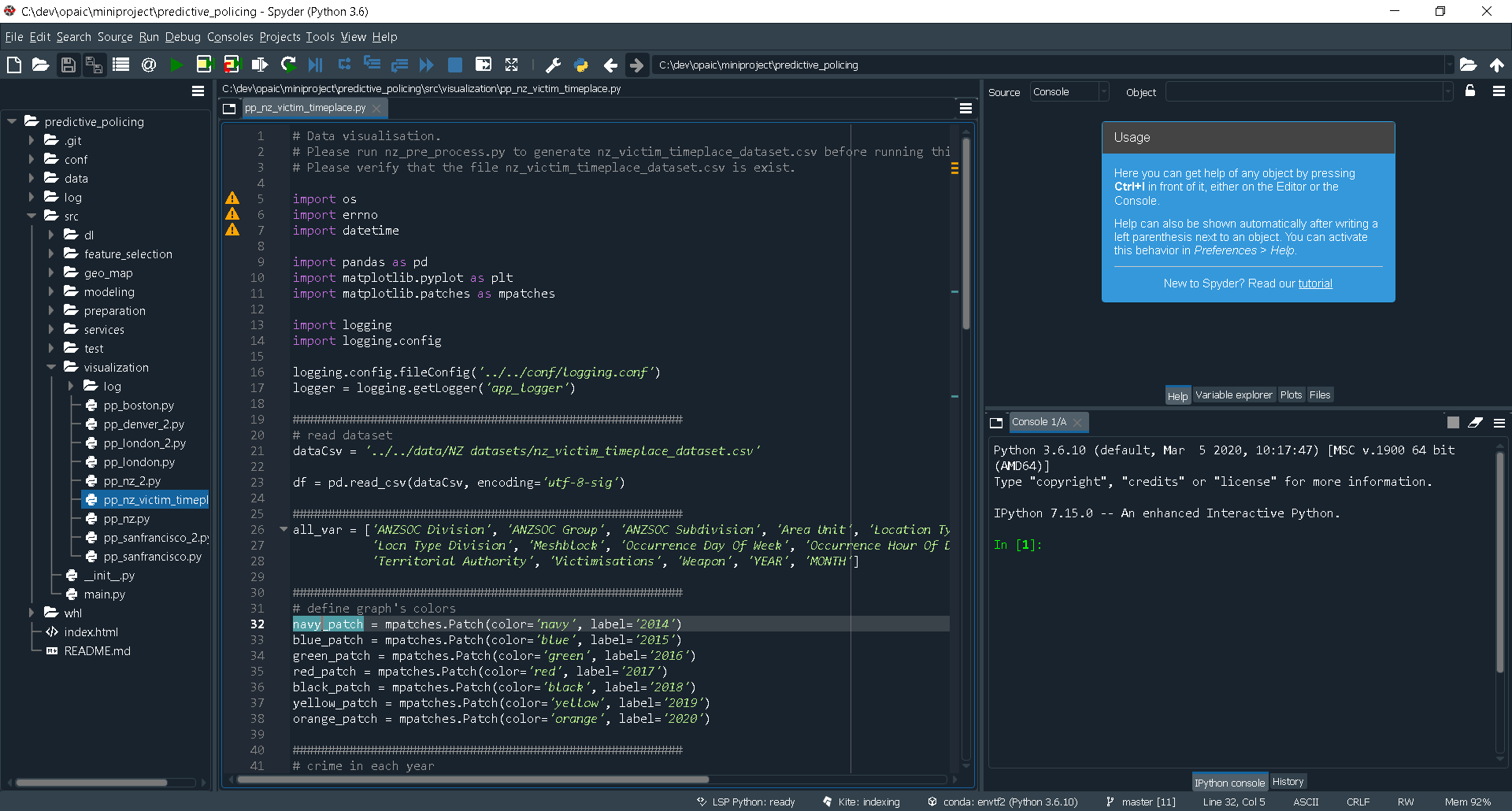


Figure 17. Project view in Spyder IDE.

## Test the Workspace

We can verify the workspace setup by running the following modules. All executions need to be run in the correct sequence.

1. Machine Learning Code

Table 4. Test the workspace setup by executing the code.

|  |  |  |
| --- | --- | --- |
| No. | Actions | Executions |
| 1 | Data pre-processing and feature engineering | /predictive\_policing/src/preparation/nz\_pre\_process.py |
| 2 | Train and test the models | /predictive\_policing/src/modeling/nz\_mod.py |
| 3 | Test the models with unknown data | /predictive\_policing/src/prediction/nz\_load\_predict\_no\_pca\_mock.py |

1. Flask Web Application
   1. Open the command prompt.
   2. Run the following commands

conda activate env\_dev

cd {home\_directory}/predictive\_policing/src/web

set FLASK\_APP=nz\_flask\_app.py

flask run

* 1. The web is ready to serve at <http://localhost:5000>

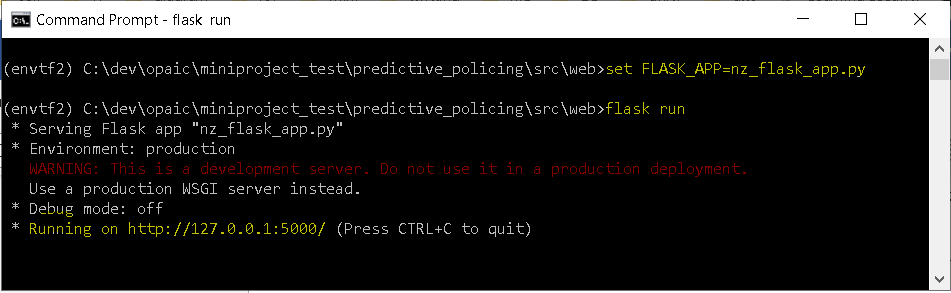


Figure 18. Flask application is running.

# Implementation

## Crime Datasets

There are 5 crime datasets provided in CSV files for this project. The original datasets are the files downloaded from public websites. The post-processed datasets are the cleaned version datasets, which have been processed with feature engineering techniques. The CSV files are located in the following folders:

1. Original datasets: /predictive\_policing/data/{folder\_name\_as\_city\_or\_country}/
2. Post-processed datasets: /predictive\_policing/data/post\_processed/

For the New Zealand crime dataset, we need to acquire a new crime dataset (nz\_victim\_timeplace.csv) because the first dataset does not contain the Day of Week and Hour of Day columns.

The table below shows all 6 original datasets with their columns. Most of them have some common information such as day, month, year, time, and location. Except for London and New Zealand (1) dataset that they do not have day and time values. This missing information would impact the accuracy of the prediction because the inputs for the predictive model are too broad when there are no specific days and time in the crime data.

Table 5. Crime datasets and columns.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Boston | Denver | London | San Francisco | New Zealand (1) | New Zealand (2) (nz\_victim\_timeplace.csv) |
| INCIDENT\_NUMBER | INCIDENT\_ID | lsoa\_code | IncidntNum | Age Group 5Yr Band | ANZSOC Division |
| OFFENSE\_CODE | OFFENSE\_ID | borough | Category | ANZSOC Division | ANZSOC Group |
| OFFENSE\_CODE\_GROUP | OFFENSE\_CODE | major\_category | Descript | ANZSOC Group | ANZSOC Subdivision |
| OFFENSE\_DESCRIPTION | OFFENSE\_CODE\_EXTENSION | minor\_category | DayOfWeek | ANZSOC Subdivision | Area Unit |
| DISTRICT | OFFENSE\_TYPE\_ID | value | Date | Person/Organisation | Table 1 |
| REPORTING\_AREA | OFFENSE\_CATEGORY\_ID | year | Time | Table 1 | Location Type |
| SHOOTING | FIRST\_OCCURRENCE\_DATE | month | PdDistrict | Selected Period | Locn Type Division |
| OCCURRED\_ON\_DATE | LAST\_OCCURRENCE\_DATE |  | Resolution | Previous Period | Meshblock |
| YEAR | REPORTED\_DATE |  | Address | Age Group | Number of Records |
| MONTH | INCIDENT\_ADDRESS |  | X | % Variance | Occurrence Day Of Week |
| DAY\_OF\_WEEK | GEO\_X |  | Y | Ethnicity | Occurrence Hour Of Day |
| HOUR | GEO\_Y |  | Location | Ethnic Group | Territorial Authority |
| UCR\_PART | GEO\_LON |  | PdId | Months Ago | Victimisations |
| STREET | GEO\_LAT |  |  | Mop Division | Weapon |
| Lat | DISTRICT\_ID |  |  | Mop Group | Year Month |
| Long | PRECINCT\_ID |  |  | Mop Subdivision | Year Month (copy 2) |
| Location | NEIGHBORHOOD\_ID |  |  | Method of Proceeding | Month Year |
|  | IS\_CRIME |  |  | Number of Records |  |
|  | IS\_TRAFFIC |  |  | Police Area |  |
|  |  |  |  | Police District |  |
|  |  |  |  | Proceedings |  |
|  |  |  |  | Year Month |  |
|  |  |  |  | Ytd Month |  |

## Descriptive Analysis

The basic descriptive analysis can provide a big picture of the datasets and initiate the idea and action items to transform the data.

Table 6. Crime datasets and features' details.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Datasets | # of Records | # of Columns | # of  Crime Types | Has Date | Has Time | Has Location |
| Boston | 319,074 | 17 | 66 | Y | Y | Y |
| Denver | 466,841 | 19 | 15 | Y | Y | Y |
| London | 13,490,605 | 7 | 9 | year|month | N | Y |
| San Francisco | 150,501 | 13 | 39 | Y | Y | Y |
| New Zealand (1) (16 csv files) | 641,641 | 23 | 16 | year|month | N | Y |
| New Zealand (2) (nz\_victim\_timeplace.csv) | 1,194,764 | 17 | 6 | Y | Y | Y |

## Data Visualization

A simple data visualization such as plotting a bar graph to show the total number and percentages of the categories of each field is helpful to detect the pattern of data. For example, the following graphs of crime types show the imbalanced data, which is needed to be managed before training the models.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Figure 19. Visualize all variables in the dataset (New Zealand).

Another advantage of data visualization is to detect the outlier of data. For example, in the New Zealand crime dataset, only the Unlawful Entry crime has values in the Location Type column. If the Location Type is used as a feature variable for the predictive model, the model will not do predictions. It will just select the Unlawful Entry crime type as a result.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Figure 20. Visualize the outlier in the dataset (New Zealand).

## Baseline Feature and Target Variables

In order to generate the baseline scores from the predictive models, we have not performed the feature selection at this stage because the business needs of the crime prediction tool certainly require the time and location as the inputs.

Table 7. Define feature and target variables for the baseline.

|  |  |  |
| --- | --- | --- |
| Datasets | Feature Variables | Target Variable |
| Boston | MONTH, DAY\_OF\_WEEK, HOUR, DISTRICT, REPORTING\_AREA, STREET | OFFENSE\_CODE\_GROUP |
| Denver | MONTH, DAY\_OF\_WEEK, HOUR, DISTRICT\_ID, PRECINCT\_ID | OFFENSE\_CATEGORY\_ID |
| London | borough, month | major\_category |
| San Francisco | MONTH, DAY\_OF\_WEEK, HOUR, PdDistrict, Address | Category |
| New Zealand (1) (16 CSV files) | MONTH, Age Group 5Yr Band, Ethnicity, Police District | ANZSOC Division |
| New Zealand (2) (nz\_victim\_timeplace.csv) | MONTH, DAY\_OF\_WEEK, HOUR, Area Unit, Territorial Authority | ANZSOC Division |

## Baseline Scores for All Datasets

From the previous section, the variables of each dataset are defined and ready to use for the model training. The trained models have generated the baseline scores from the original datasets as shown in the following table.

Table 8. Baseline scores for all datasets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithms | Boston | Denver | London | San Francisco | New Zealand (1) | New Zealand (2) (nz\_victim\_timeplace.csv) |
| LogisticRegression | 0.12 | 0.26 | 0.29 | 0.27 | 0.16 | 0.57 |
| GaussianNB | 0.12 | 0.28 | 0.29 | 0.27 | 0.16 | 0.59 |
| DecisionTreeClassifier | 0.12 | 0.33 | 0.29 | 0.24 | 0.10 | 0.54 |
| XGBClassifier | 0.19 | 0.40 | - | 0.31 | 0.18 | 0.63 |
| RandomForestClassifier | 0.14 | 0.36 | 0.29 | 0.28 | 0.08 | 0.58 |

## Prototyping with New Zealand (2) Dataset

The baseline scores show that New Zealand (2) dataset could have the potential to be used for further development because the models can return the best scores comparing to the other datasets.

By considering the information from descriptive analysis, New Zealand (2) dataset has a high number of records (1,194,764) for 6 crime types. This relation between record count and target label is important and affect the accuracy. The difference between New Zealand (1) and New Zealand (2) dataset is that New Zealand (2) dataset has day and time values, which we can use them as feature variables. Besides, the date and time values can be used for feature engineering.

# Feature Selection

The selection of features allows for faster training of the machine learning algorithm, reduces model complexity, and makes it easy to interpret. Through applying specific feature selection technique enables highly correlated score-based feature to be taken for the next phase. We are exploring four feature selection techniques that are Extratree classifier, SelectKBest, Chi2, and Mutual Info Classifier.

The results of the feature selection technique are shown in the tables below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 9. Feature selection with ExtraTrees.   |  |  | | --- | --- | | ExtraTrees | score | | ANZSOC Group | 0.45 | | ANZSOC Subdivision | 0.37 | | Location Type | 0.12 | | Locn Type Division | 0.04 | | Occurrence Hour Of Day | 0.01 | | Occurrence Day Of Week | 0.01 | | Area Unit | 0.00 | | Meshblock | 0.00 | | Territorial Authority | 0.00 | | Weapon | 0.00 | | Victimisations | 0.00 | | Month Year | 0.00 | | Year Month | 0.00 | | Year Month (copy 2) | 0.00 | | Table 1 | 0.00 | | Number of Records | 0.00 | | | | Table 10. Feature selection with SelectKBest.   |  |  | | --- | --- | | SelectKBest | score | | Meshblock | 22672178.59 | | Location Type | 11775086.99 | | ANZSOC Group | 1145990.54 | | ANZSOC Subdivision | 776311.76 | | Area Unit | 427709.02 | | Occurrence Hour Of Day | 242718.00 | | Locn Type Division | 202103.12 | | Occurrence Day Of Week | 62543.47 | | Territorial Authority | 55941.36 | | Victimisations | 16218.72 | | Weapon | 2723.59 | | Year Month | 328.31 | | Year Month (copy 2) | 328.31 | | Month Year | 328.31 | | Table 1 |  | | Number of Records |  | |
| Table 11. Feature selection with Mutual Information.   |  |  | | --- | --- | | mutual\_info\_classif | score | | ANZSOC Group | 1.02 | | ANZSOC Subdivision | 1.02 | | Location Type | 0.62 | | Weapon | 0.34 | | Locn Type Division | 0.26 | | Meshblock | 0.22 | | Occurrence Hour Of Day | 0.10 | | Area Unit | 0.08 | | Occurrence Day Of Week | 0.06 | | Territorial Authority | 0.01 | | Victimisations | 0.01 | | Year Month | 0.00 | | Year Month (copy 2) | 0.00 | | Month Year | 0.00 | | Table 1 | 0.00 | | Number of Records | 0.00 | | Table 12. Feature selection with Chi Contingency.   |  |  |  | | --- | --- | --- | | chi2\_contingency | stat | Degrees of freedom | | ANZSOC Group | 5973805 | 80 | | ANZSOC Subdivision | 5973805 | 40 | | Weapon | 1361714 | 85 | | Location Type | 1194761 | 165 | | Meshblock | 615441 | 218960 | | Locn Type Division | 611735 | 20 | | Occurrence Hour Of Day | 236760 | 120 | | Area Unit | 195296 | 9685 | | Occurrence Day Of Week | 132295 | 35 | | Territorial Authority | 25648.7 | 325 | | Victimisations | 13665.2 | 90 | | Year Month | 2616.84 | 340 | | Year Month (copy 2) | 2616.84 | 340 | | Month Year | 2616.84 | 340 | | Table 1 | 0 | 0 | | Number of Records | 0 | 0 | | | |

# Detect Invalid Data

We identified the invalid data for all variables with the record counts and defined the actions to the variables. Some records can be simply removed from the dataset if it seems to have no impact to the dataset.

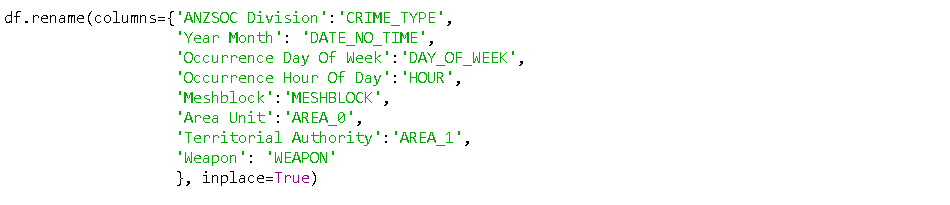
Table 13. Invalid data in New Zealand dataset.

|  |  |  |
| --- | --- | --- |
| Columns & Values | Record Counts | Actions |
|  | 4,709 | There are 4,709 records have ‘-29’ and ‘999999’ as the Area Unit’s values. It could be considered as a small proportion compared to 1 million records of the dataset. So, the records can be dropped. |
|  | 376,155 | Impute the values by the mode value of the day of week grouped by ANZSOC Division and Area Unit. |
|  | 547,297 | Impute the values by the mean value of hours grouped by the ANZSOC Division and Area Unit. |
|  | All | The dataset provides the first day of months as of date values. This effects the feature engineering technique, which calculates the crime ratio and mean value of a specific date. So, the day’s value of this field will be replaced by the day index of the day of the week column. |

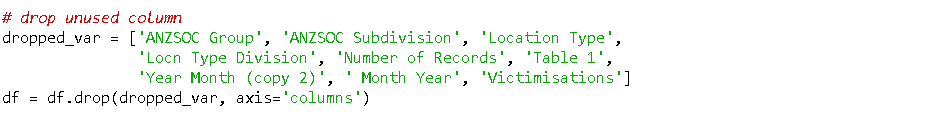
# Data Pre-Processing

We performed data cleaning and feature engineering to enrich the data aiming to improve the accuracy score of the predictive model.

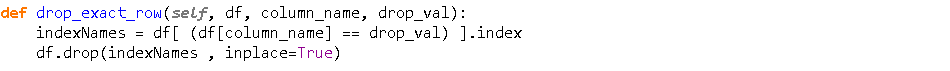
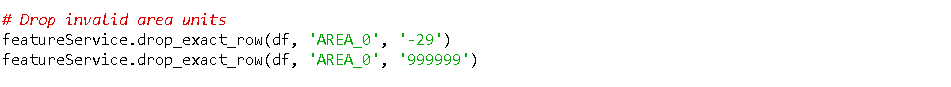
## Rename Columns

Easier to define column by renaming the column name. Here the column name of the ANZSOC Division will provide the details on various forms of crime. Therefore, modifying the ANZSOC Division as CRIME TYPE would make recognition simpler.

## Drop Unused Columns

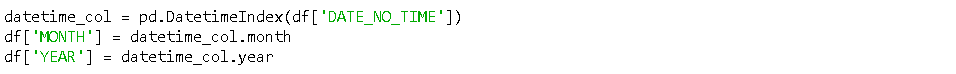
We dropped the columns which are not significant after performed the feature selection technique based on the score. Lower the score of features is more dependent.

## Drop Invalid Area Unit

As we have stated in the Detect Invalid Data section, all 4,709 records of the Area Unit that contain invalid values are removed.

## Extract Month and Year

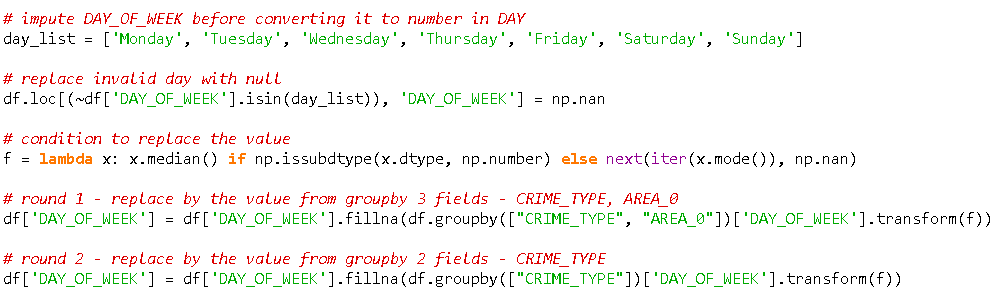
The original data has provided the date values, we extract the month and year values for the use of the other feature engineering.



## Impute Day of Week

The ‘UNKNOWN’ values are replaced by mode values of the day of the week. This stage requires 2 steps for the imputation.

1. In the first round, we replace ‘’UNKNOWN’ values by mode values in the same crime type and area unit (AREA\_0). However, there can be some null values left.
2. For the second round, we replace them with the mode values in the same crime type only.



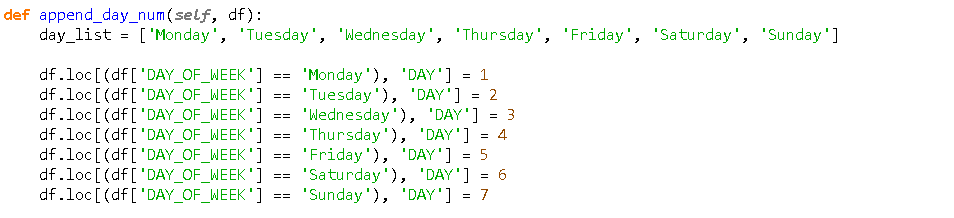
|  |  |
| --- | --- |
|  |  |

Figure 21. Transform the UNKNOWN data into DAY\_OF\_WEEK.

## Convert Day of Week to Day Index

After we imputed all ‘UNKNOWN’ values in this column in the previous step, this step will define Monday to be the first day of the week.





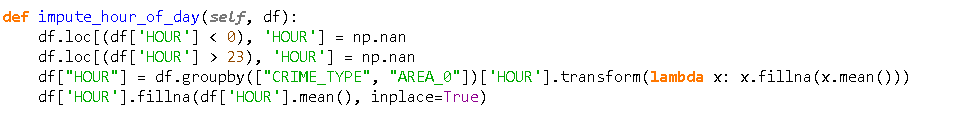
|  |  |
| --- | --- |
|  |  |

Figure 22. Transform the DAY\_OF\_WEEK into DAY.

## Impute Hour of Day

The invalid hours of day values are replaced by the mean values in the same area unit and crime type. And if there still be some records with the null values, they will be replaced by the mean values calculated from all records.





|  |  |
| --- | --- |
|  |  |

Figure 23. Transform the invalid hours into the valid hours.

## Generate Crime Factors

The crime factors were calculated by considering the total crime in specific times and locations. Since the original dataset has provided the crime type, day, month, area unit, and territorial authority, we were able to calculate the ratio, rank, and risk level of crimes in the area and period of interest.

The following steps show how to calculate the factors to derive the crime’s risk by the given information from the dataset.

1. Calculate the crime counts.
   1. MONTH\_AREA\_CRIME\_COUNT
   2. DAY\_AREA\_CRIME\_COUNT
   3. MONTH\_AREA\_CRIME\_TYPE\_COUNT
2. Derive the mean value from DAY\_AREA\_CRIME\_COUNT.
   1. 3\_DAY\_AREA\_MEAN\_CRIME
3. Calculate the crime ratios.
   1. DAY\_AREA\_CRIME\_RATIO
   2. MONTH\_AREA\_CRIME\_RATIO
   3. CRIME\_RATIO
4. Put the CRIME\_RATIO into a rank.
   1. CRIME\_RANK.
5. Divide the CRIME\_RANK into 3 groups, and then map the groups into the risk levels.
   1. RISK.

The following sub-sections elaborate on the correlation of crime type, time, and location of each variable.

### MONTH\_AREA\_CRIME\_COUNT

As a first stage to calculate the ratios of the crime in a specific month and territorial authority, the crime’s records are counted grouped by month and territorial authority (AREA\_1).

|  |  |  |
| --- | --- | --- |
|  |  | Territorial Authority (T) |
|  |  |  |
| Month (M) |  | Total Crime |
|  |
|  |
|  |
|  |

Figure 24. Total crime grouped by month and territorial authority.

Thus, we obtained the equation and implemented the code in Python.

*Total Crime(M,T) = Crime Count Grouped by Month and Territorial Authority*



### DAY\_AREA\_CRIME\_COUNT

In this stage, we were interested in crime events of area units (AREA\_0) and days, which are the subset of territorial authority (AREA\_1) and month respectively. The relations of 4 variables can be illustrated in the following picture.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Territorial Authority (T) | | | | | | |
|  |  |  | | | | | | |
| Month (M) | Area Unit 1 | Mon | Tue | Wed | Thu | Fri | Sat | Sun |
| Area Unit 2 | … | … | … | … | … | … | … |
| Area Unit 3 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |
| Area Unit n |  |  |  |  |  |  |  |

Figure 25. Total crime grouped by day and area unit.

Thus, we obtained the equation.

*Total Crime(M,T) = Sum of Everyday Crime Count of All Area Unit*

And then we implemented the code to count the crime events in a specific day, month, and area unit.



### 3\_DAY\_AREA\_CRIME\_MEAN

After we derived the DAY\_AREA\_CRIME\_COUNT, we can simply calculate the mean values of every 3 days in the level of the area unit.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Territorial Authority | | | | | | |
|  |  |  | | | | | | |
| Month | Area Unit 1 | Mon | Tue | Wed | Thu | Fri | Sat | Sun |
|  | |------- Mean 1 -------| | | |  |  |  |  |
|  |  | |------- Mean 2 -------| | | |  |  |  |
|  |  |  | |------- Mean 3 -------| | | |  |  |
|  |  |  |  | … | | |  |

Figure 26. 3 days mean of the crime.

And then we implemented the code. The mean value was performed with the round function to avoid floating numbers.



### MONTH\_AREA\_CRIME\_TYPE\_COUNT

We attempted to generate the crime ratio with a different view, rather than by using only the time and location. So, we counted the crime events separated by crime types.

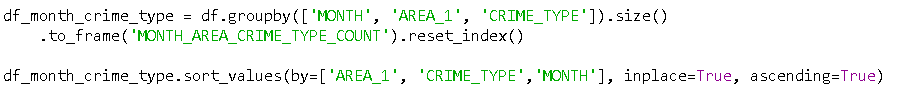
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Territorial Authority | | | | | |
|  |  |  | | | | | |
| Month |  | Crime Type 1 | Crime Type 2 | Crime Type 3 | Crime Type 4 | Crime Type 5 | Crime Type 6 |
|  |
|  |
|  |
|  |

Figure 27. Total crime grouped by crime type

Thus, we obtained the equation.

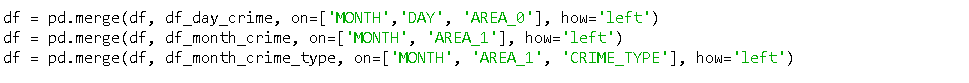
*Total Crime(M,T) = Sum of Crime Count by Crim Type*

And then we implemented the code in Python.



### Merge Crime Factors into the Dataframe

After we derived all 3 variables for the ratio calculation, we merged them into the main dataframe.



### DAY\_AREA\_CRIME\_RATIO

To calculate the ratio of a day crime in an area unit, we divided the DAY\_AREA\_CRIME\_COUNT by MONTH\_AREA\_CRIME\_COUNT.



### MONTH\_AREA\_CRIME\_RATIO

To calculate the ratio of a crime type in a territorial authority, we divided the MONTH\_AREA\_CRIME\_TYPE\_COUNT by MONTH\_AREA\_CRIME\_COUNT.



### CRIME\_RATIO

The final crime ratio was the result of DAY\_AREA\_CRIME\_RATIO multiplied by MONTH\_AREA\_CRIME\_RATIO.



### CRIME\_RANK

We put all crime ratios into the ranks in the area unit level because we anticipated displaying the risk of each area unit on geolocation.



### RISK

In the last stage of the calculation, we mapped the range of crime ranks into the risk levels.



As a result, the total crime at each risk level is almost balanced, and they were influenced by crime type, time, and location variables.

|  |  |
| --- | --- |
|  |  |

Figure 28. The total crime at each risk level.

### Unit Testing

Regarding the equations to calculate the total crime in a specific month and territorial authority.

*Total Crime(M,T) = Crime Count Grouped by Month and Territorial Authority*

*Total Crime(M,T) = Sum of Everyday Crime Count of All Area Unit*

*Total Crime(M,T) = Sum of Crime Count by Crim Type*

We rewrote the equations in terms of ratios as follows.

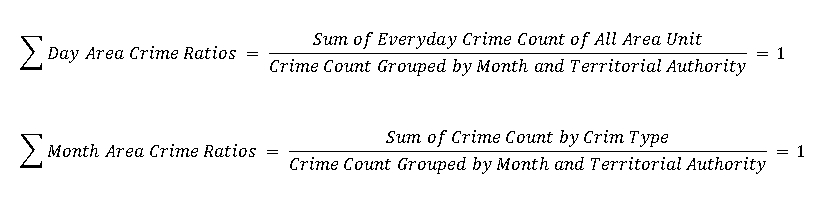
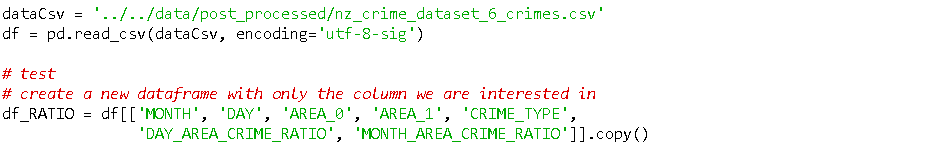


Figure 29. The equations of crime ratios.

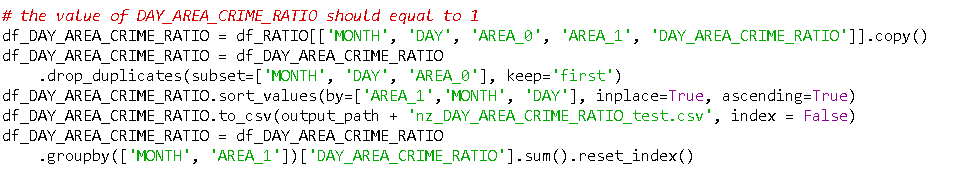
So that we can verify the correctness of the calculation by performing the sum of ratio values grouped by month and territorial authority, which should give the results equal to 1 for all groups.

1. Verify DAY\_AREA\_CRIME\_RATIO

At first, we read the dataset and created a new dataframe by copying only the columns we wanted.



Then, we verified the sum of DAY\_AREA\_CRIME\_RATIO by coding the unit test.



And the test result showed the sum values in the last column.

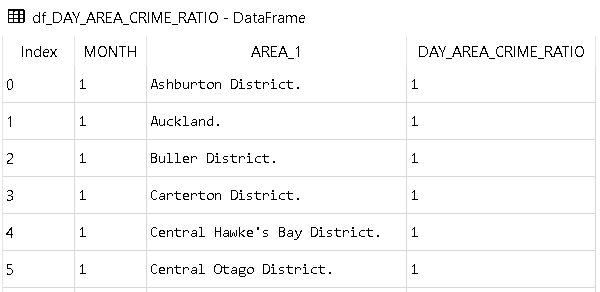
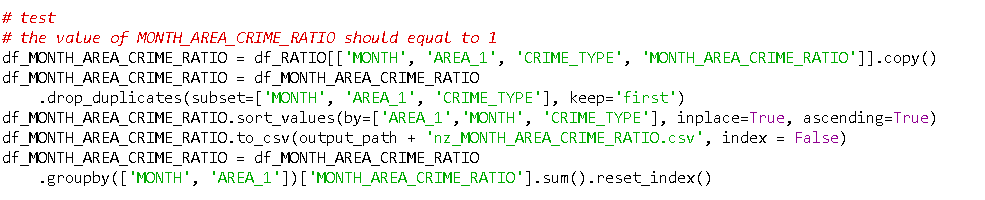


Figure 30. The total day crime ratio grouped by month and area\_1 should equal to 1.

1. Verify MONTH\_AREA\_CRIME\_RATIO

Here is the unit test to calculate the sum of MONTH\_AREA\_CRIME\_RATIO.



And the test result showed the sum values in the last column.

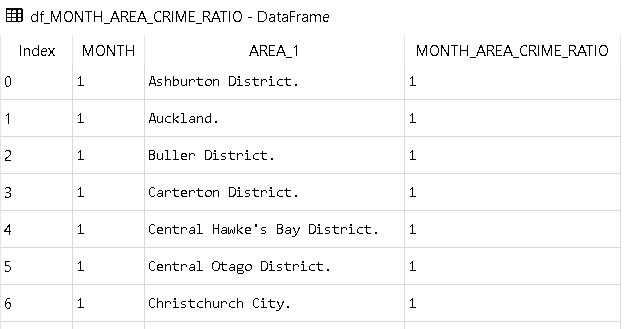
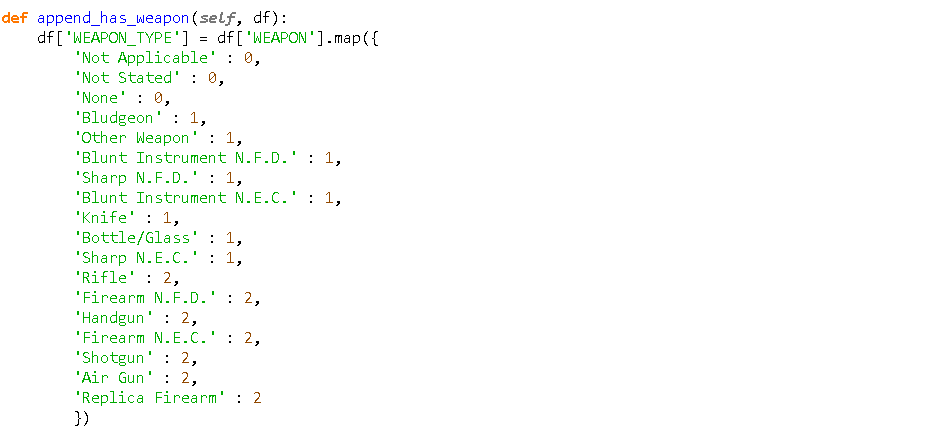


Figure 31. The total month crime ratio grouped by month and area\_1 should equal to 1.

## Group the weapon type.

We attempted to reduce the number of weapons in the dataset. However, using the weapon type as a feature variable did not improve the accuracy of the models.





|  |  |
| --- | --- |
|  |  |

Figure 32. Group the weapon types.

## Convert Month to Quarter

Since the percentage of the months in the dataset were similar, we reduced the number of months by mapping them into the quarters.





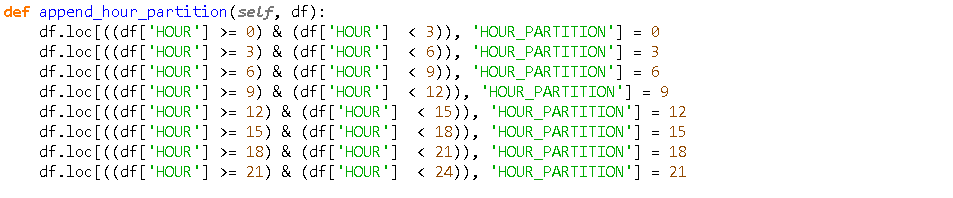
|  |  |
| --- | --- |
|  |  |

Figure 33. Group months into quarters.

## Convert Hour to Partition

We divided 24 hours of a day into 8 partitions. The trend of data before and after the transformation was still similar.



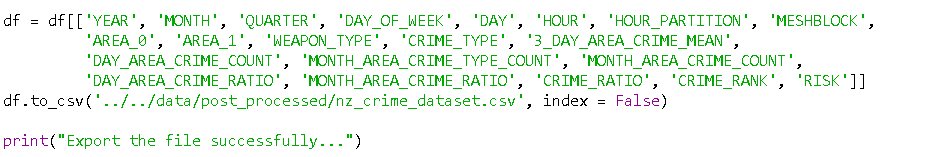


|  |  |
| --- | --- |
|  |  |

Figure 34. Group hours into hour partitions.

## Export a new CSV file

At the end of the process, a new dataset was generated. We exported the CSV file including many columns rather than choosing only the feature columns because we wanted to retain the information of the dataset and allow the predictive models to have an option to be trained with the original data.



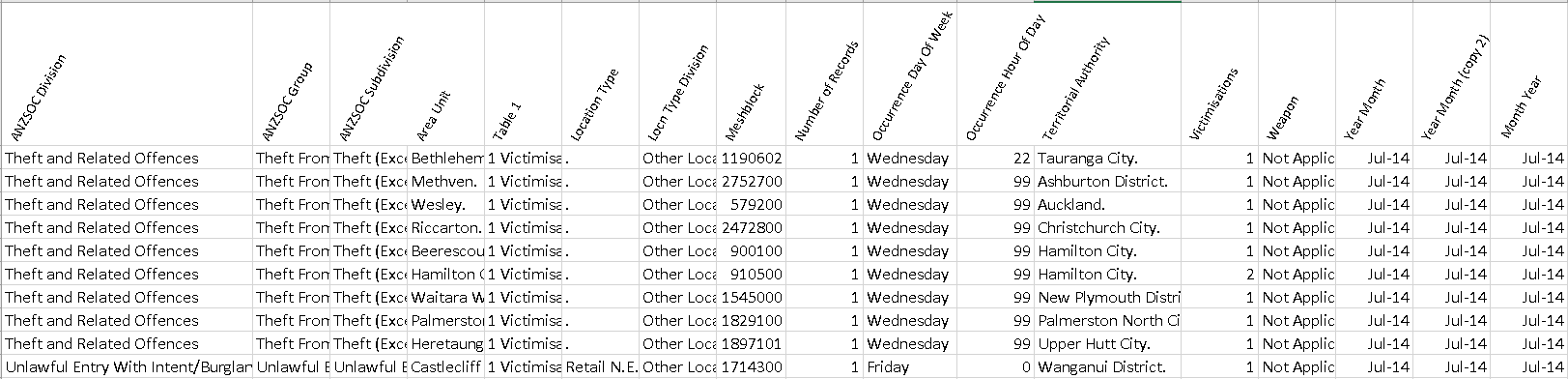


Figure 35. The variables in the original New Zealand crime dataset.

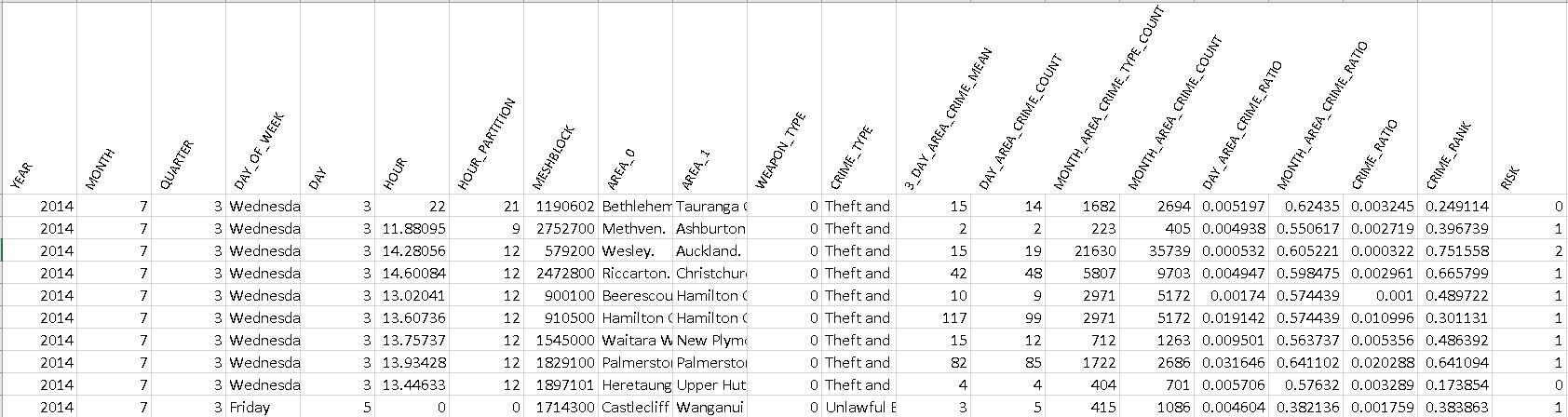
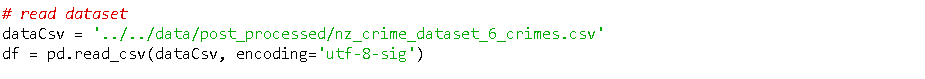


Figure 36. The variables after feature engineering in the New Zealand dataset.

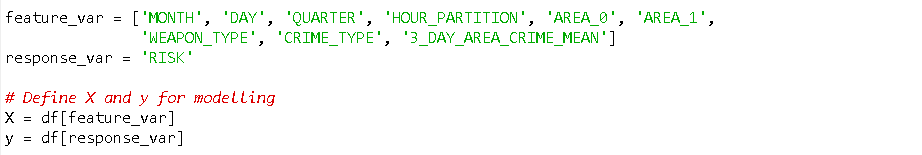
# Predictive Modelling

After the pre-processing and feature engineering process, the newly generated dataset was ready to use for predictive modelling. The following steps show how the predictive models with different algorithms were created starting from reading the crime dataset to the end of the process, which generated the classification report and confusion matrix.

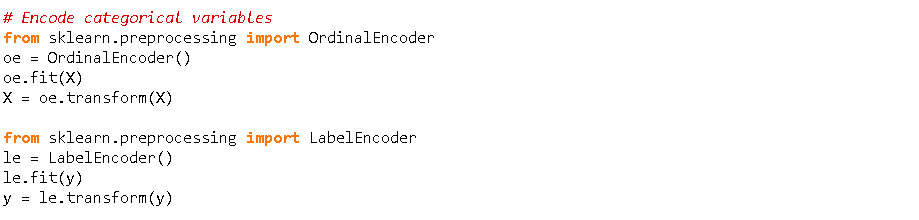
1. Read the crime dataset.



1. Define Feature and Target Variables



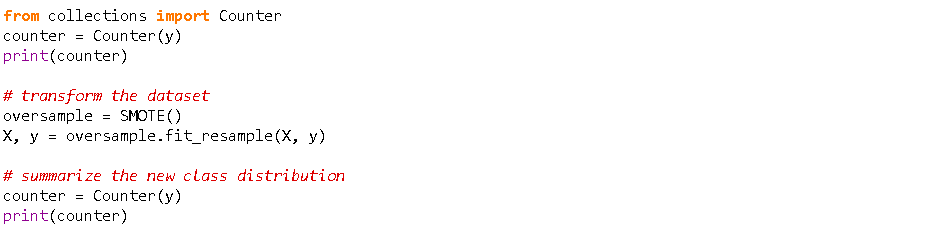
1. Encode Variables



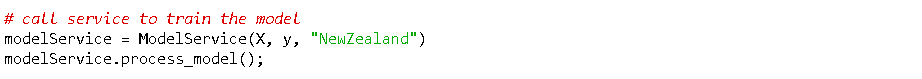
1. Apply PCA to reduce the number of feature variables and avoid overfitting.

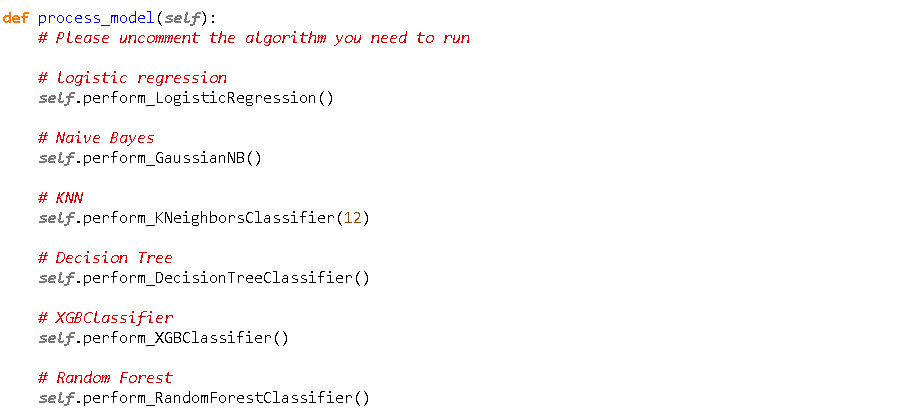


1. Apply SMOTE to manage imbalanced data.

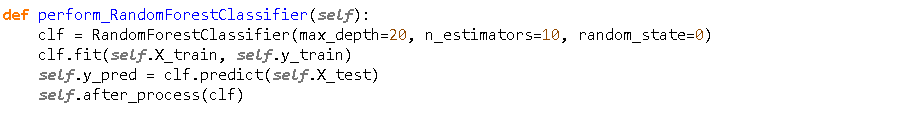


1. Train the models by calling functions in the ModelService class.

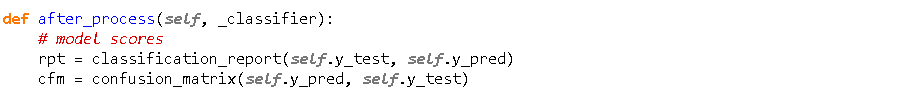




1. The function of every algorithm was implemented with the same code structure.



1. At the end of the process, the classification report and confusion matrix were generated to evaluate the models.



# Cross-Validation

After we had the acceptable accuracy scores from the trained models. It is important to verify that the scores were reliable and usable with the unseen data. In some situations, we could have a very high score from the trained model, but the score decreases when performing the cross-validation, which means to overfitting because the cross-validation reserves some part of the data as the unseen data to evaluate the model.

StratifiedKFold is a variation of k-fold which returns stratified folds: each set contains approximately the same percentage of samples of each target class as the complete set. We chose this type of cross-validation because it works well in both balanced and imbalanced data.

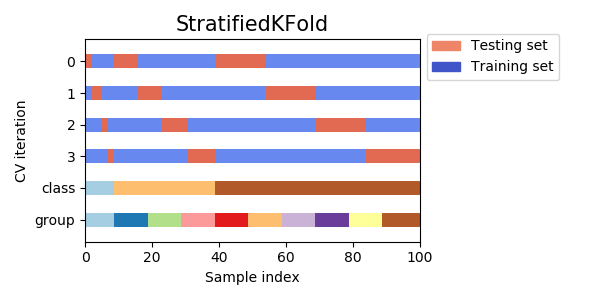
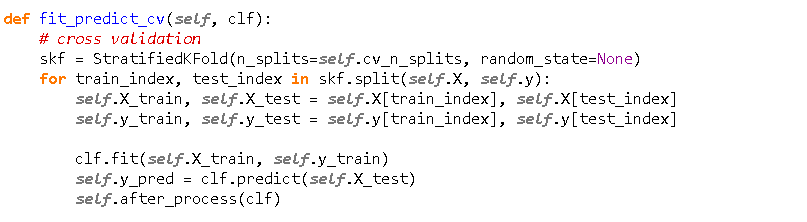


Figure 37. Cross-validation in sklearn.



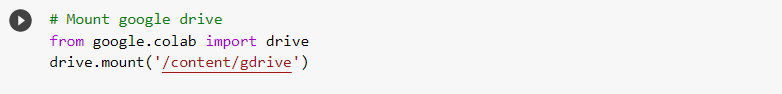
# Deep Learning: TensorFlow

In the later part of the sprint, we have explored Deep Learning: TensorFlow (developed by Google). Tensors are the standard way of representing data, and TensorFlow computes the data as a dataflow graph. We have applied 32 and 64 epoch into the new dataset that undergoes data transformation. For our model to learn more details and relationships within the data, we have applied five layers. While for visualization, TensorBoard enables us to visualize our model. It shows and compares the train and validation movement in every iteration. In terms of performance, we found TensorFlow train faster compared to PyTorch. Comparing the result of deep learning to machine learning, we got almost the same as what have in Decision Tree (0.76).

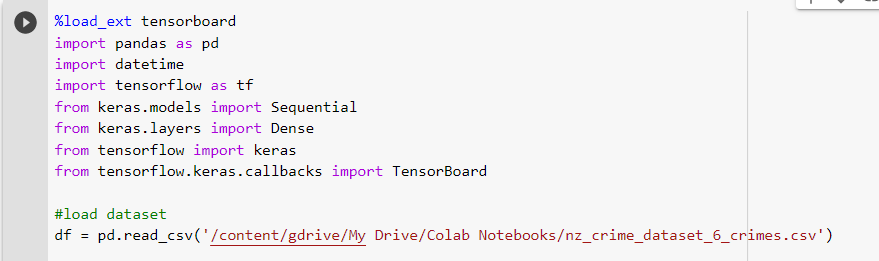
Table 14. Scores by TensorFlow.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | epoch=32 | | epoch=64 | |
|  | **accuracy** | **loss** | **accuracy** | **loss** |
| Deep Learning: TensorFlow  Batch size: 64 | 0.76 | 0.54 | 0.78 | 0.51 |

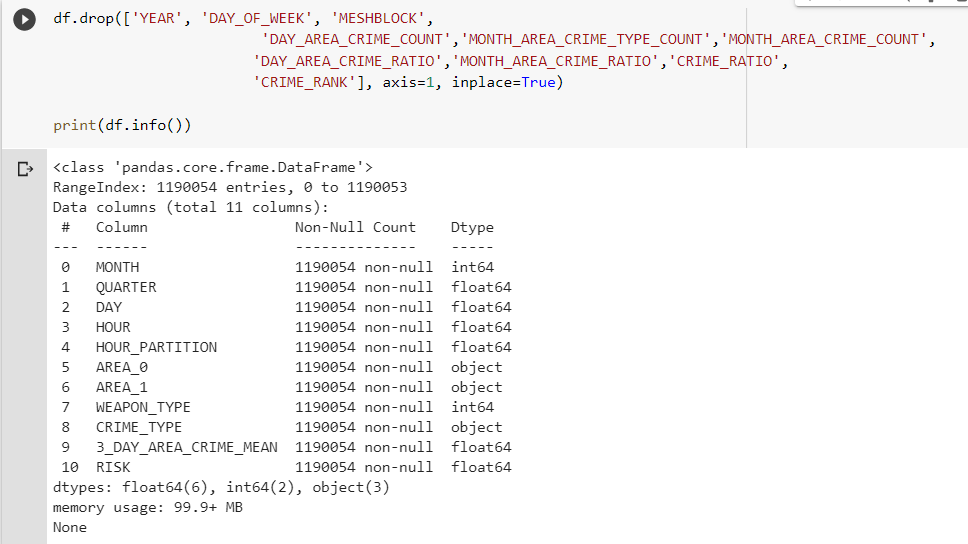
* + - 1. Mount the google drive where the dataset is stored.



* + - 1. Read the crime dataset.

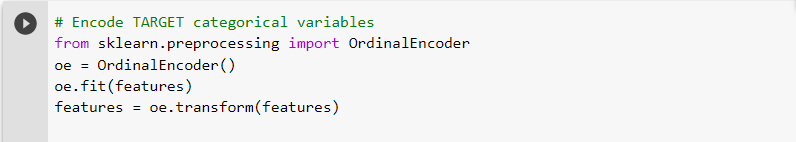


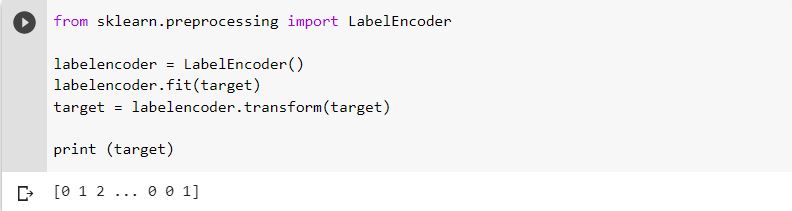
* + - 1. Define Feature and Target Variables.



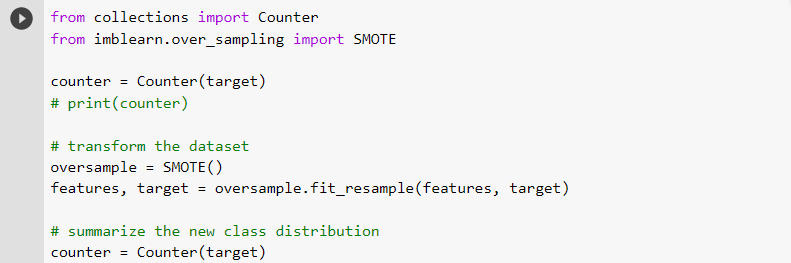


* + - 1. Encode variables.

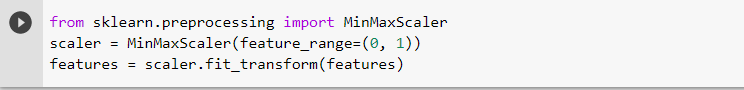




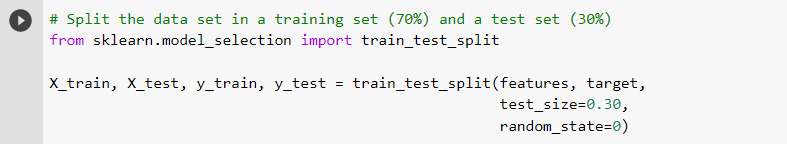
* + - 1. Apply SMOTE to manage imbalanced data.



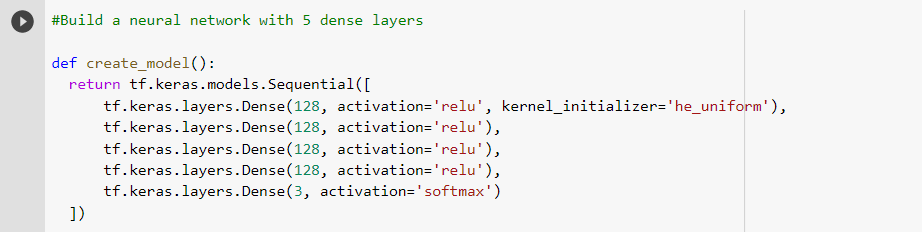
* + - 1. Transform features by scaling.



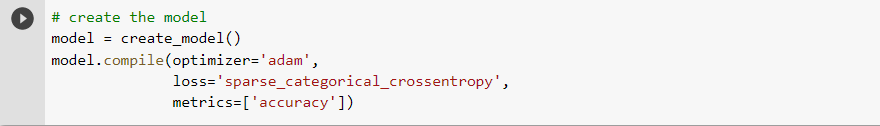
* + - 1. Split the dataset in a training set 70% and a test set 30%.



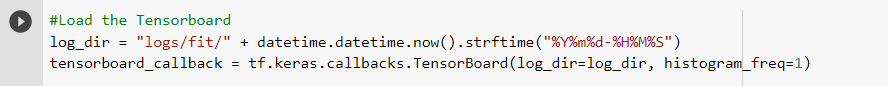
* + - 1. Build a neural network with 5 dense layers.



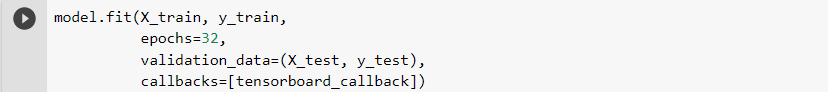
* + - 1. Create a predictive model.



* + - 1. Load the TensorBoard.



* + - 1. Fit the Model.



* + - 1. Visualize the movement of train and validation.

|  |  |
| --- | --- |
|  |  |

Figure 38. Visualize the model by TensorFlow Board.

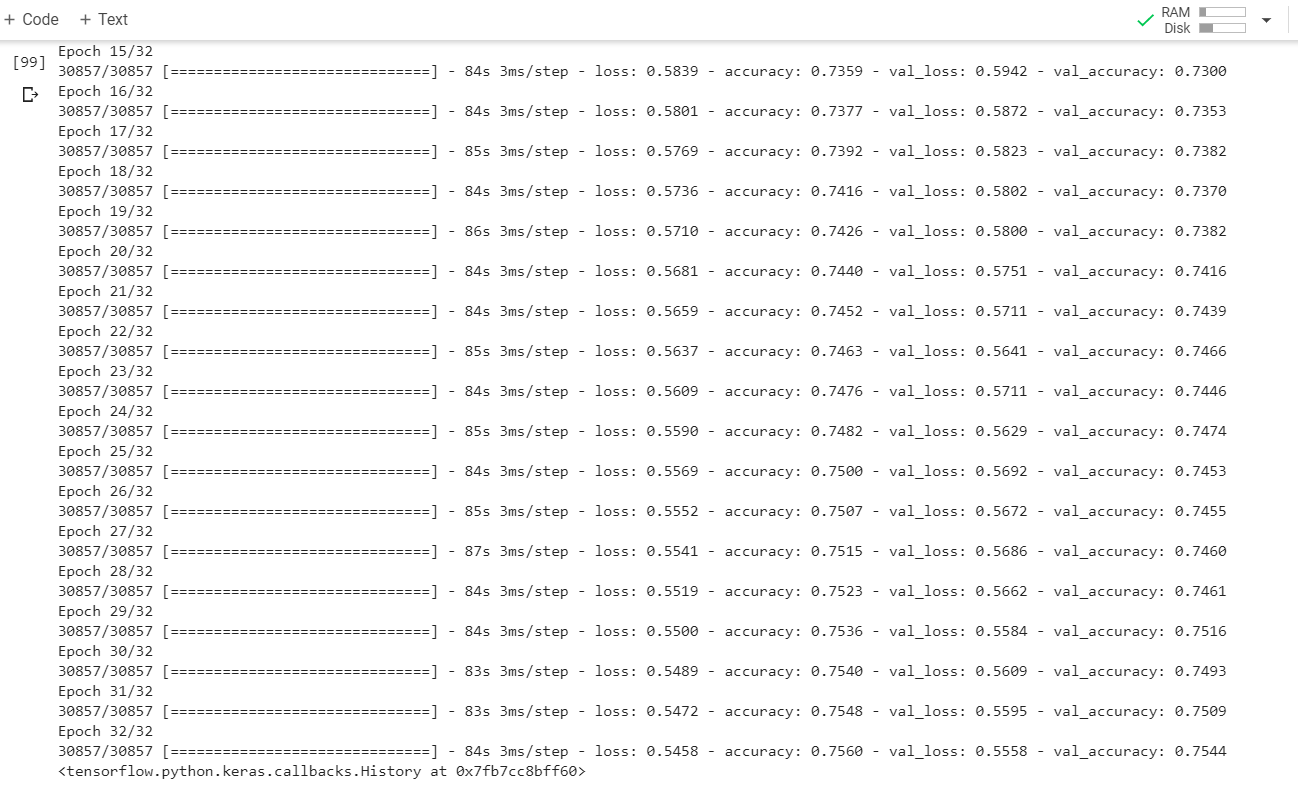


Figure 39. TensorFlow's scores by epochs.

# Results Evaluation

This section shows the score comparison, evaluation, and visualization. We captured the scores with 2 different target variables, which are the crime type and risk because we attempted to predict the crime type at the beginning of the project. Unfortunately, we were unable to improve the scores by predicting the crime type. The models were able to reach only 60% accuracy because of the complexity of the classification problem and the imbalanced data.

Further analysis showed that we could balance the target variable and still involved the crime type, time, and location variables for model training and prediction. Through the use of feature engineering techniques to calculate the crime ratios, rank, and risk, we have managed to balance the data and improve the score to around 80%-90% accuracy for the tree-based algorithm.

## Score Comparison

The predictive models were trained with various combinations of feature and target variables, which can be divided into 3 groups regarding the target variable.

1. The target variable is the top 3 crime types.

2. The target variable is the risk of the top 3 crime types.

3. The target variable is the risk of all crime types.

### Predict the Crime Type

First, we attempted to train the models to predict the crime type based on historical data, which we set the results as baseline scores as shown in the table below. We also performed feature engineering techniques to reduce the number of inputs by generating 3 new columns, which were QUARTER, HOUR\_PARITTION, and WEAPON\_TYPE. Unfortunately, we were unable to improve the score.

Table 15. Baseline scores of New Zealand crime dataset when the target variable is the crime type.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature vars. >>> | MONTH  DAY  HOUR  AREA\_0  AREA\_1  MESHBLOCK | | MONTH  DAY  HOUR  AREA\_0  AREA\_1  QUARTER  HOUR\_PARTITION  WEAPON\_TYPE | |
| **Target vars. >>>** | **CRIME\_TYPE** | | | |
|  | **Baseline** | | **Train=70/Test=30**  **Cycle 1** | |
| **algorithm** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** |
| LogisticRegression | 0.57 | 0.97 | 0.59 | 0.87 |
| GaussianNB | 0.59 | 0.98 | 0.59 | 4.41 |
| DecisionTree | 0.54 | 14.23 | 0.56 | 13.10 |
| XGBoost | 0.63 | 0.85 | 0.65 | 0.76 |
| RandomForest | 0.58 | 4.08 | 0.58 | 3.68 |

### Predict the Risk of Top 3 Crime Types

We were aware that the prediction by targeting the crime types may have two limitations. The first is the imbalanced data in classification problem causing the low accuracy in the results. The second is the correlation among feature variables and between the feature variables and the target variable. When there are many feature variables and most of them are categorical data, it will be difficult to identify the correlation for the trained models. These underline the difficulty of achieving an accurate prediction.

This implies that if we could build the correlation and balance the variables, it should probably improve the accuracy of the models. We hypothesize that we can generate the crime risk by using the time, location, and crime type as the factors to overcome the limitations. As a result, we can improve the accuracy of the tree-based models, which are Decision Tree, XG Boost, and Random Forest. The table below shows the scores of ML models when attempted to predict the risk of top-3 crime types including the scores when we performed the cross-validation technique, which both were similar and acceptable.

Table 16. Scores of New Zealand’s top 3 crime dataset when the target variable is the crime risk.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feature vars. >>> | MONTH DAY AREA\_0 AREA\_1 QUARTER HOUR\_PARTITION WEAPON\_TYPE CRIME\_TYPE | | MONTH DAY AREA\_0 AREA\_1 QUARTER HOUR\_PARTITION WEAPON\_TYPE CRIME\_TYPE 3\_DAY\_AREA\_CRIME\_MEAN | | | | | |
| **Target vars. >>>** | **RISK (Top 3 crime types)** | | | | | | | |
|  | **train=70 / test=30** | | **train=70 / test=30** | | **K-Fold CV. (k=5)** **avg. scores** | | **K-Fold CV. (k=10** **avg. scores** | |
| **algorithm** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** |
| LogisticRegression | 0.39 | 1.09 | 0.41 | 1.08 | 0.40 | 1.09 | 0.37 | 1.09 |
| GaussianNB | 0.43 | 1.07 | 0.38 | 1.11 | 0.37 | 1.12 | 0.35 | 1.12 |
| KNeighbors (k=12) | 0.74 | 1.06 | 0.78 | 1.09 | 0.78 | 1.09 | 0.79 | 1.09 |
| DecisionTree (max\_depth=20) | 0.76 | 1.67 | 0.76 | 2.01 | 0.76 | 1.90 | 0.77 | 1.79 |
| XGBoost (max\_depth=12) | 0.83 | 0.49 | 0.82 | 0.54 | 0.81 | 0.55 | 0.81 | 0.55 |
| RandomForest (max\_depth=20) | 0.81 | 0.54 | 0.83 | 0.51 | 0.83 | 0.52 | 0.83 | 0.52 |

### Predict the Risk of All Crime Types

We have managed to balance the data, build the correlation among variables, and improve the accuracy by predicting the risk of the top 3 crime types. This implies that we can achieve the high accuracy of the prediction if we can balance the data using crime risk calculation. So, we trained the models with 6 crime types. As a result, we still have high scores in tree-based algorithms.

Table 17. Scores of New Zealand crime dataset when the target variable is the crime risk (all crime types).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | train=70 / test=30 | | K-Fold CV. (k=5) avg. scores | | K-Fold CV. (k=10 avg. scores | |
| **algorithm** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** |
| LogisticRegression | 0.4 | 1.09 | 0.39 | 1.09 | 0.37 | 1.09 |
| GaussianNB | 0.37 | 1.12 | 0.36 | 1.12 | 0.34 | 1.12 |
| KNeighbors (k=12) | 0.77 | 1.03 | 0.78 | 1.04 | 0.79 | 1.04 |
| DecisionTree (max\_depth=20) | 0.76 | 1.81 | 0.75 | 1.81 | 0.75 | 1.7 |
| XGBoost (max\_depth=12) | 0.84 | 0.49 | 0.84 | 0.49 | 0.83 | 0.5 |
| RandomForest (max\_depth=20) | 0.84 | 0.51 | 0.83 | 0.52 | 0.83 | 0.52 |

The previous results from Decision Tree, XG Boost, and Random Forest were generated by specifying the max\_depth parameter, which limits the number of maximum depths of the tree from the root node to a leaf. Thus, the following table shows the results of tree-based models when the maximum depths were not specified. We reached about 90% accuracy scores in Decision Tree and Random Forest. But the score decreased in the XG Boost model, which we got 0.84 accuracy score when its maximum depths were 12 nodes.

Table 18. Scores of the New Zealand crime dataset when the target variable is the crime risk (all crime types without maz\_depth).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | train=70 / test=30 | | K-Fold CV. (k=5) avg. scores | | K-Fold CV. (k=10 avg. scores | |
| **algorithm** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** | **accuracy** | **log\_loss** |
| DecisionTree | 0.88 | 4.14 | 0.88 | 4.12 | 0.89 | 3.92 |
| XGBoost | 0.64 | 0.81 | 0.63 | 0.82 | 0.62 | 0.82 |
| RandomForest | 0.90 | 0.51 | 0.90 | 0.5 | 0.91 | 0.47 |

## Results Visualisation

An alternative but crucial way to evaluate the predictive models is by using plots to visualize the result because it is easier to understand the overall trend of the result. Usually, we can suspect the problems and root causes by investigating the plots. Another benefit of using the plots is to quickly describe the result to people.

This section shows 3 different plots generated by ML models when the models attempted to predict the crime risk:

1. Confusion Matrix.

2. Receiver Operating Characteristic Curve (ROC)

3. Precision-Recall Curve (PR)

### Confusion Matrix

The following confusion matrixes show the actual and predicted results from ML models. Logistic Regression shows the highest number of the prediction in the risk level 2 and the rest of the prediction is sparse. A similar result has shown in the Gaussian Naive Bayes but the highest number happens in the risk level 1.

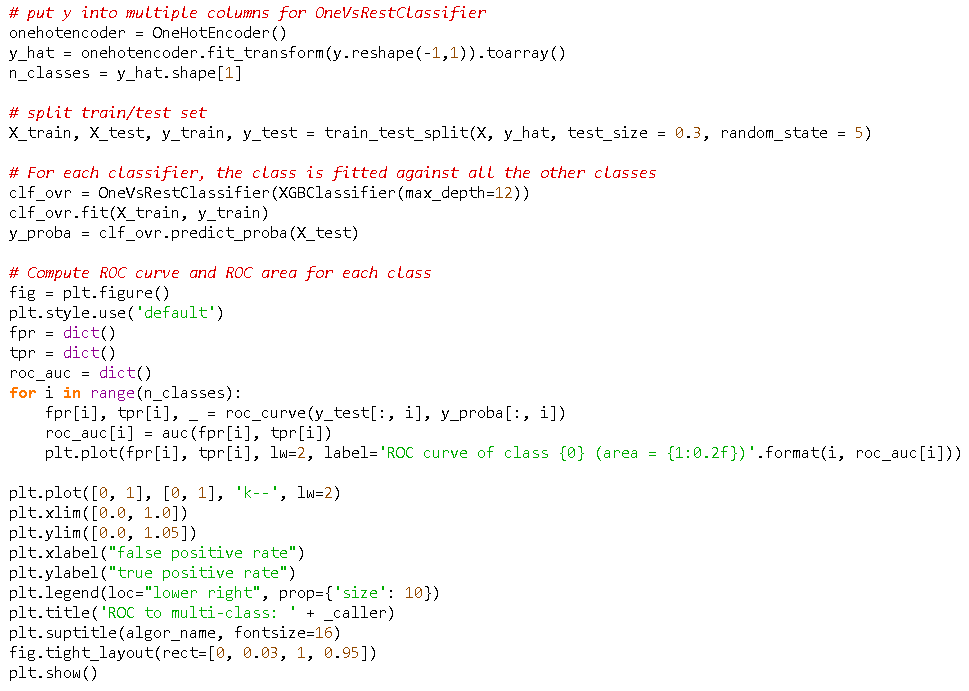
For the other models such as K-Nearest Neighbors, Decision Tree, Random Forest, and XG Boost, these models have the high accuracy scores and their confusion matrixes also show the same trends of the results, which most of the prediction are accurate.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Figure 40. Confusion matrixes of all models.

### Receiver Operating Characteristic Curve (ROC)

The perfect performance of the ROC curve is when the area under the ROC curve (AUC) equals to 1. In the plots below, the dashed line in the plots mean to 50% probability that the prediction result will be 0 or 1. We used OneVsRestClassifier technique to plot the curve of each risk level by comparing the probability of the target class to the other 2 classes. As a result, we drew 3 lines with an additional dash line for each model.

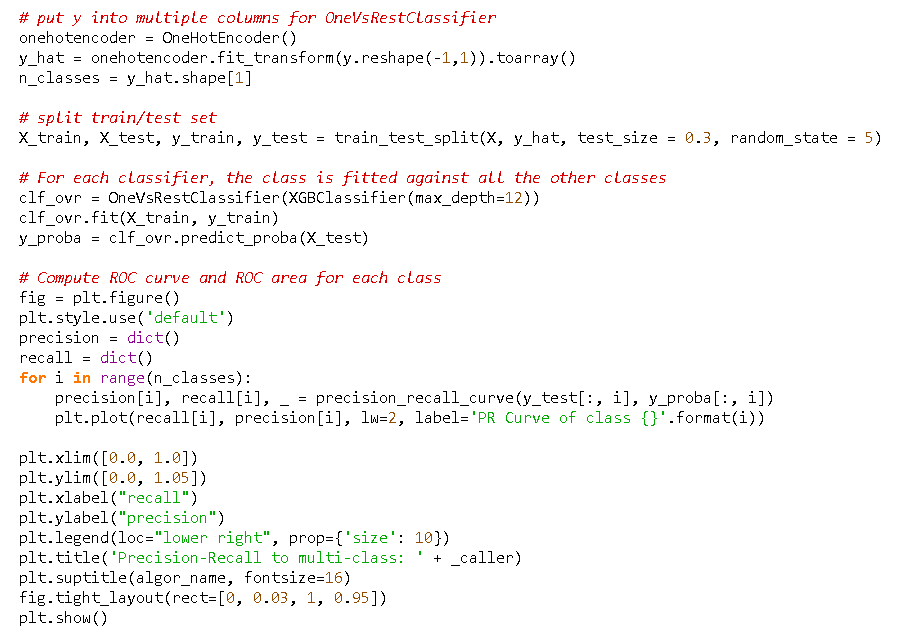


|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Figure 41. ROC curves of all models.

### Precision-Recall Curve (PR)

The PR curve is suggested when the classes are imbalanced. The ideal curve is similar to the ROC curve, in which the area under the curve should equal to 1. Thus, it represents both high recall and high precision, where high precision relates to a low false-positive rate, and high recall relates to a low false-negative rate. We also used OneVsRestClassifier technique to draw the curves.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Figure 42. PR Curves of all models.

## Quantitative Evaluation

At the beginning of predictive modelling, all models returned similar scores at about 55% - 60% accuracy, which we declared this score set as a baseline. Then we performed feature engineering techniques and generated the crime risk as a target variable for the prediction by aiming to reach high accuracy.

Our findings show that the tree-based models such as Decision Tree, XG Boost, and Random Forest can predict the crime risk with 80% - 90% accuracy scores. This is because the mechanism of tree-based algorithms is designed to solve the problem by creating the rules and split the nodes based on the feature variables, which will show a great performance in the large dataset.

The K-Nearest Neightbors model also achieved a fair accuracy at 76% when the number of nearest neightbors equal to 12. The outcome of this algorithm is a class membership when the members of that class have a set of common features. It can be one of the simple and powerful algorithms if we perform the parameter tuning in further development.

Both Logistic Regression and Gaussian Naive Bayes returned the poor results in this classification problem because the target classes cannot be divided into clear patterns. So, these 2 algorithms are not suitable to use for the prediction.

Deep Learning by using TensorFlow with 64 epochs performed a good prediction with about 80% accuracy score. We believe that the score can be increased with some further research and development. However, it would be less worth to build a deep learning model when we can have the same scores in machine learning models. Because training the model in deep learning is more expensive than machine learning in terms of computer resources and time spent in the training process.

# Application User Interface

The tool will predict the crimes by using a set of criteria from the web application. Users can select a specific ML algorithm for the prediction. The prediction results are the risk of crimes in the area units of the selected territorial authority. The users can click the pinpoint to see more details of the crime in each area unit.

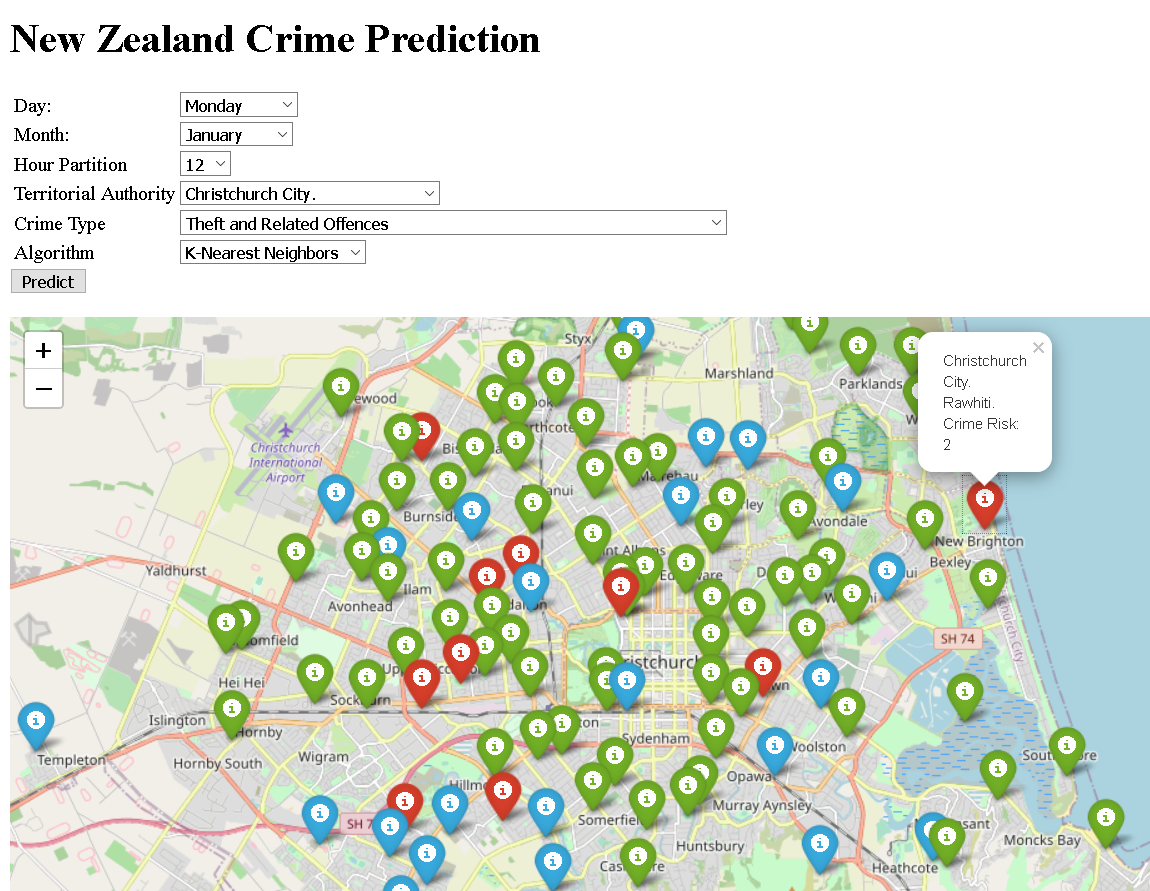


Figure 43. The prototype of an application user interface for the crime prediction in New Zealand.

## Using Microsoft Power BI

Microsoft Power BI is a tool for data analytics and visualization. It also allows us to run the trained models. We used this tool to explore and learn its functionality to run our machine learning model.

We used the power BI tool to predict the probability of crime by selecting the parameters given. By default, the map redirects the risk of various crimes to the Auckland area. This can also adjust a different position depending on the selection of the user. The test purpose is to give input as random data in a CSV file to run a machine learning model.

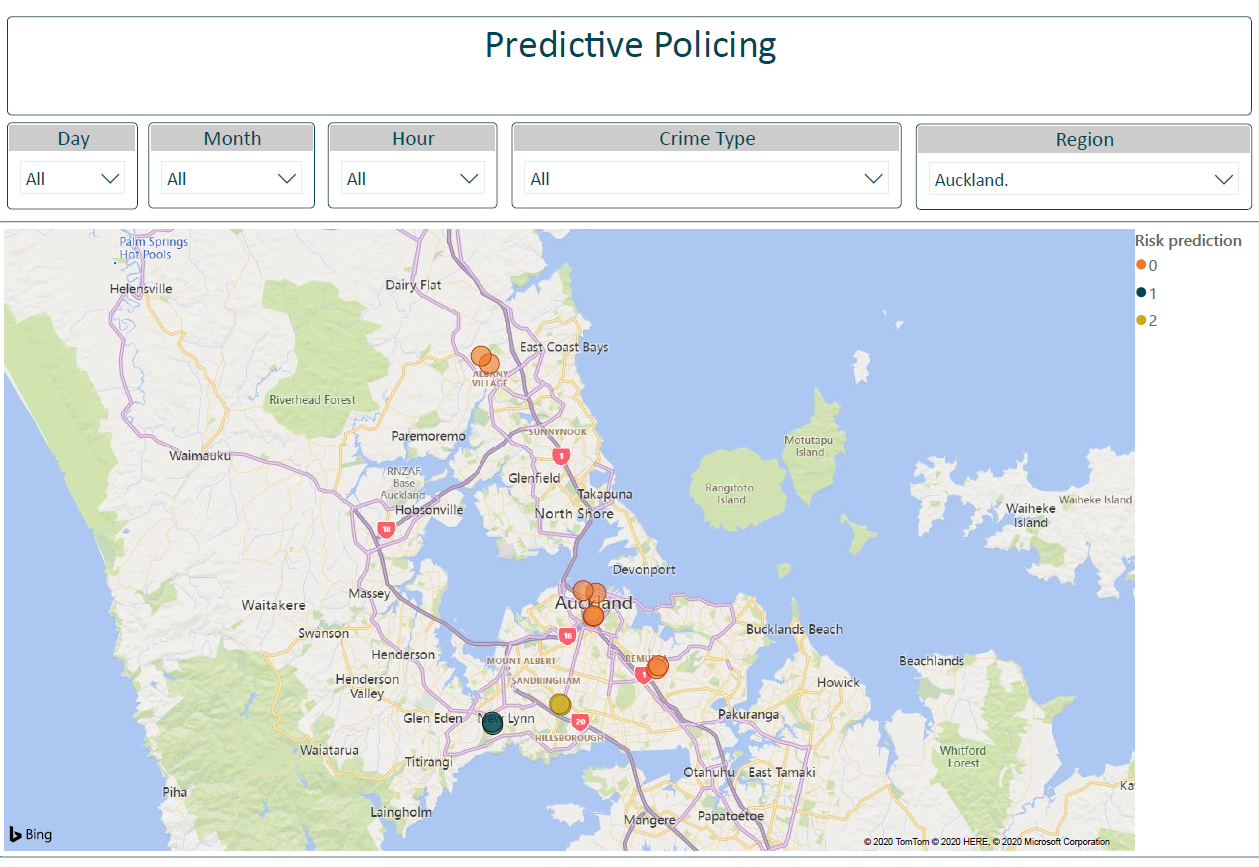


Figure 44. The predictive models’ integration with Power BI.

The above picture shows the various bubbles of red, green, and yellow (0, 1 and 2) reflecting the risk value of three different levels of lower, middle and higher risk of crime in the Auckland region.

# Conclusion

The original goal is to predict the type of crime. However, we have failed to predict the crime type and then we have come up with a new approach by predicting the risk of crime. By using different machine learning algorithms Logistic Regression, Gaussian Naive Bayes, Decision Tree, XG Boost, and Random Forest for prediction. Out of this tree-based (Decision Tree, XG Boost and Random Forest) algorithms are giving satisfying accuracy results 0.88, 0.64, and 0.9 respectively. Therefore, the Random Forest is 90 percent sure to predict the likelihood of crime, its best for our application.

We used the cross-validation, confusion matrix, and ROC curve to test the model results. The deep - learning concept of tensor-flow models has provided a result close to the model of machine training. By applying various techniques and imputing features to predict crime risk in each area by specifying the time and day of the week. The models would depend 80-90% on predicting the probability of crime risk. By developing a web application, we illustrate the use of models. The models could potentially integrate with commercial product such as Microsoft Power BI platform.

# Future Work

Further studies based on this project could focus on both improving the predictive models and integrating the models with the existing systems. For example:

* Train the separate models that are more suitable for predicting the crime risk for specific crime types, time, and location.
* Build a web form to collect the input from users and stores the data into a CSV file or the database. Then load the data as inputs by the commercial product such as Power BI to predict and visualize the result on geolocation.

# Discussion

We gained valuable experiences through the works in this project, which will be discussed.

## Basic Data Science and Coding Skills

We have learned the basics of data science and machine learning from the class. It is useful to have a good understanding of the concepts and programming languages such as Python and Object-Oriented programming because most of the tasks are research and reviewing someone’s code on the Internet. We would not be able to understand how the ML code works if we lack the basic skills. Math is extremely important in feature engineering and data visualization. By understanding how the data behaves in math will lead to a creative way to solve the problem.

## Research & Believe in the Feasible Solution

The data science project is completely different from the software projects. In software development, we prepare a very clear requirement and technical solution before the implementation. Otherwise, it will be the change requests when we need to modify the software.

In the data science project, we would say that there is no one time task. All tasks need to be researched, implemented, reviewed, and improved along the development process until we can have an acceptable result, in our case, the predictive models with acceptable accuracy scores. Then the models could be reliable only for a period and will be needed to improve again when we have more new data.

## Plot Graphs

Graphs can show some hidden problems behind the accuracy scores. For example, the following ROC Curves were generated to show the probability of classes predicted by Logistic Regression. The problem was that the area of class 5 was equal to 1, which gave us the concern that there could be some conditions or feature variables made an overfitting model only for class 5. Then, we reviewed the dataset and found that the field ‘Location Type’ exists only in class 5. To resolve this problem, we needed to remove the ‘Location Type’ from the feature variables.

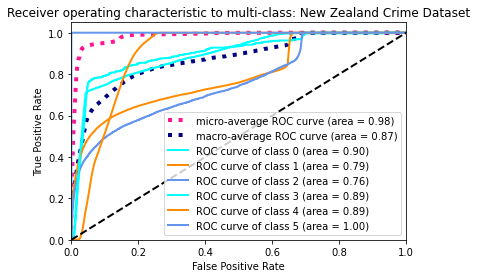


Figure 45. ROC curves reveal the overfitting.

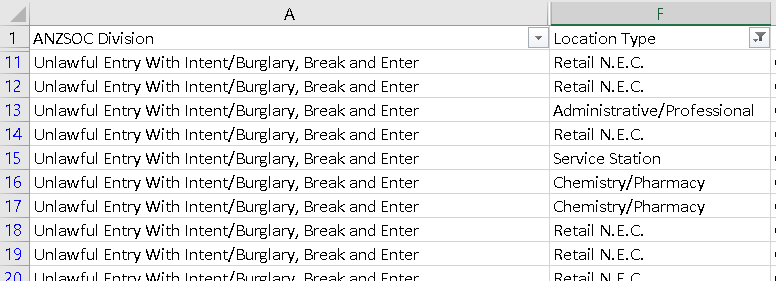


Figure 46. The location type found only in the Unlawful Entry crime.

# Reference List

Almanie T, Mirza R, Lor E. Crime Prediction Based on Crime Types and Using Spatial and Temporal Criminal Hotspots. International Journal of Data Mining & Knowledge Management Process. 2015;5(4):01–19.

Almaw A, Kadam K. Crime Data Analysis and Prediction Using Ensemble Learning. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). 2018;

Deadman, D. (2000). Forecasting residential burglary. Unpublished paper.

Horrocks, James, and Andrea Kutinova Menclova. “The Effects of Weather on Crime.” New Zealand Economic Papers, vol. 45, no. 3, 2011, pp. 231–254., doi:10.1080/00779954.2011.572544.

Kang H-W, Kang H-B. Prediction of crime occurrence from multi-modal data using deep learning. Plos One. 2017;12(4).

Martegiani, Gabriele. “Crime Prediction Using Data Analytics: the Case of the City of Boston.” *Academia.edu*, www.academia.edu/25277353/Crime\_Prediction\_Using\_Data\_Analytics\_the\_Case\_of\_the\_City\_of\_Boston.

Pradhan I. Exploratory Data Analysis And Crime Prediction In San Francisco. 2018;

Rumi SK, Deng K, Salim FD. Crime event prediction with dynamic features. EPJ Data Science. 2018;7(1).

Yuki, J. Q., Sakib, M. M. Q., Zamal, Z., Habibullah, K. M., & Das, A. K. (2019). Predicting Crime Using Time and Location Data. Proceedings of the 2019 7th International Conference on Computer and Communications Management. doi: 10.1145/3348445.3348483

# Appendices

## Jira Board & Trello

Jira Board: <https://opaiccrimeprediction.atlassian.net/browse/PP>

Trello: <https://trello.com/b/HNwI6tnI/sprint-retrospectives>

## Shared Documents

<https://otagopoly.sharepoint.com/sites/PredictivePolicing/Shared%20Documents/Forms/AllItems.aspx>

## Source Code

Wisanu: <https://bitbucket.org/wisanuboonrat/predictive_policing/src/master/>

Love: <https://bitbucket.org/acnaviza/predictive_policing_nz/src/master/>

Vimita: <https://bitbucket.org/vimi_vaidya/predictive_policing-vimi/src/master/>

## Web Resources & Tools

<https://www.atlassian.com/software/jira/scrum-boards>

<https://trello.com/>

<https://bitbucket.org/product/>

<https://colab.research.google.com/>

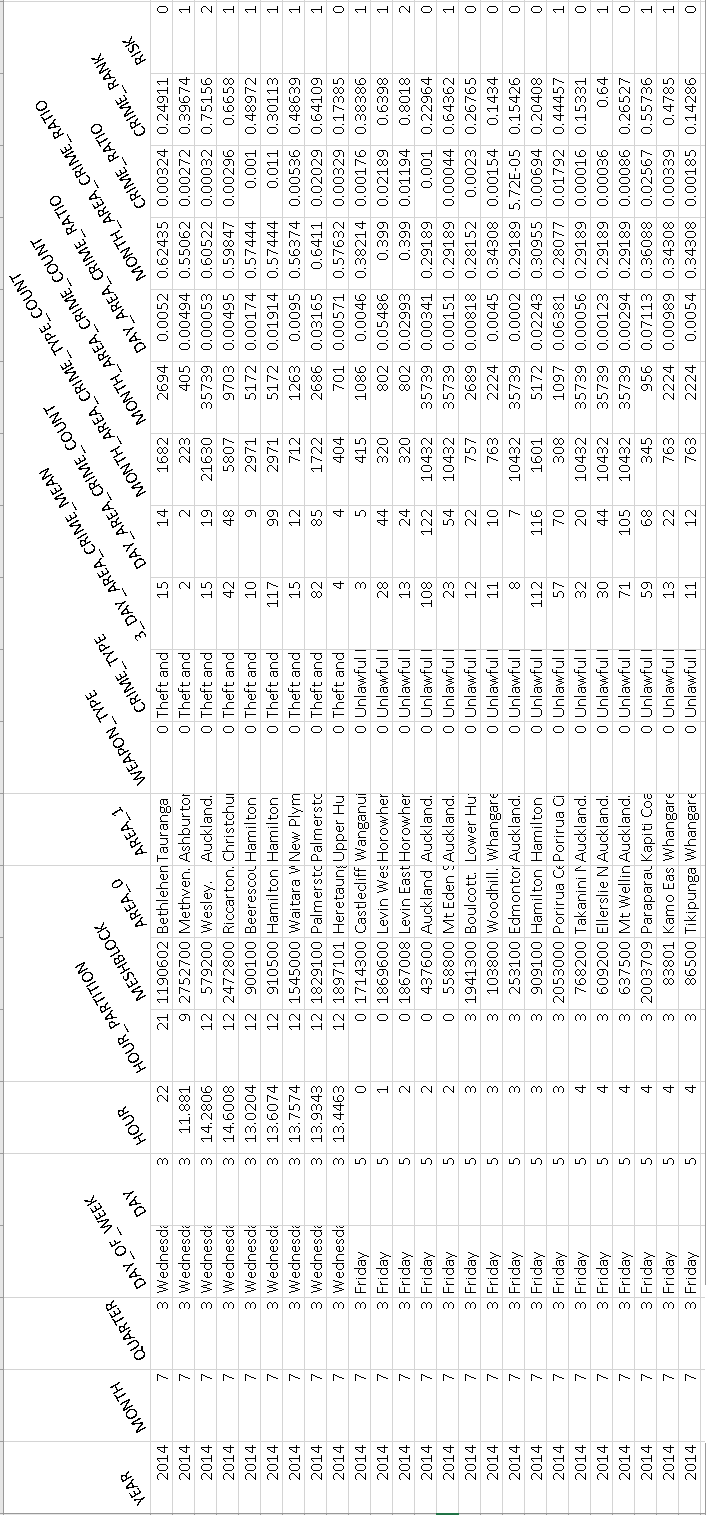
<https://github.com/>

<https://www.anaconda.com/>

<https://docs.anaconda.com/anaconda/>

<https://docs.conda.io/en/latest/miniconda.html>

## New Zealand Dataset (Final version)



1. <https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/#:~:text=Top%20reasons%20to%20use%20feature,the%20right%20subset%20is%20chosen.> [↑](#footnote-ref-2)
2. <https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/> [↑](#footnote-ref-3)