# CASA0006 Assessment: Body Text

Student Number: 22186878

### Background

This notebook presents the development and evaluation of a machine learning model applied to a New Zealand traffic crash dataset, to predict crash injury severity and determine the most influential contributing factors.

### Literature Review

Vehicle crash injuries present a major public health and economic burden on a global scale. In New Zealand, the government has implemented an explicit ‘Road to Zero’ strategy which aims to develop a road system where nobody is killed or seriously injured (New Zealand Government, 2019). In response to this complex challenge there is an extensive body of research, developed over decades, that has applied statistical and machine learning models to determine contributing factors in crash likelihood and improve prediction of crash severity (Savolainen *et al.*, 2011; Santos, Dias and Amado, 2022). Understanding the association of specific factors with crash outcomes allows for the design of evidence-based countermeasures, whether in the vehicle or in the built environment, whilst predictive capacity has potential implications for directing emergency response, and in the decision making of autonomous vehicles.

In recent years, there has been a broad spectrum of ML methods that have rapidly become the most popular choice for crash severity analysis due to their freedom from prior assumptions, robustness to outliers, and high performance in big data scenarios (Ziakopoulos and Yannis, 2020). Established methods include regression and decision trees (Jeong *et al.*, 2018), support vector machines (Effati, Thill and Shabani, 2015), neural networks (Delen *et al.*, 2017), and ensemble approaches such as Random Forest and Gradient Boosting (Iranitalab and Khattak, 2017; Wang and Kim, 2019). Of these various approaches, a systematic literature review by Santos et al. (2022) found that Random Forest (RF) models proved the highest performing algorithm in the most cases, achieving best performance, comparative to alternatives, in 70% of the studies it was applied.

Irrespective of the approach used, there are certain characteristics of vehicle crash data that need to be accounted for in designing a model. Firstly, these datasets predominantly collect an ordinal categorical outcome – such as crash severity categorised into fatal, serious, minor, or non-injury. These classes are generally heavily imbalanced, with a very high proportion of observations in the low-severity class, and a low proportion classed as severe/fatal. If this imbalance is not addressed, the model may produce overtly promising results with a high overall accuracy, but have very poor predictive performance for the high-severity classes, which are inherently the observations of greatest interest in crash analysis (Jeong *et al.*, 2018). In addition, crash events will have spatial and temporal dependence, where crashes occurring close in time and space are likely to share effects not otherwise captured in the data, which may introduce bias in model estimates (Savolainen *et al.*, 2011). Therefore, the aim of this study is to develop a reproducible python workflow that evaluates the effectiveness of ensemble ML models in a New Zealand context and quantifies the impact of class imbalance on performance.

### Research Question

1. How effectively can road and environmental factors predict vehicle crash severity?
2. Of these factors, which have the greatest influence on the prediction performance?

### Methodology

Data for this analysis are drawn from the New Zealand Transport Agency’s Crash Analysis System (CAS), which contains a record of all police-reported traffic crashes on New Zealand roadways from 1st January 2000 (Waka Kotahi NZ Transport Agency, 2023). There are over 70 features captured in the dataset, which can be broken down into road factors (such as number of lanes and speed limit), environmental factors (such as location, weather, and conditions), and crash outcomes (such as objects hit, vehicles involved, and injury severity).

Two ensemble classification methods, Random Forest and Gradient Boosting, are fitted to a training subset of the data and performance evaluated against a separate testing set. Numeric features are scaled, and one-hot encoding used to prepare categorical features for analysis. Default models are compared with balanced estimators from the python and `before [Insert sampling method] resampling method is applied to account for the class imbalance.

### Results

#### Read data

The CSV file is read in as a pandas data frame, and columns changed to the appropriate data types.

#### Clean missing data

The table below shows the default data type of imported columns, and a count of non-null values for each column. There are 807,933 observations (rows) in the dataset, and 72 features (columns).

Of these, 7 have a very high proportion of missing data (>90% of observations) - they are `advisorySpeed`, `crashRoadSideRoad`, `holiday`, `intersection`, `pedestrian`, `temporarySpeedLimit`, and `weatherB`. Through background reading of the [metadata](https://opendata-nzta.opendata.arcgis.com/pages/cas-data-field-descriptions), and investigation of the data, it can be seen that these columns contain a mix of true missing data and incorrectly coded real data. For example, `pedestrian` is defined as a "\[d\]erived variable to indicate how many pedestrians were involved in the crash. This includes pedestrians on skateboards, scooters and wheelchairs." However, non-missing values for `pedestrian` range from \[1,6\], meaning that no crash events have a value of '0' recorded. It is reasonable to assume that crashes with a null value for `pedestrian` actually represent events where no pedestrians were involved, as opposed to events where the number of pedestrians involved were not known or not recorded. Similarly, `holiday` only records an entry if the event falls into one of the four categories of Christmas/New Year, Easter, Labour Weekend, or Queens Birthday. Missing data in this column therefore likely represent events that occurred outside of these holiday dates.

However, the columns `crashRoadSideRoad` and `intersection` are missing all data, and these were removed from the analysis. For `weatherB`, the observed possible values of 'Strong Wind', 'Frost', 'None' or missing do not align with the metadata description (where values are stated as 'Strong Wind', 'Frost' or 'Unknown'). Due to the ambiguity between 'None' and missing data, this column has been removed.

For `temporarySpeedLimit` and `advisorySpeed`, missing data represents a true absence - in these cases, no temporary or advisory speed limit was in effect at the time and location of the crash event. Due to the low proportion of observations with a temporary or advisory speed limit in effect, these fields have been converted to boolean - where a present limit is True and no limit classed as False.

Finally, free-text columns have been removed as these will not be useful in the ML model, as well as a series of redundant fields where the information is contained in another variable. These are `OBJECTID`, `areaUnitID`, `crashDirectionDescription`, `crashFinancialYear`, `crashLocation1`, `crashLocation2`, `crashFinancialYear`, `meshblockId`, `tlaName` and `tlaId`.

A second subset of 23 columns have a non-null count of 325,939, meaning they are missing data for approximately 60% of observations. These are `bridge`, `cliffBank`, `debris`, `ditch`, `fence`, `guardRail`, `houseOrBuilding`, `kerb`, `objectThrownOrDropped`, `otherObject`, `overBank`, `parkedVehicle`, `phoneBoxEtc`, `postOrPole`, `roadworks`, `slipOrFlood`, `strayAnimal`, `trafficIsland`, `trafficSign`, `train`, `tree`, `vehicle`, and `waterRiver`. These columns all collect data on the number of objects of \*type\* that were struck in the crash, and are systematically either present or missing. Because of the large set of columns in this group, a subset of the dataframe has been created which keeps these columns but removes observations where they have missing data.

After removing unnecessary columns and creating the ColumnSetB subset of the dataframe, the data is checked for any remaining missing values.

The missing values in these columns all represent a very small percentage (<0.05%) of the total observations. Therefore, these observations have been removed.

#### Dependent variable

The outcome variable of interest for this study is the crash severity, which is measured by three variables:

* crashSeverity: categorised as Non-injury Crash, Minor Crash, Serious Crash, or Fatal Crash
* seriousInjuryCount: count of the serious injuries associated with this crash
* fatalCount: count of the fatal casualties associated with this crash

As fatal count and serious injury count are both heavily skewed towards zero, crash severity category will be used as the main dependent variable for the ML model. Of the four crash severity categories, there is a comparatively very low count of fatal crashes. Therefore, 'Fatal Crash' and 'Serious Crash' have been combined into a single category for the analysis.

As fatal count and serious injury count are both heavily skewed towards zero, crash severity category will be used as the main dependent variable for the ML model. Of the four crash severity categories, there is a comparatively very low count of fatal crashes. Therefore, 'Fatal Crash' and 'Serious Crash' have been combined into a single category for the analysis.

#### Set data types

Several columns in the dataset represent categorical data, where the input has been coded to one value out of a specific set - for example, the local government region where the crash occurred. Categorical columns in the dataframe are set accordingly, and the levels for each category checked to ensure it matches the metadata provided.

See below for a summary table (Table 1) of each categorical column, broken down proportionally by category. Tables 2 & 3 show the descriptive statistics for numeric variables, for column sets A and B, respectively.

#### Training Testing Split[¶](http://localhost:8888/lab/tree/work/CASA0006/assessment/final-assessment.ipynb#Training-Testing-Split)

Firstly, the dataset is split into training and testing subsets, before any of the preprocessing steps take place. This ensures that there is no data leakage from the testing set, which can lead to inflated performance estimates (Tharwat, 2020).

#### Model Pipeline

To prepare the data for modelling, two pre-processing steps are run: standardisation of numeric variables and one-hot encoding of categorical variables. After defining the pre-processing steps, an estimator is added to the ML pipeline, which specifies the ML algorithms to be used. For this project, a remote forest classifier (Breiman, 2001) has been chosen.

In the ML literature, there are two main approaches to dealing with class imbalanced data: under-sampling and over-sampling (Wen et al. 2019). Due to the large size of the cleaned dataset (still over 300,000 observations), an undersampling method has been applied, as this maintains a high performance but at significantly reduced computational load.

#### Evaluating performance

The balanced\_accuracy\_score function computes the balanced accuracy, which avoids inflated performance estimates on imbalanced datasets. Additionally, the geometric mean (G-mean), calculated as the square root of the product of sensitivity and specificity measures, has been proposed as a more appropriate metric for evaluating models developed from imbalanced datasets (Jeong et al. 2018).

#### Iterate through multiple models

After testing proof-of-concept with the RF model above, a loop is built to iterate through alternative models and compare the performance results.

#### Hyperparameter tuning

The following method finds the optimal values for RF hyperparameters, from the specified set. For this model, the best performing set used a max tree depth of 50 and a minimum leaf sample of 2.

**NOTE:** The tuning cell takes approximately 40 minutes to run. Therefore, has been default set to not run, unless explicitly called by setting run\_tuning to True.

### Discussion

### References

Breiman, L. (2001) ‘Random Forests’, *Machine Learning*, 45(1), pp. 5–32. Available at: https://doi.org/10.1023/A:1010933404324.

Delen, D. *et al.* (2017) ‘Investigating injury severity risk factors in automobile crashes with predictive analytics and sensitivity analysis methods’, *Journal of Transport & Health*, 4, pp. 118–131. Available at: https://doi.org/10.1016/j.jth.2017.01.009.

Effati, M., Thill, J.-C. and Shabani, S. (2015) ‘Geospatial and machine learning techniques for wicked social science problems: analysis of crash severity on a regional highway corridor’, *Journal of Geographical Systems*, 17(2), pp. 107–135. Available at: https://doi.org/10.1007/s10109-015-0210-x.

Iranitalab, A. and Khattak, A. (2017) ‘Comparison of four statistical and machine learning methods for crash severity prediction’, *Accident Analysis & Prevention*, 108, pp. 27–36. Available at: https://doi.org/10.1016/j.aap.2017.08.008.

Jeong, H. *et al.* (2018) ‘Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data’, *Accident Analysis & Prevention*, 120, pp. 250–261. Available at: https://doi.org/10.1016/j.aap.2018.08.025.

New Zealand Government (2019) *Road to Zero: New Zealand’s Road Safety Strategy 2020-2030*. Wellington, NZ. Available at: https://www.nzta.govt.nz/safety/what-waka-kotahi-is-doing/nz-road-safety-strategy/.

Santos, K., Dias, J.P. and Amado, C. (2022) ‘A literature review of machine learning algorithms for crash injury severity prediction’, *Journal of Safety Research*, 80, pp. 254–269. Available at: https://doi.org/10.1016/j.jsr.2021.12.007.

Savolainen, P.T. *et al.* (2011) ‘The statistical analysis of highway crash-injury severities: A review and assessment of methodological alternatives’, *Accident Analysis & Prevention*, 43(5), pp. 1666–1676. Available at: https://doi.org/10.1016/j.aap.2011.03.025.

Tharwat, A. (2020) ‘Classification assessment methods’, *Applied Computing and Informatics*, 17(1), pp. 168–192. Available at: https://doi.org/10.1016/j.aci.2018.08.003.

Waka Kotahi NZ Transport Agency (2023) *Crash Analysis System (CAS) Open Data*. Available at: https://opendata-nzta.opendata.arcgis.com/search?tags=CAS (Accessed: 20 April 2023).

Wang, X. and Kim, S.H. (2019) ‘Prediction and Factor Identification for Crash Severity: Comparison of Discrete Choice and Tree-Based Models’, *Transportation Research Record*, 2673(9), pp. 640–653. Available at: https://doi.org/10.1177/0361198119844456.

Ziakopoulos, A. and Yannis, G. (2020) ‘A review of spatial approaches in road safety’, *Accident Analysis & Prevention*, 135, p. 105323. Available at: https://doi.org/10.1016/j.aap.2019.105323.