milestone3

Frank Guo

2024-05-04

Milestone 3 - Crashes Analysis

By Xiaodong Guo

1. Goal.

Our project has a significant objective- to construct models identifying the pivotal factors contributing to severe crashes. This crucial task is based on the data provided by the New Zealand Transport Agency(NZTA).

2. Data Source.

The original data, meticulously collected, originated from the Waka Kotahi NZ Transport Agency's open data portal(the tutor provided the link in the assignment piece). We specifically downloaded the dataset named "Crash Analysis System (CAS) data" from the "Crash" catalogue, which encompasses all traffic crashes reported to us by the NZ Police. The data format is a "CVS" file. It was created on 3/25/2020 and last updated on 3/14/2024.

The data includes crash datas from 2000 to 2023.

3. Data Processing.

We load the data from csv flie. The dataset we got have 72 columns,and 821744 rows.

```
## [1] 821744     72
```

The first things we can do is to drop columns not related to our objective.Thus, we select following columns by common sense of mine(not sure if 100% correct,maybe need some hint from Lisa Chen).There are also have some description columns, that is long string values to desript an event or street name. That looks no sense,should drop them too. Like "crashLocation1","crashLocation2".

```
#clean data
columns_to_drop <- c("X","Y","OBJECTID","areaUnitID","crashDirectionDescription","","crashDistance","tla
data <- select(data, -one_of(columns_to_drop))
```

Then drop columns that almost all values are Null(more then 99% of the data is null). Columns like these are too sparse for the inference. The column names are crashRoadSideRoad" and "intersection".

```
na_percentage <- colMeans(is.na(data))
columns_with_high_na <- names(na_percentage[na_percentage > 0.99])
#print(columns_with_high_na)

data <- data %>% select(-columns_with_high_na)

#table(data$crashSeverity)
```

Define crashSeverity == “Fatal Crash” and crashSeverity == “Serious Crash” as severe crashes given numeric value 1,

Define crashSeverity == “Minor Crash” | crashSeverity == “Non-Injury Crash” as not severe crashes given numeric value 0.

The “crashSeverity” Label will be the target label.

```
data <- data %>%
  mutate(crashSeverity = ifelse(crashSeverity == "Fatal Crash" | crashSeverity == "Serious Crash", 1,
                        ifelse(crashSeverity == "Minor Crash" | crashSeverity == "Non-Injury Crash", 0,
                               2))) %>%  filter(crashSeverity != 2)
```

Attributes “weatherA” and “weatherB”, are String values could be treated as factors,not too many factors in each attribute, and the combinations also not too many factors but are more sensitive to understand the whole weather situation. In my opinion, these two could be combined as one attribute “weather”,much easier to display and dealing with it later.

The Na value or String “None” are replaced as “Others” condition,

```
data$weatherA <- ifelse(data$weatherA %in% c("None", "Null"), "Others", data$weatherA)
data$weatherB<- ifelse(data$weatherB %in% c("None", "Null"), "", data$weatherB)

data <- data %>% unite(weatherA,weatherB,col=weather,sep=" ")
```

Replace the “”,”Null”,“None” value in region with “Others”;

Replace the “ ” in other character attributes with “Others”.

```
#knitr::kable(table(data$region))

data$region <- ifelse(data$region %in% c("None", "Null"), "Others", data$region)

data[data == ""] <- "Others"
```

List all the attributes with Na value(in Appendix).

According the descriptions of these attributes, we can use 0 to fill na value.Here is 2 examples about why 0 be used:

For “advisorySpeed” or “temporarySpeedLimit” attribute, the value is mean special speed limitation applied or advised in the road which is involed in the crash. use 0 here means no special speed limit applied(according the rode code, that is open road.follows open road speed limit).

For other attributes in the list upon, the value indicates the number of item involved in the crash. the Na value means no item(named by attribute name) is involved,that equals to 0.

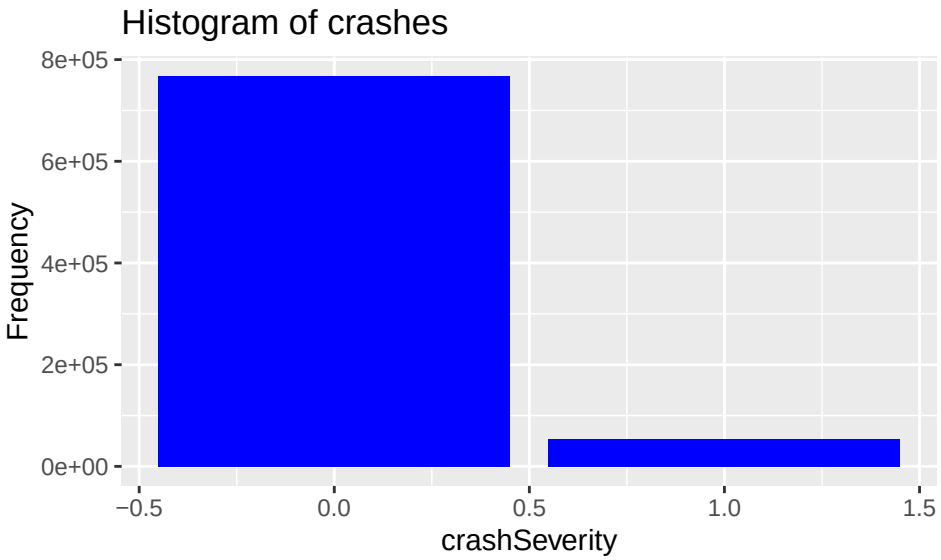
At the end, all the Na or missing data is imputated and remedied.

```
na_columns <- sapply(data, function(x) any(is.na(x)))
columns_with_na <- names(data)[na_columns]

#print(columns_with_na)

data <- data %>%
  mutate_at(vars(one_of(columns_with_na)), ~replace_na(., 0))

#glimpse(data)
```

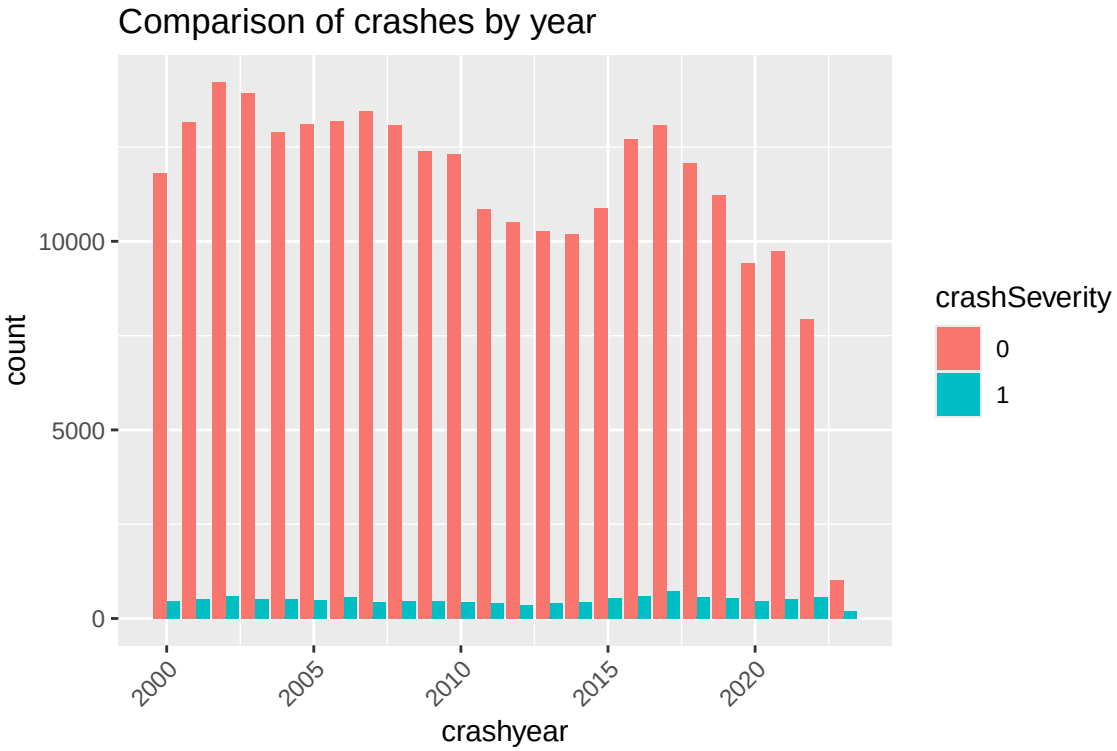
4. Data Exploration.

First of All, we can find the unbalance of the target class..

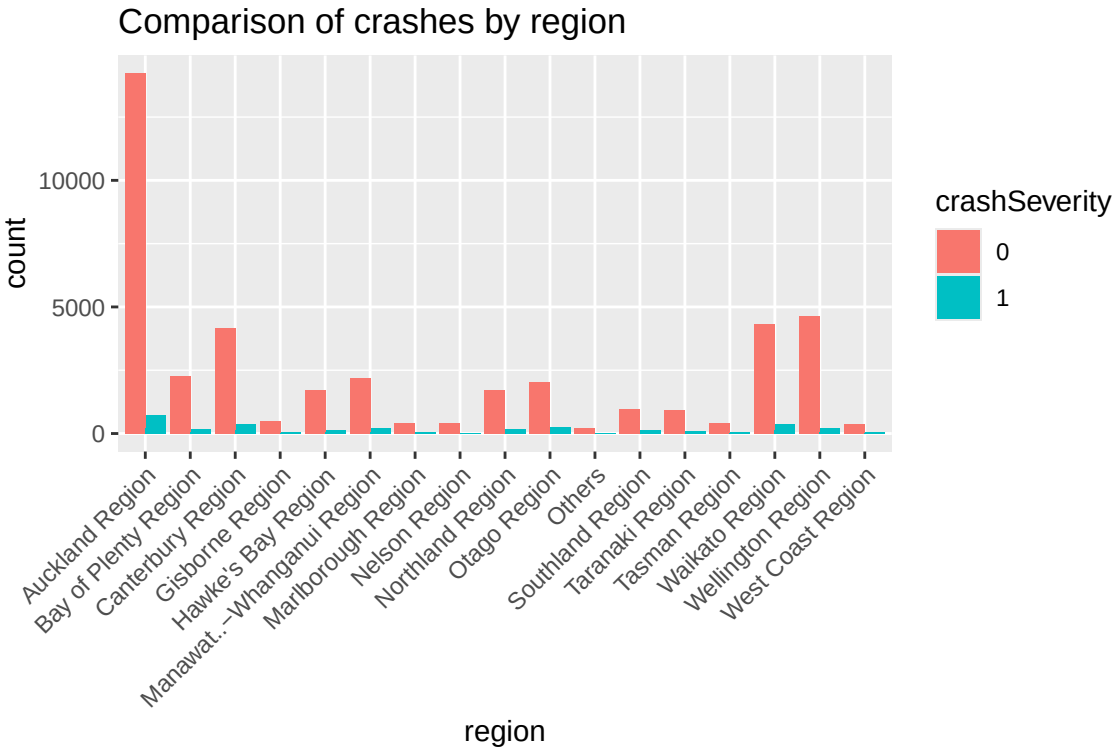
There are 767290 regular crashes and 54454 severe crashes. Our target is to find the cause of severe crashes,but the severe observations' size is really small compares to the regular crashes. So we can't use the whole dataset directly, it will overfit the majority(regular crashes) and underfit the minority(severe crashes), cause bias to majority,loss the importance information for our target(server crashes) inference.



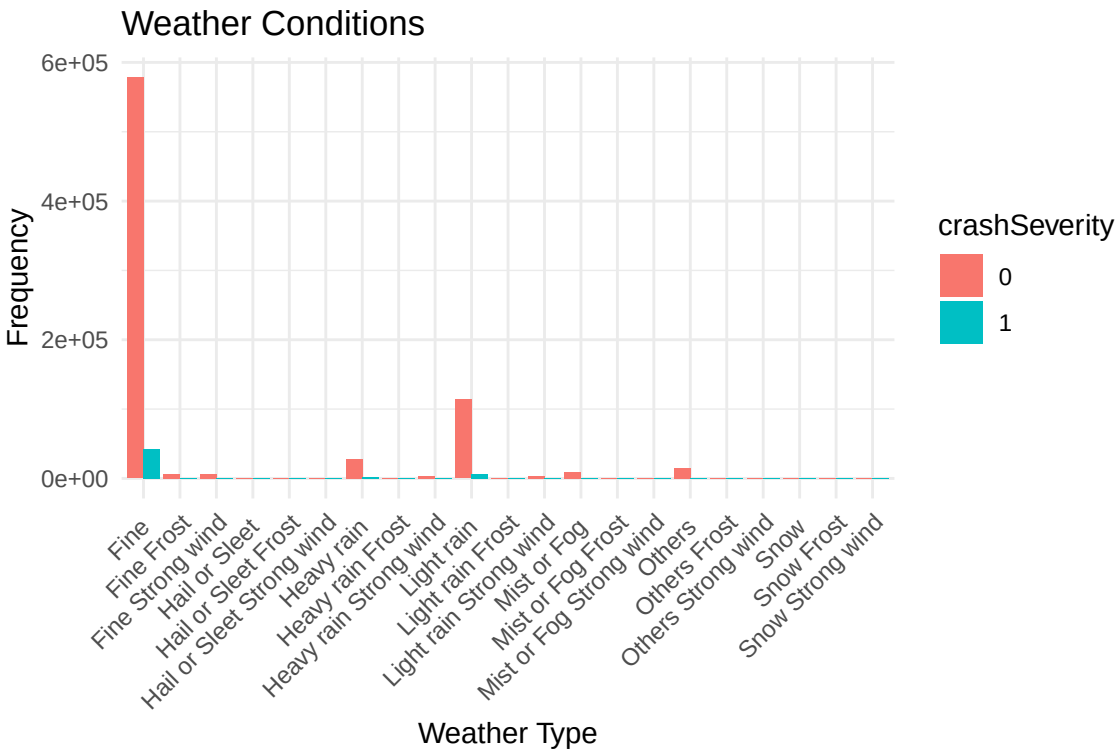
Comparing the crashes count by year, Though the regular crashes had fluctuated from year 2000 to 2022(since the data of 2023 is not complete yet.),but the regular crashes number in 2020,2021 and 2022 is obviouly smaller,that maybe match the Covid lockdown situation during these years. Regarding as the number of severe crashes, it is very steady during these years, even not be influenced by Covid.



Comparing the crashes by region, Auckland region looks obviously high than others, considering of population density, it looks reasonable.While the ratio of severe crashes looks low. While regions like Gisbone,northland, southland, hawkesbay and westcoast looks has much hight ratio of severe crashes.

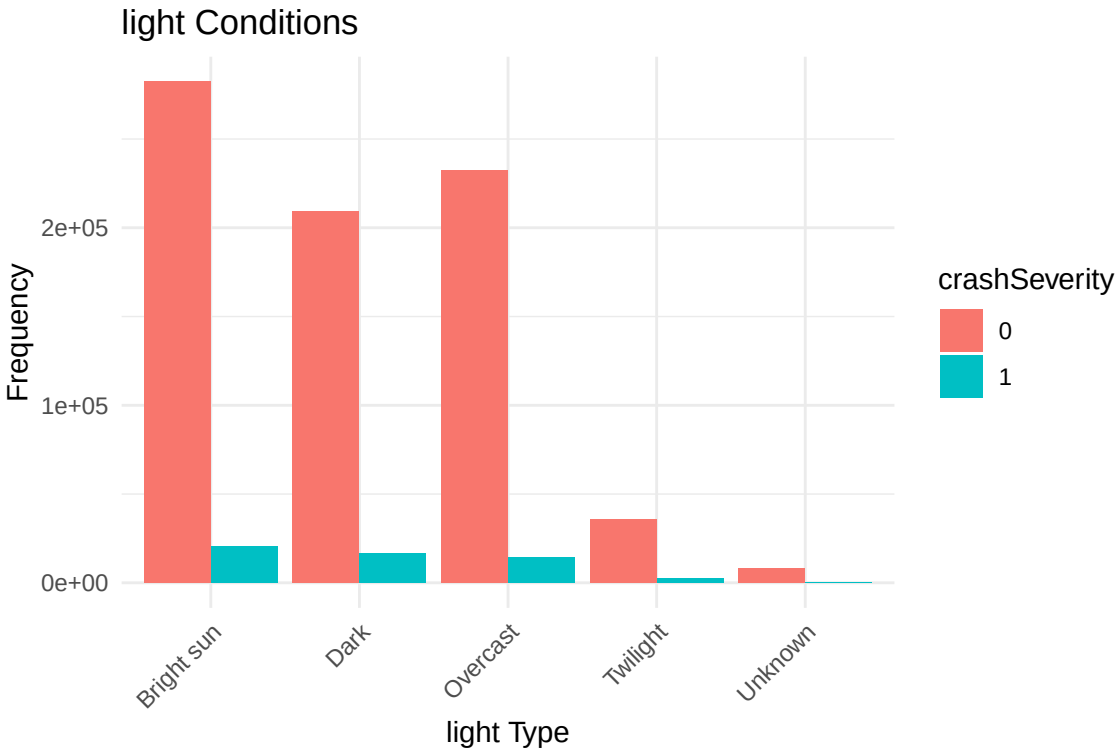


Most of crashes happen in fine weather, but the light rain weather looks notable too.

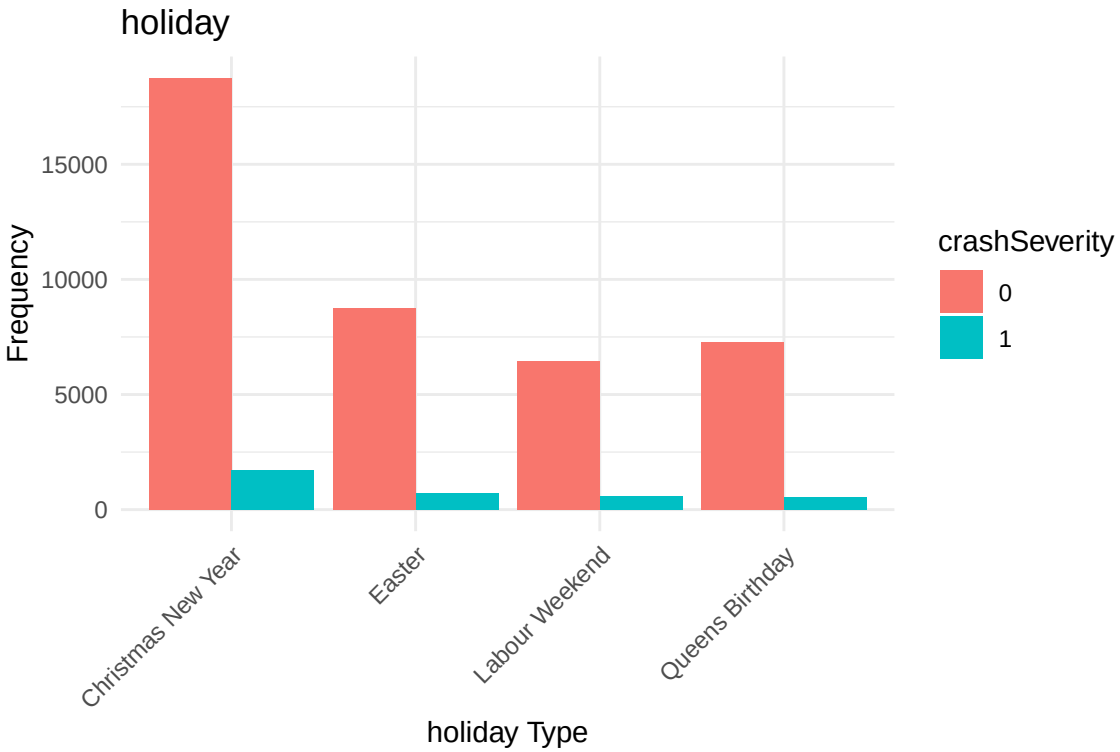


Crashes happened all the light situation,the number of severe crashes looks almost the same in sunny ,dark

or overcast.While in dark or twilight,the severe crashes ratio looks higher.



If we chech the crashes happened in holidays, it shows that Christmas New Year have higher crashes numbers and severe crashes. The severe crashes in other holidays are almost the same.



5. Analytical Plan

Because of the unbalanc of the dataset. We decisied to equally bootstrap from severe crashes and regular crashes,that is 5000 observations from each.

We will use 80% of the sample data as training data, and 20% of the sample data as test data. To fit the model, we will factorise the charactor attributes,and numberic them.

Fit logistic regression model. using lr_model <- glm(crashSeverity ~ ., data = training_data, family = "binomial"). fit a logistic regression model with given training data, and predict using : predictions <- predict(lr_model, newdata = test_lr, type = "response") with sampled test_data.

Output accuracy and sorted first 15 importance of coefs.

	0	1
0	740	292
1	264	704

```
## [1] "Accuracy is:"
```

```
## [1] 0.722
```

	predictor	coefficient	abs_coefficient
temporarySpeedLimit	temporarySpeedLimit	-0.0030427	0.0030427
crashYear	crashYear	0.0040643	0.0040643
region	region	0.0044349	0.0044349
streetLight	streetLight	0.0050716	0.0050716
advisorySpeed	advisorySpeed	0.0053881	0.0053881
roadCharacter	roadCharacter	0.0070847	0.0070847
taxi	taxi	0.0181197	0.0181197
trafficControl	trafficControl	0.0185655	0.0185655
light	light	-0.0247316	0.0247316
speedLimit	speedLimit	0.0359725	0.0359725
holiday	holiday	-0.0412843	0.0412843
flatHill	flatHill	0.0439846	0.0439846
weather	weather	-0.0444815	0.0444815
unknownVehicleType	unknownVehicleType	0.0654334	0.0654334
roadSurface	roadSurface	0.0865774	0.0865774

fit decision tree model with random forest ensemble

using rf_model <- ranger(crashSeverity ~ ., data = training_data,importance = "impurity") fit the training data with random forest. and predict using: predictions <- predict(rf_model, data = test_data).

Ouput the accuracy of predicted value for with sampled test_data. Then list the first 15 importance of attributes.

	0	1
0	701	214
1	303	782

```
## [1] "Accuracy is:"
```

```
## [1] 0.7415
```

	importance_measures
crashYear	320.66923
carStationWagon	252.72895
region	252.29751
motorcycle	205.49293
pedestrian	184.44004
speedLimit	172.21453
streetLight	136.55853
trafficControl	128.40618
light	115.49994
NumberOfLanes	105.05812
weather	101.58563
urban	98.70049
bicycle	69.46378
vanOrUtility	65.65178
flatHill	60.08474

In summary,Logistic regression model and Random forest ensemble works well in this scenarios. Random forest ensemble has better predict accuracy,but not too much. The sorted importance of factors are different with two methods. The Logistic regression model looks more confident to interpret by choose coefs with P value < 0.05 significant level

Discussion.

Our target is about the minority value in target class, using cluster sampling the Minority Class and the Minority Class to get equally sample size from original data set.

The Data provided not only unbalance in target class,but also very unbalance in predictors.It makes troubles when I try to factorise the attributes and fit a model. Logistic regression model and Random forest ensemble of decision based on ranger package looks good after convert all the predictor as numeric.

Appendix.

List all the attributes with Na value,got attributes below:
“advisorySpeed” “bicycle” “bridge” “bus” “carSta- tionWagon” “cliffBank” “ditch” “fence” “guardRail” “house-OrBuilding” “kerb” “moped” “motorcycle” “NumberOfLanes” “otherVehicleType” “overBank” “parkedVehicle” “pedestrian” “postOrPole” “roadworks” “schoolBus” “slipOrFlood” “speedLimit” “strayAnimal” “suv” “taxi” “temporarySpeedLimit” “trafficIsland” “trafficSign” “train” “tree” “truck” “unknownVehicleType” “vanOrU-tility” “vehicle” “waterRiver”