Predictive Analysis of Crash Incidents in Hillsborough County

By: Bo Fethe

Group 123
April 2024

Table of Contents

Tables

Figures

Abstract

Hillsborough County is a rapidly growing region located in southwest Florida and includes 3 incorporated cities: Tampa, Plant City, and Temple Terrace. As a region's population grows, the demand on its transportation infrastructure and the importance of incorporating safety designs also increase. Unintentional injury was the #3 leading cause of death in Florida during 2022 (Florida Department of Health, 2023). To help reduce or eliminate traffic fatalities and serious injuries, transportation authorities and international programs research and test various roadway designs to improve public safety. In addition, the state maintains a digital archive of all crash events for making data-driven decisions using prescriptive analytics possible. This initiative raises the question of how well can machine learning predict the severity of automotive crashes.

Using all crash events in 2023 within Hillsborough County, stepwise variable selection was performed to reduce dimensionality to 15 predictors and 1 binary response representing the case of an incapacitating event. Geographically, most crashes occurred along local roads when considering all crashes as well as only incapacitating events. This implies that while roadway safety is a high concern for all agencies, this is largely a concern for the 3 cities within Hillsborough County. Crashes were more common during the middle of the day on a weekday ($\frac{12,614}{43,960}$; 29%) while off-peak traffic hours at night had the highest number of incapacitating events ($\frac{410}{1,068}$; 38%). Crash severity is naturally unbalanced in favor of not being involved in an incapacitating event which can cause issues for model training and testing purposes, so random over-sampling was used on the training data prior to the analysis to balance the response variable.

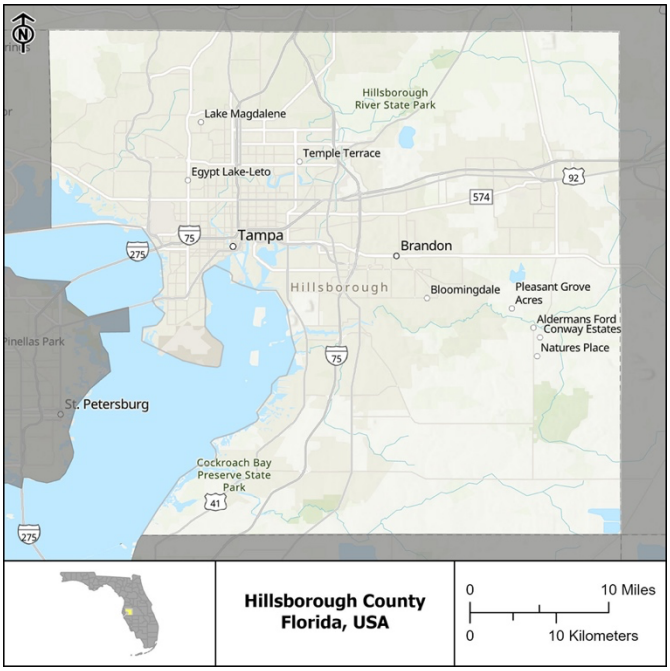
The results of 3 models were compared with and without the stepwise-selected predictors: logistic regression, random forest, and generalized boosting regression. In summary, generalized boosting regression with all predictors had the lowest testing error of 0.0271, but struggled predicting true incapacitating events with a specificity of 0.0331, while logistic regression with stepwise predictors had the highest specificity of 0.7101, but misclassified more crashes overall compared to the other models with a testing error of 0.1981.

1. Introduction

Hillsborough County, located in southwest Florida, is home to a diverse population and a rapidly expanding network of roadway. Given its urban communities, tourism appeal, access to natural resources, and well-developed transportation network, the county sees substantial Average Annual Daily Traffic (AADT) levels on its roadways. Unfortunately, along with this traffic comes the risk of road accidents and crashes, posing challenges to public safety and transportation management. Hillsborough County's population is rapidly growing and is expected to increase by 461,371 (+30.4%) from 2020-2050 (Plan Hillsborough, 2023), which causes roadway conditions to become more congested; and thus, increases the probability of traffic accidents occurring. The purpose of this report is to serve as a comprehensive examination of all crash data in Hillsborough County to offer valuable insights into the dynamics of traffic accidents to support safer roadways and communities.

Machine learning models trained on historical crash data can forecast the probability of future accidents along specific corridors or under certain conditions. Analyzing these events is crucial to recognize patterns, trends, and contributing factors, enabling authorities to prioritize safety improvements at specific locations. Many governments integrate principles from the Vision Zero initiative, aiming to eliminate fatalities and serious injuries in road traffic incidents, into their design strategies (Vision Zero Network, 2024). Predictive analytics enable stakeholders to incorporate best practices during the design phase, leveraging machine learning to comprehend spatial and temporal trends, pinpoint key crash contributors, and develop anticipatory models. By doing so, preventative measures can be integrated into transportation designs, mitigating the occurrence of crash events.

Figure 1: Study Area



2. Methodology

2.1. Software

The project uses the programming language R version 4.3.2 (“Eye Holes”) for statistical analysis. In addition, ArcGIS Pro version 2.9 is used for generating maps and ArcGIS Online is used for creating an interactive dashboard for data exploration. All data and code used for this analysis is available on the project’s GitHub (Fethe, signal-four, 2024)

2.2. Data Sources

Signal Four Analytics is a statewide geospatial crash analysis system developed and hosted by the University of Florida’s GeoPlan Center (University of Florida, 2024). This system receives data from Florida’s statutory custodian of records, the Florida Department of Highway Safety and Motor Vehicles (FLHSMV). The data used for this analysis was obtained on March 6, 2024, and filtered to all crash events in the year of 2023 within Hillsborough County.

Due to the number of columns represented in Signal Four Analytics, only the following columns were considered in this analysis:

Table 1: Data Columns Used

Name	Description
INCAPACITATING FLAG	Binary response whether the crash severity resulted in a serious injury/fatality or not derived from S4 CRASH TYPE SIMPLIFIED
PEAK	Peak traffic period derived from CRASH DATE AND TIME
TOTAL NUMBER OF VEHICLES	Count of all vehicles involved in the crash
TOTAL NUMBER OF PERSONS	Count of all persons involved in the crash (drivers, passengers, and non-motorists)
RURAL OR URBAN	Check if the traffic crash occurred inside the corporate limits of the city
ROAD SYSTEM IDENTIFER	This classification is used to identify the primary road system on which the traffic crash occurred
ROAD SURFACE CONDITION	This classification is used to identify the surface condition of the street, road or highway at the time of the traffic crash
S4 CRASH TYPE SIMPLIFIED	Crash type simplified for practitioners who desire less detailed crash types
S4 DAY OR NIGHT	To identify if the crash happened in the daytime or nighttime
S4 IS AGGRESSIVE DRIVING	To identify the presence of aggressive driving in the crash
S4 IS ALCOHOL RELATED	To identify the presence of alcohol by the driver in the crash
S4 IS CMV INVOLVED	To identify the presence of a commercial motor vehicle is involved in the crash
S4 IS DISTRACTED	To identify the presence of driving distraction in the crash
S4 IS DRUG RELATED	To identify if the crash is drug related due to refusal of drug test or positive drug test of driver

Name	Description
S4 IS HIT AND RUN	To identify if the crash is hit and run related
S4 IS INTERSECTION RELATED	To identify if the crash is intersection related
S4 IS LANE DEPARTURE RELATED	To identify the presence of lane departure in the crash
S4 IS SPEEDING RELATED	To identify if the crash is speed related
S4 TRAILER COUNT	Number of trailers in the crash
S4 MOTORCYCLE COUNT	Number of motorcycles in the crash
S4 MOPED COUNT	Number of mopeds in the crash
S4 BICYCLIST COUNT	Number of bicyclists in the crash
S4 AGING DRIVER COUNT	Number of aging drivers in the crash who are 65 or older
S4 TEENAGER DRIVER COUNT	The number of drivers involved in the crash whose age at time of crash is between 15 and 19
S4 UNRESTRAINED COUNT	Number of motor vehicle occupants not using restraint system(s) at time of crash

2.3. Analytical Methods

This analysis uses machine learning to determine how well the severity of crash events can be predicted and if there are any regions with a statistically significant amount of crash events. Due to the nature of the data, it is anticipated the severity classification in the data raw data will be unbalanced as most crash events do not result in an injury, and that will be handled using techniques such as random over-sampling examples (ROSE) for the training datasets only to ensure each response factor has 2000 results. In addition, the time values are placed into bins for the peak traffic hours of AM Peak (M-F; 07:00 – 10:00), Mid Peak (M-F; 10:00-16:00), PM Peak (M-F; 16:00-19:00), Weekend (Sa-Su; 06:00-20:00), and Off Peak for all other values for compatibility between other transportation model results (Florida Department of Transportation Systems Forecasting and Trends Office, 2023).

This analysis uses the following models to predict the severity of crash events occurring events occurring:

- Logistic Regression (Logit): A probabilistic classification model used for binary classification. Suitable for scenarios when you need to understand variable relationships the probability of a particular outcome.
- Random Forest (RF): An ensemble bagging model that bootstrap aggregates multiple, independent decision trees using randomly sampled subsets of data and features to predict or classify data. This is beneficial for handling large datasets with high dimensionality and complexity.
- Generalized Boosting Regression (GBM): An ensemble gradient boosting model that iteratively builds decision trees and trains weak learners to correct the weights used by the previously models. Suitable for a wide range of supervised learning tasks when you can tolerate longer training times.

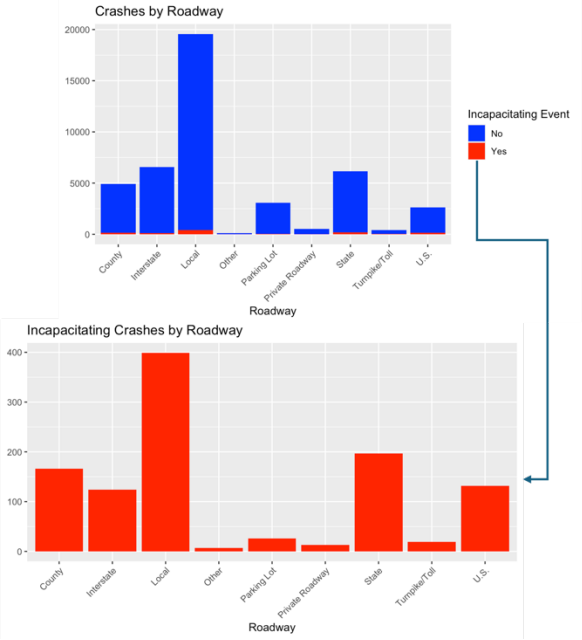
To measure performance, each model will predict the response variable using the reserved testing dataset in its natural state and compared to the actual values to calculate the testing error.

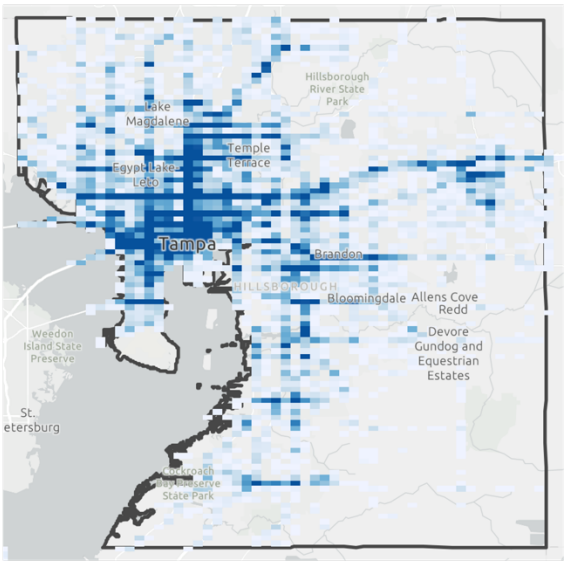
3. Results

3.1. Exploratory Data Analysis

In 2023, Hillsborough County had 43,960 reported crashes consisting of 1,068 incapacitating events. The charts below represent the distribution of incapacitating events by crash type and peak traffic period. For more details and charts, visit the interactive dashboard found here.

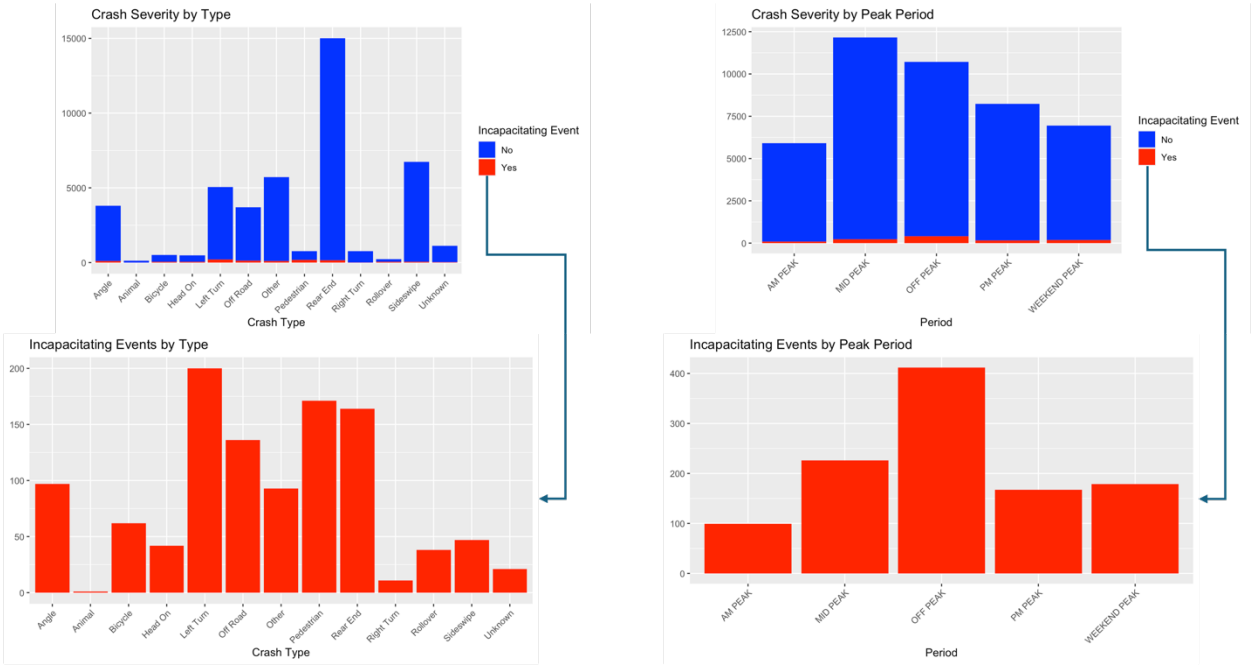
Figure 2: Crash Location





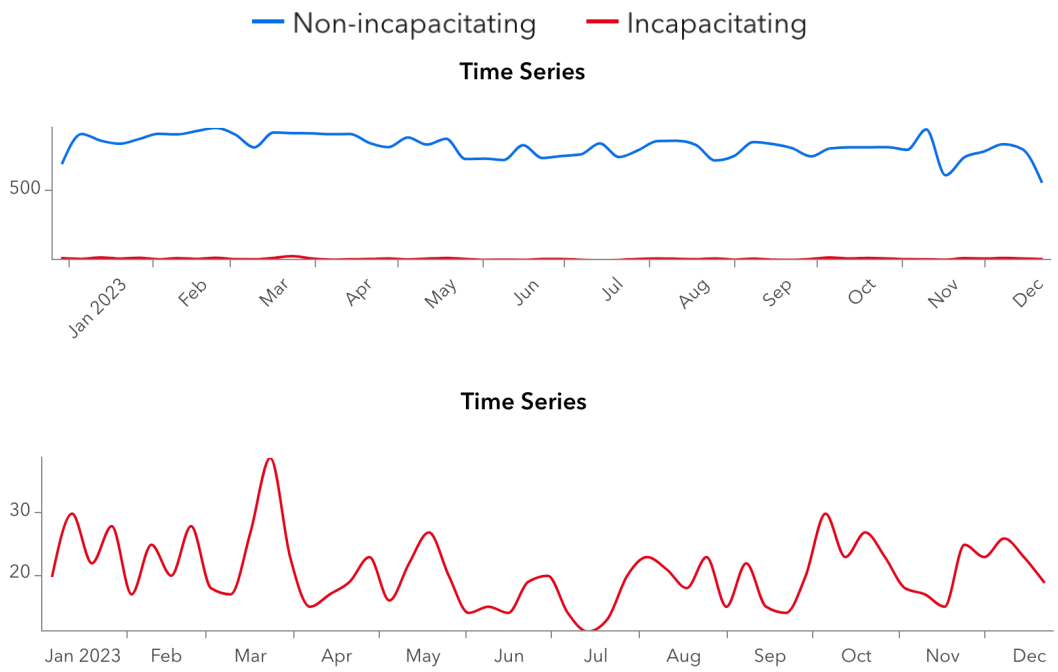
The bin aggregate map shows that the majority of the crashes occur within city limits with linear tangents along the highways heading east towards Orlando and north towards Gainesville. A significant portion of all crashes occurred on local roadways with almost 3x more events than the next leading roadway system. This trend stays relatively true when considering only incapacitating events with the proportion dropping closer to 2x more than the next leading system.

Figure 3: Crash Summary



These charts show that the majority of crashes are classified as rear ends and occur in the Mid Peak traffic hours. When specifically referring to incapacitating events, crossing over traffic while making a left turn shows to have the highest count of events with accidents involving pedestrians and rear ends being the next highest crash types in the off-peak hours when congestion is likely not as high; thus, allowing faster travel speeds. For crash types, most crashes are classified as rear end events, but when considering only incapacitating events, left turn, pedestrians, rear ends, and off-road were the most common.

Figure 4: Crash Time Series



The plots above show the crash temporal trends. Overall, there is little to no seasonality when considering all crash severities with a dip in November. When focusing on incapacitating events, winter and fall months have more events compared to summer and spring that coincides with temporary transplants escaping the harsh winters in northern climates. There is also a notable spike in March that coincides with the spring break vacation. Holidays with an anticipated higher volume of traffic such as Independence Day and Halloween do not show to be significant based on these plots.

3.2. Variable Selection

A stepwise regression with the penalty term k=2 degrees of freedom for an AIC method was used to reduce dimensionality by selecting only the predictors that contribute to the model's performance. The final selection resulted in 15 predictors:

Table 2: Variable Selection Predictors

TOTAL NUMBER OF VEHICLES	S4 CRASH TYPE SIMPLIFIED	S4 IS HIT AND RUN	S4 MOPED COUNT
TOTAL NUMBER OF PERSONS	S4 IS AGGRESSIVE DRIVING	S4 IS INTERSECTION RELATED	S4 AGING DRIVER COUNT
RURAL OR URBAN	S4 IS ALCOHOL RELATED	S4 IS LANE DEPARTURE RELATED	S4 UNRESTRAINED COUNT
ROAD SYSTEM IDENTIFER	S4 IS DRUG RELATED	S4 MOTORCYCLE COUNT	

3.3. Logistic Regression

A logistic regression model was trained using a binomial distribution. The results of the prediction accuracy for the full model as well as the reduced model with only the stepwise predictors are shown below.

Table 3: Logistic Regression Confusion Matrix

Logit	N	Y
N	3429	38
Y	846	83
Test Error	0.2011	
Specificity	0.6860	

Logit-SW	N	Y
N	3439	35
Y	836	86
Test Error	0.1981	
Specificity	0.7101	

3.4. Random Forest

The Random Forest model was trained with the consideration for the importance of the predictors and the proximity measure among rows. The results of the prediction accuracy for the full model as well as the reduced model with only the stepwise predictors are shown below.

Table 4: Random Forest Confusion Matrix

RF	N	Y
N	4164	91
Y	111	30
Test Error	0.0460	
Specificity	0.2479	

RF-SW	N	Y
N	4169	87
Y	106	34
Test Error	0.0439	
Specificity	0.2810	

3.5. Generalized Boosting Regression

The boosting algorithm was initially trained with 5000 trees, a shrinkage factor of 0.01, and interaction depth of 1, and 10 cross-fold validations. Cross-fold validations was used to find the optimal number of iterations to be used for prediction, which results in 4856 iterations. The results of the prediction accuracy for the full model as well as the reduced model with only the stepwise predictors are shown below.

Table 5: Boosting Confusion Matrix

GBM	N	Y
N	4273	117
Y	2	4
Test Error	0.0271	
Specificity	0.0331	

GBM-SW	N	Y
N	4260	109
Y	15	12
Test Error	0.0282	
Specificity	0.0992	

Table 6: GBM-Stepwise Relative Influence

Value	Rel. Influence
S4 MOTORCYCLE COUNT	44.08
S4 MOPED COUNT	31.22
S4 CRASH TYPE SIMPLIFIED	9.93
TOTAL NUMBER OF VEHICLES	6.38
S4 UNRESTRAINED COUNT	2.61
S4 AGING DRIVER COUNT	2.27
TOTAL NUMBER OF PERSONS	1.61
ROAD SYSTEM IDENTIFER	0.96
RURAL OR URBAN	0.43
S4 IS HIT AND RUN	0.20
S4 IS DRUG RELATED	0.17
S4 IS AGGRESSIVE DRIVING	0.12
S4 IS LANE DEPARTURE RELATED	0.01
S4 IS ALCOHOL RELATED	0.01
S4 IS INTERSECTION RELATED	0.00

In addition, each predictor's relative influence was calculated and is shown in the table above. This table shows 95% of the relative influence of determining if a crash event is incapacitating or not using boosting is made up of number of motorcycles, number of mopeds, crash type, number of vehicles, and number of unrestrained passengers.

4. Conclusion

When comparing the results of these models, it's important to understand what metric is more important based on the background of the data. Testing error can give you a good sense of how well the model predicts overall across all classes. In this example, generalized boosting regression with all predictors showed the lowest testing error at 0.0271 while logistic regression with all predictors had the highest at 0.2011. However, crash data is naturally unbalanced being skewed in favor of not having an incapacitating event. Specificity measures the proportion of actual negative results that are correctly identified by the model, meaning how many incapacitating events were correctly identified as such. Incorrectly predicting a crash event as incapacitating when it is non-incapacitating likely results in additional safety measures and isn't as nearly as costly of an error as missing an incapacitating event all together since that could have potentially saved someone's life. Logistic regression with stepwise predictors had the highest specificity of 0.7101 and generalized boosting regression with all predictors had the lowest of 0.0331.

The results of this analysis show that there are challenges when using machine learning and data collected from FLHSMV and provided by Signal Four Analytics. Human driving behavior has a lot of randomness that might be a challenge to model. Also, when dealing with situation with heavy

consequences such as providing details to an officer after an accident, information may be withheld or fabricated to protect themselves and potentially avoid financial burdens. There are also predictors not included in this analysis that could potentially increase the performance. A binary indicator whether speeding was involved was included in the analysis, but there is a significant difference between going 5mph over the speed limit than going 50mph over the speed limit. Real-time travel metrics obtained from a car's On-Board Diagnostics (OBD2) can provide valuable insight into what the machine was experiences at the time of the crash.

4.1. Lessons Learned

I found this topic very interesting because it is using recent and real data that affects real people in Hillsborough County. During this project, I learned how challenging it is working with unbalanced data. There are different techniques on how to handle it, whether it's over-sampling, under-sampling, or synthetic sampling, and they each have their weaknesses and can decrease your model's performance if not used correctly, such as generating too many synthetic samples. In addition, I had to learn how to navigate unbalanced data on a database with over 40,000 records, so code efficiency and finding the right train/test splitting size with cross-validation was a new challenge that sample datasets don't have, and a larger training dataset doesn't necessarily mean better performance. I also found the discrepancy between which model is better based on what performance metric is used a valuable insight as I'm sure it's very tempting to just look at overall accuracy and the model with the highest accuracy as the best model, but the decision could be a matter of life or death. Please let me know it you think this project is worthy of submitting to a journal for publication, or even a Medium post.

References

Fethe, B. (2023). *Hillsborough County Crashes*. Retrieved from ArcGIS Online: https://www.arcgis.com/home/item.html?id=131319e3bac346e8a3054e43ba6ee53b

Fethe, B. (2024). *signal-four*. Retrieved from GitHub: https://github.com/bofethe/signal-four

Florida Department of Health. (2023). *Leading Causes of Death Profile*. Retrieved from https://www.flhealthcharts.gov/ChartsReports/rdPage.aspx?rdReport=ChartsProfiles.LeadingCausesOfDeathProfile

Florida Department of Transportation Systems Forecasting and Trends Office. (2023). *FDOT Source Book Methodologies: A Technical Report.*

Plan Hillsborough. (2023). *Hillsborough County attracting most new residents through 2050*. Retrieved from https://planhillsborough.org/hillsborough-county-attracting-most-new-residents-through-2050

University of Florida. (2024). *Florida Traffic Safety Dashboard*. Retrieved from Signal Four Analytics: https://signal4analytics.com

Vision Zero Network. (2024). *What is Vision Zero*. Retrieved from https://visionzeronetwork.org/about/what-is-vision-zero

Appendix

Signal Four Analytics in Hillsborough County

Prep the workspace

```
rm(list=ls())
```

Import libraries

```
library(ggplot2)
library(lubridate)
library(dplyr)
library(caret)
```

load the data

```
data = read.csv('data/crash_event.csv', sep=',', header=T)
head(data)
```

Add FDOT's peak period based on the crash date and time.

```
data = data %>%
  mutate(CRASH_DATE_AND_TIME = dmy_hm(CRASH_DATE_AND_TIME)) %>%
  mutate(PEAK = as.factor(ifelse(
              between(as.numeric(format(CRASH_DATE_AND_TIME, "%H")), 7, 9) & !(weekdays(CRASH_DATE_AND_T
                  ifelse(
              between(as.numeric(format(CRASH_DATE_AND_TIME, "%H")), 10, 15) & !(weekdays(CRASH_DATE_ANI
                    ifelse(
              between(as.numeric(format(CRASH_DATE_AND_TIME, "%H")), 16, 18) & !(weekdays(CRASH_DATE_ANI
                     ifelse(
              between(as.numeric(format(CRASH_DATE_AND_TIME, "%H")), 6, 19) & (weekdays(CRASH_DATE_AND_T
            "OFF PEAK"
        ))))))) %>%

  mutate(INCAPACITATING_FLAG = ifelse(
    S4_CRASH_SEVERITY %in% c('Fatality', 'Serious Injury'), 1, 0)) %>%

  mutate(ROAD_SURFACE_CONDITION = ifelse(
    ROAD_SURFACE_CONDITION %in% c('Sand', "Mud, Dirt, Gravel"),
                           "Mud, Dirt, Gravel, Sand",
                           ROAD_SURFACE_CONDITION))

summary(as.factor(data$INCAPACITATING_FLAG))
```

Filter to only relevant columns for analysis and ensure proper datatypes

```r
selected_columns <- c('INCAPACITATING_FLAG',
                      'PEAK',
                      'TOTAL_NUMBER_OF_VEHICLES',
                      'TOTAL_NUMBER_OF_PERSONS',
                      'RURAL_OR_URBAN',
                      'ROAD_SYSTEM_IDENTIFER',
                      'ROAD_SURFACE_CONDITION',
                      'S4_CRASH_TYPE_SIMPLIFIED',
                      'S4_DAY_OR_NIGHT',
                      'S4_IS_AGGRESSIVE_DRIVING',
                      'S4_IS_ALCOHOL_RELATED',
                      'S4_IS_CMV_INVOLVED',
                      'S4_IS_DISTRACTED',
                      'S4_IS_DRUG_RELATED',
                      'S4_IS_HIT_AND_RUN',
                      'S4_IS_INTERSECTION_RELATED',
                      'S4_IS_LANE_DEPARTURE_RELATED',
                      'S4_IS_SPEEDING_RELATED',
                      'S4_TRAILER_COUNT',
                      'S4_MOTORCYCLE_COUNT',
                      'S4_MOPED_COUNT',
                      'S4_BICYCLIST_COUNT',
                      'S4_AGING_DRIVER_COUNT',
                      'S4_TEENAGER_DRIVER_COUNT',
                      'S4_UNRESTRAINED_COUNT'
                      )
dataCrash = data %>%
  select(selected_columns) %>%
  mutate(across(c('RURAL_OR_URBAN',
                  'ROAD_SYSTEM_IDENTIFER',
                  'ROAD_SURFACE_CONDITION',
                  'S4_CRASH_TYPE_SIMPLIFIED',
                  'S4_DAY_OR_NIGHT',
                  'S4_IS_AGGRESSIVE_DRIVING',
                  'S4_IS_ALCOHOL_RELATED',
                  'S4_IS_CMV_INVOLVED',
                  'S4_IS_DISTRACTED',
                  'S4_IS_DRUG_RELATED',
                  'S4_IS_HIT_AND_RUN',
                  'S4_IS_INTERSECTION_RELATED',
                  'S4_IS_LANE_DEPARTURE_RELATED',
                  'S4_IS_SPEEDING_RELATED'), as.factor))



head(dataCrash)
```

```r
# Crash type plots
ggplot(dataCrash, aes(x = S4_CRASH_TYPE_SIMPLIFIED)) +
  geom_bar(aes(fill = factor(INCAPACITATING_FLAG, levels = c(0, 1), labels = c("No", "Yes")))) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "red")) +
  labs(fill = "Incapacitating Event", x = "Crash Type", y = NULL) +
  ggtitle("Crash Severity by Type") +
```

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(subset(dataCrash, INCAPACITATING_FLAG == 1), aes(x = S4_CRASH_TYPE_SIMPLIFIED)) +
  geom_bar(fill = "red") +
  labs(x = "Crash Type", y = NULL) +
  ggtitle("Incapacitating Events by Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Peak plots
ggplot(dataCrash, aes(x = PEAK)) +
  geom_bar(aes(fill = factor(INCAPACITATING_FLAG, levels = c(0, 1), labels = c("No", "Yes")))) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "red")) +
  labs(fill = "Incapacitating Event", x = "Period", y = NULL) +
  ggtitle("Crash Severity by Peak Period") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(subset(dataCrash, INCAPACITATING_FLAG == 1), aes(x = PEAK)) +
  geom_bar(fill = "red") +
  labs(x = "Period", y = NULL) +
  ggtitle("Incapacitating Events by Peak Period") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Roadway plots
ggplot(dataCrash, aes(x = ROAD_SYSTEM_IDENTIFER)) +
  geom_bar(aes(fill = factor(INCAPACITATING_FLAG, levels = c(0, 1), labels = c("No", "Yes")))) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "red")) +
  labs(fill = "Incapacitating Event", x = "Roadway", y = NULL) +
  ggtitle("Crashes by Roadway") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(subset(dataCrash, INCAPACITATING_FLAG == 1), aes(x = ROAD_SYSTEM_IDENTIFER)) +
  geom_bar(fill = "red") +
  labs(x = "Roadway", y = NULL) +
  ggtitle("Incapacitating Crashes by Roadway") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Split the data into train and test data by an 90/10 split and re-sample data to balance the response

```
library(ROSE)
set.seed(123)
train_ind <- sample(seq_len(nrow(dataCrash)), size = floor(nrow(dataCrash)*0.9))

crashtrain = dataCrash[train_ind,]
crashtest = dataCrash[-train_ind,]

crashtrain_balanced = ROSE(INCAPACITATING_FLAG~., crashtrain, N=2000)$data
summary(as.factor(crashtrain_balanced$INCAPACITATING_FLAG))
```

Logistic regression - full

```
modellogit = glm(INCAPACITATING_FLAG~., crashtrain_balanced, family='binomial'(link='logit'))
summary(modellogit)
```

```
predlog = ifelse(predict(modellogit, crashtest[,-1], type="response" ) < 0.5, 0, 1)

clog = confusionMatrix(as.factor(predlog), as.factor(crashtest$INCAPACITATING_FLAG))
clog

paste0('Logit stepwise testing error: ', 1-clog$overall['Accuracy'])
```

Logistic regression - stepwise with k=2 penalty degrees of freedom (AIC)

```
logstep = step(modellogit, trace = F)
summary(logstep)
stepformula = logstep[['formula']]

predlogstep = ifelse(predict(logstep, crashtest[,-1], type="response" ) < 0.5, 0, 1)

clogstep = confusionMatrix(as.factor(predlogstep), as.factor(crashtest$INCAPACITATING_FLAG))
clogstep

paste0('Logit stepwise testing error: ', 1-clogstep$overall['Accuracy'])
```

Random Forest

```
library(randomForest)
modelrf = randomForest(as.factor(INCAPACITATING_FLAG)~., crashtrain_balanced, proximity=T, importance=T
predrf <- predict(modelrf, crashtest)
crf = confusionMatrix(predrf, as.factor(crashtest$INCAPACITATING_FLAG))
crf

paste0('Random Forest Test Error: ', 1-crf$overall['Accuracy'])
```

Random Forest - stepwise

```
modelrfstep = randomForest(as.factor(INCAPACITATING_FLAG)~TOTAL_NUMBER_OF_VEHICLES +
    TOTAL_NUMBER_OF_PERSONS + RURAL_OR_URBAN + ROAD_SYSTEM_IDENTIFER +
    S4_CRASH_TYPE_SIMPLIFIED + S4_IS_AGGRESSIVE_DRIVING + S4_IS_ALCOHOL_RELATED +
    S4_IS_DRUG_RELATED + S4_IS_HIT_AND_RUN + S4_IS_INTERSECTION_RELATED +
    S4_IS_LANE_DEPARTURE_RELATED + S4_MOTORCYCLE_COUNT + S4_MOPED_COUNT +
    S4_AGING_DRIVER_COUNT + S4_UNRESTRAINED_COUNT, crashtrain_balanced, proximity=T, importance=T)
predrfstep <- predict(modelrfstep, crashtest)
crfstep = confusionMatrix(predrfstep, as.factor(crashtest$INCAPACITATING_FLAG))
crfstep

paste0('Random Forest - Stepwise Test Error: ', 1-crfstep$overall['Accuracy'])
```

Boosting

```
library(gbm)

modelgbm = gbm(INCAPACITATING_FLAG~., data=crashtrain_balanced, distribution = 'bernoulli', n.trees = 5
summary(modelgbm)
```

```r
## Find the estimated optimal number of iterations shown in a dashed blue line and store as a variable
perf_gbm = gbm.perf(modelgbm, method="cv")

## prediction
gbmpred = ifelse(predict(modelgbm,newdata = crashtest, n.trees=perf_gbm, type="response") < 0.5, 0, 1)
cgbm = confusionMatrix(as.factor(gbmpred), as.factor(crashtest$INCAPACITATING_FLAG))
cgbm

paste0('GBM Test Error: ', 1-cgbm$overall['Accuracy'])
```

Boosting - stepwise

```r
modelgbmstep = gbm(INCAPACITATING_FLAG~TOTAL_NUMBER_OF_VEHICLES +
    TOTAL_NUMBER_OF_PERSONS + RURAL_OR_URBAN + ROAD_SYSTEM_IDENTIFER +
    S4_CRASH_TYPE_SIMPLIFIED + S4_IS_AGGRESSIVE_DRIVING + S4_IS_ALCOHOL_RELATED +
    S4_IS_DRUG_RELATED + S4_IS_HIT_AND_RUN + S4_IS_INTERSECTION_RELATED +
    S4_IS_LANE_DEPARTURE_RELATED + S4_MOTORCYCLE_COUNT + S4_MOPED_COUNT +
    S4_AGING_DRIVER_COUNT + S4_UNRESTRAINED_COUNT, data=crashtrain_balanced, distribution = 'bernoulli'
summary(modelgbmstep)

## Find the estimated optimal number of iterations shown in a dashed blue line and store as a variable
perf_gbmstep = gbm.perf(modelgbmstep, method="cv")

## prediction
gbmpredstep = ifelse(predict(modelgbmstep,newdata = crashtest, n.trees=perf_gbmstep, type="response") <
cgbmstep = confusionMatrix(as.factor(gbmpredstep), as.factor(crashtest$INCAPACITATING_FLAG))
cgbmstep

paste0('GBM - Stepwise Test Error: ', 1-cgbmstep$overall['Accuracy'])
```