

# Capstone Project: Texas Traffic Data

Stiina Kilk

1/5/2021

## Overview

The purpose of this report is to introduce the reader to the machine learning project completed within the course “HarvardX Professional Certificate in Data Science”. The data set consists of information about traffic accidents, which occurred in Texas from February 2016 to June 2020. The full dataset, which covers 49 states can be found here: <https://arxiv.org/abs/1906.05409>. The original database has 49 variables ranging from coordinates of the accident to near-by traffic signs and weather condition. Each row contains information about a car accident, which has been evaluated by a level of severity. The goal of this project is to predict the severity of a car accident based on the variables. Most of the variables can be monitored in real time. First part of the report is the data cleaning. After that an analysis and visual exploration of the data is carried out along with explanations. Then the removal of unuseful variables is needed to take place before getting started with model building. Four machine learning algorithms are used in order to achieve the goal along with explanations. Finally, the author will conclude the report with a brief summary.

## Data cleaning

### Loading neccessary packages

```
#downloading and loading necessary packages

if (!require(tidyverse)) install.packages('tidyverse')
library(tidyverse)
if (!require(caret)) install.packages('caret')
library(caret)
if (!require(data.table)) install.packages('data.table')
library(data.table)
if (!require(dslabs)) install.packages('dslabs')
library(dslabs)
if (!require(ggplot2)) install.packages('ggplot2')
library(ggplot2)
if (!require(dplyr)) install.packages('dplyr')
library(dplyr)
if (!require(stringr)) install.packages('stringr')
library(stringr)
if (!require(knitr)) install.packages('knitr')
library(knitr)
if (!require(readr)) install.packages('readr')
library(readr)
if (!require(grDevices)) install.packages('grDevices')
library(grDevices)
if (!require(stats)) install.packages('stats')
```

```

library(stats)
if (!require(lubridate)) install.packages('lubridate')
library(lubridate)
if (!require(corrplot)) install.packages('corrplot')
library(corrplot)
if (!require(matrixStats)) install.packages('matrixStats')
library(matrixStats)
if (!require(ModelMetrics)) install.packages('ModelMetrics')
library(ModelMetrics)
if (!require(xgboost)) install.packages('xgboost')
library(xgboost)

```

## Downloading dataset

The original dataset is filtered to include only the observations about the state of Texas.

```

#filtered to only include accidents in TX

#download required packages
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
if(!require(utils)) install.packages("utils", repos = "http://cran.us.r-project.org")

#download zip file from author's github repo and unzip file
temp <- tempfile()
download.file("https://github.com/silk-dat/us_traffic_data/raw/main/TX_car_accidents.csv.zip",temp)
my_data <- read_csv(unz(temp, "TX_car_accidents.csv"))
unlink(temp)
remove(temp)

```

## Data cleaning

Data class needs to be changed from tibble to data frame to make it easier to work with.

```
my_data <- as.data.frame(my_data)
```

The symbols included in the names of the columns make it hard to use them in the code. Column names are changed in the next step.

```
colnames(my_data)
```

## [1] "X1"	"ID"	"Source"
## [4] "TMC"	"Severity"	"Start_Time"
## [7] "End_Time"	"Start_Lat"	"Start_Lng"
## [10] "End_Lat"	"End_Lng"	"Distance(mi)"
## [13] "Description"	"Number"	"Street"
## [16] "Side"	"City"	"County"
## [19] "State"	"Zipcode"	"Country"
## [22] "Timezone"	"Airport_Code"	"Weather_Timestamp"
## [25] "Temperature(F)"	"Wind_Chill(F)"	"Humidity(%)"
## [28] "Pressure(in)"	"Visibility(mi)"	"Wind_Direction"
## [31] "Wind_Speed(mph)"	"Precipitation(in)"	"Weather_Condition"
## [34] "Amenity"	"Bump"	"Crossing"

```

## [37] "Give_Way"                "Junction"                 "No_Exit"
## [40] "Railway"                  "Roundabout"                "Station"
## [43] "Stop"                     "Traffic_Calming"          "Traffic_Signal"
## [46] "Turning_Loop"              "Sunrise_Sunset"            "Civil_Twilight"
## [49] "Nautical_Twilight"         "Astronomical_Twilight"

names(my_data)[names(my_data) == "Temperature(F)"] <- "Temperature"
names(my_data)[names(my_data) == "Humidity(%)"] <- "Humidity"
names(my_data)[names(my_data) == "Pressure(in)"] <- "Pressure"
names(my_data)[names(my_data) == "Visibility(mi)"] <- "Visibility"

```

Here the relevant variables are chosen. The chosen variables are all available to monitor in real time. Variables naming the city and county are removed since coordinates and the zipcode of the accidents are included. There are many variables specifying the weather conditions from which the most important ones are preserved. All the variables describing traffic signs and road conditions are chosen.

```

my_data <- my_data %>% select(Start_Time, Start_Lat, Start_Lng, Zipcode,
                                Amenity, Bump, Crossing, Give_Way, Junction,
                                Roundabout, Stop, Severity, Traffic_Signal)

```

Here we see which variables need to be formatted for the machine learning algorithms.

```

##   Start_Time           Start_Lat        Start_Lng
##   Min.   :2016-06-14 20:06:43  Min.   :25.92  Min.   :-106.59
##  1st Qu.:2017-05-02 22:52:53  1st Qu.:29.75  1st Qu.:-97.72
##  Median :2018-05-03 08:24:12  Median :30.26  Median :-96.82
##  Mean   :2018-05-18 15:59:12  Mean   :30.90  Mean   :-97.04
##  3rd Qu.:2019-05-12 18:17:00  3rd Qu.:32.72  3rd Qu.:-95.49
##  Max.   :2020-06-30 22:56:52  Max.   :36.38  Max.   :-93.72
##
##   Zipcode           Temperature      Visibility       Amenity
##   Length:329284    Min.   :-40.00  Min.   : 0.000  Mode :logical
##   Class :character 1st Qu.: 60.10  1st Qu.: 10.000 FALSE:323282
##   Mode  :character Median : 73.00  Median : 10.000 TRUE :6002
##                           Mean   : 70.89  Mean   : 9.167
##                           3rd Qu.: 82.40  3rd Qu.: 10.000
##                           Max.   :161.60  Max.   :111.000
##                           NA's   :6047   NA's   :6707
##   Bump             Crossing       Give_Way        Junction
##   Mode :logical    Mode :logical  Mode :logical  Mode :logical
##   FALSE:329226     FALSE:303825   FALSE:327296   FALSE:311379
##   TRUE :58         TRUE :25459    TRUE :1988     TRUE :17905
##
##   No_Exit          Railway        Roundabout      Stop
##   Mode :logical    Mode :logical  Mode :logical  Mode :logical
##   FALSE:328906     FALSE:326208   FALSE:329272   FALSE:323425
##   TRUE :378        TRUE :3076    TRUE :12       TRUE :5859
##
##   
```

```

##          Severity      Traffic_Signal
##  Min.    :1.000   Mode :logical
##  1st Qu.:2.000   FALSE:234588
##  Median  :2.000   TRUE :94696
##  Mean    :2.298
##  3rd Qu.:3.000
##  Max.    :4.000
##

```

Rows with missing values will be removed. In the next step, 2,2% of the data is removed due to missing values. First row shows the number of observations with missing values and the second row is without the missing values.

```
## [1] 329284
```

```
## [1] 322205
```

Zipcodes may have an added 4 digits to specify the location, which will need to be removed to format this variable. Summary showed that this variable is a character type and needs to be changed into numeric type.

```
my_data$Zipcode <- gsub(my_data$Zipcode, pattern="-.*", replacement = "")  
all(str_length(my_data$Zipcode) == 5)
```

```
## [1] TRUE
```

```
my_data$Zipcode <- as.numeric(my_data$Zipcode)
```

Start\_Time column in it's current form is not very descriptive. In the next step this column will be divided into 4 further columns: Month, Week, Day and Hour.

```
my_data <- my_data %>% mutate(Month = month(as.Date(Start_Time)))  
my_data <- my_data %>% mutate(Week = week(as.Date(Start_Time)))  
my_data <- my_data %>% mutate(Day = day(as.Date(Start_Time)))  
my_data <- my_data %>% mutate(Hour = hour(my_data$Start_Time))  
  
my_data <- my_data[,-1]
```

According to the summary the maximum temperature is 161.60 F and minimum is -40 F. Neither of these temperatures are possible in Texas. The highest temperature in Texas during the period included in the data is 112 F. Any temperatures higher than that will be removed. 25 rows are removed.

```
#check the unique values  
unique(my_data$Temperature)
```

```

## [1] 66.0 62.1 60.1 61.0 59.0 66.2 66.9 64.4 57.0 57.2 63.0 62.6  
## [13] 64.0 57.9 51.8 55.4 52.0 60.8 54.0 53.6 52.2 46.9 50.0 55.0  
## [25] 53.1 48.9 43.0 39.7 37.9 46.4 46.0 43.2 48.2 48.0 34.0 44.1  
## [37] 45.0 37.6 38.3 42.1 41.0 37.0 39.9 44.6 55.9 51.1 59.7 62.2  
## [49] 64.9 68.0 70.0 55.6 56.5 48.7 41.4 48.6 39.2 61.9 63.5 56.3  
## [61] 54.3 57.7 52.7 56.1 55.8 54.9 50.9 49.6 46.8 47.3 49.5 49.3

```

```

## [73] 48.4 51.6 71.6 69.1 69.8 71.1 53.2 47.7 38.8 39.0 42.3 45.7
## [85] 46.6 49.8 54.7 42.8 39.4 35.1 37.4 33.1 33.8 35.6 36.0 35.4
## [97] 34.9 32.0 33.3 32.9 30.9 31.6 27.0 28.4 30.0 30.7 32.5 34.2
## [109] 35.8 40.8 47.5 66.7 73.0 73.4 75.0 75.2 73.9 74.5 72.0 68.9
## [121] 65.1 65.3 66.4 72.9 56.7 50.4 68.4 75.9 74.3 77.0 76.8 79.0
## [133] 78.1 45.5 45.9 44.2 49.1 43.3 44.4 51.3 60.3 67.3 68.7 59.9
## [145] 61.2 63.9 63.1 67.1 78.8 80.1 80.6 78.3 73.2 79.2 76.3 81.0
## [157] 28.0 19.0 19.4 26.6 29.8 21.2 23.0 25.0 25.5 24.1 18.0 16.0
## [169] 17.1 15.3 17.6 15.8 23.5 21.0 19.9 35.2 30.2 29.3 28.9 31.1
## [181] 39.6 69.3 67.8 60.6 65.5 57.4 47.8 54.5 68.2 61.3 43.9 78.6
## [193] 53.8 64.6 52.9 75.7 84.2 70.7 61.7 82.4 72.5 82.0 84.0 82.9
## [205] 81.7 69.4 64.8 37.8 38.5 50.2 62.8 71.2 52.3 58.1 58.5 45.1
## [217] 40.6 28.2 26.1 24.8 23.7 28.8 21.9 25.2 22.6 22.5 21.7 18.9
## [229] 18.5 20.7 20.3 17.4 32.4 79.9 71.8 64.2 74.1 72.1 70.9 70.3
## [241] 70.2 67.5 43.5 41.7 76.6 41.2 40.1 72.3 55.2 54.1 59.4 58.6
## [253] 63.7 43.7 61.5 62.4 52.5 59.2 74.8 65.8 65.7 44.8 63.3 86.0
## [265] 143.6 136.4 87.8 67.6 79.5 84.9 98.6 91.4 109.4 82.8 70.5 87.1
## [277] 77.4 77.2 75.6 81.9 82.2 76.1 76.5 82.6 89.1 84.6 90.0 88.5
## [289] 87.3 88.0 83.3 74.7 83.7 81.1 73.6 77.7 71.4 73.8 56.8 60.4
## [301] 59.5 58.8 57.6 75.4 77.9 79.7 80.2 77.5 72.7 66.6 58.3 68.5
## [313] 45.3 69.6 81.5 84.4 83.8 89.6 93.2 91.0 91.2 90.7 93.0 86.9
## [325] 93.9 91.9 85.6 80.8 89.4 90.9 81.3 87.4 88.9 98.1 78.4 95.0
## [337] 92.1 91.6 88.7 90.5 84.7 96.1 97.0 96.8 99.0 100.0 85.1 94.3
## [349] 91.8 87.6 86.2 90.1 86.4 83.1 85.5 107.6 85.8 104.0 100.4 86.5
## [361] 116.6 122.0 118.4 102.2 88.2 105.8 114.8 132.8 83.5 89.8 90.3 86.7
## [373] 161.6 140.0 111.2 96.6 97.7 95.2 100.9 97.2 92.3 95.5 92.5 96.3
## [385] 95.4 102.9 102.0 85.3 98.4 88.3 97.9 102.6 98.8 99.7 99.5 103.3
## [397] 96.4 94.1 92.8 79.3 94.6 96.0 92.0 87.0 93.4 94.0 83.0 89.0
## [409] 93.6 98.0 93.7 80.0 76.0 74.0 92.7 85.0 94.8 95.9 89.2 95.7
## [421] 99.9 40.5 30.4 53.4 47.1 41.9 51.4 32.2 50.5 42.4 80.4 46.2
## [433] 94.5 100.2 78.0 101.0 67.0 69.0 65.0 71.0 62.0 58.0 60.0 56.0
## [445] 53.0 51.0 47.0 49.0 42.0 40.0 44.0 26.0 38.0 33.0 35.0 29.0
## [457] 31.0 24.0 22.0 20.0 103.0 105.0 16.2 42.6 36.5 31.8 34.7 36.3
## [469] 36.7 22.1 23.4 22.8 41.5 27.1 33.4 38.7 23.9 32.7 33.6 37.2
## [481] 50.7 34.3 36.9 40.3 29.1 29.5 31.3 28.6 27.5 36.1 31.5 27.7
## [493] 26.4 30.6 129.2 38.1 97.3 97.5 98.2 99.3 100.6 99.1 101.7 101.8
## [505] 103.1 102.4 100.8 104.4 105.1 106.0 105.4 107.1 101.3 108.0 109.0 110.1
## [517] 101.1 110.3 109.9 107.4 103.8 101.5 29.7 27.9 25.7 25.3 24.6 14.0
## [529] 26.2 24.3 19.8 12.9 15.6 11.3 15.1 12.2 14.9 18.3 20.1 16.9
## [541] 17.8 16.5 17.2 127.4 25.9 105.3 104.2 -40.0

```

```

#how many rows will need to be removed.
my_data %>% filter(Temperature > 112) %>% nrow()

```

```

## [1] 25

```

```

#remove the rows
r <- with(my_data, which(Temperature > 112, arr.ind = TRUE))
my_data <- my_data[-r,]

```

As is visible above, rows showing temperature of -40 F have to be removed. Only one row includes this temperature and is removed in the next step.

```

my_data %>% filter(Temperature == -40) %>% nrow()

## [1] 1

#remove the rows
r <- with(my_data, which(Temperature == -40, arr.ind = TRUE))
my_data <- my_data[-r,]

```

Visibility of 111 mi is illogical. Dust, vapour and pollution in the air will rarely let you see more than 13mi. All readings below that point are removed.

```

summary(my_data$Visibility)

##      Min.   1st Qu.    Median     Mean   3rd Qu.    Max.
##      0.000  10.000  10.000   9.168  10.000 111.000

```

```

#how many rows will be removed
my_data %>% filter(Visibility > 13) %>% nrow()

```

```
## [1] 135
```

```

#remove rows
r <- with(my_data, which(Visibility > 13, arr.ind = TRUE))
my_data <- my_data[-r,]

```

## Data Exploration

The data consist of 317 381 observations or rows and have 21 variables. The variables are in different formats: numerical, character and logical. Each row describes a car accident and has been evaluated with a severity from 1 to 4, where 1 indicates the least impact to traffic (a short delay as a result of the traffic) and 4 indicates a significant impact on the traffic (long delay).

```

##   Severity Start_Lat Start_Lng Zipcode Temperature Visibility Amenity Bump
## 1        2  30.33650 -97.75565    78731       66.0        10 FALSE FALSE
## 2        2  30.32816 -97.69430    78752       62.1        10 FALSE FALSE
## 3        2  30.32608 -97.69231    78752       62.1        10 FALSE FALSE
## 4        2  32.66219 -96.94315    75249       60.1        10 FALSE FALSE
## 5        3  32.77879 -96.78202    75226       61.0        10 FALSE FALSE
## 6        2  32.72428 -96.76224    75215       61.0        10 FALSE FALSE
##   Crossing Give_Way Junction No_Exit Railway Roundabout Stop Traffic_Signal
## 1 FALSE FALSE
## 2 FALSE FALSE
## 3 FALSE TRUE
## 4 FALSE FALSE
## 5 FALSE FALSE
## 6 FALSE FALSE
##   Month Week Day Hour
## 1     11   48   30   16
## 2     11   48   30   16
## 3     11   48   30   16

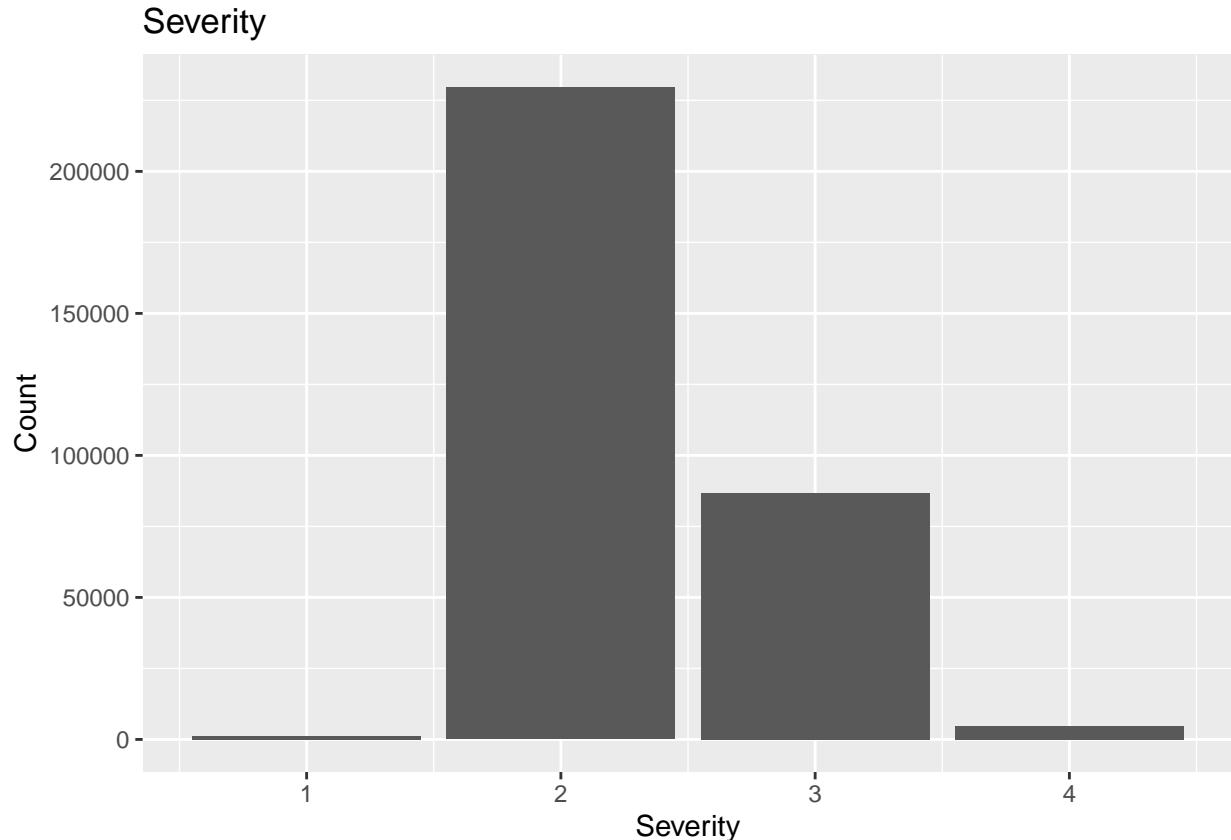
```

```
## 4     11    48   30   16
## 5     11    48   30   16
## 6     11    48   30   16
```

```
## [1] 322044
```

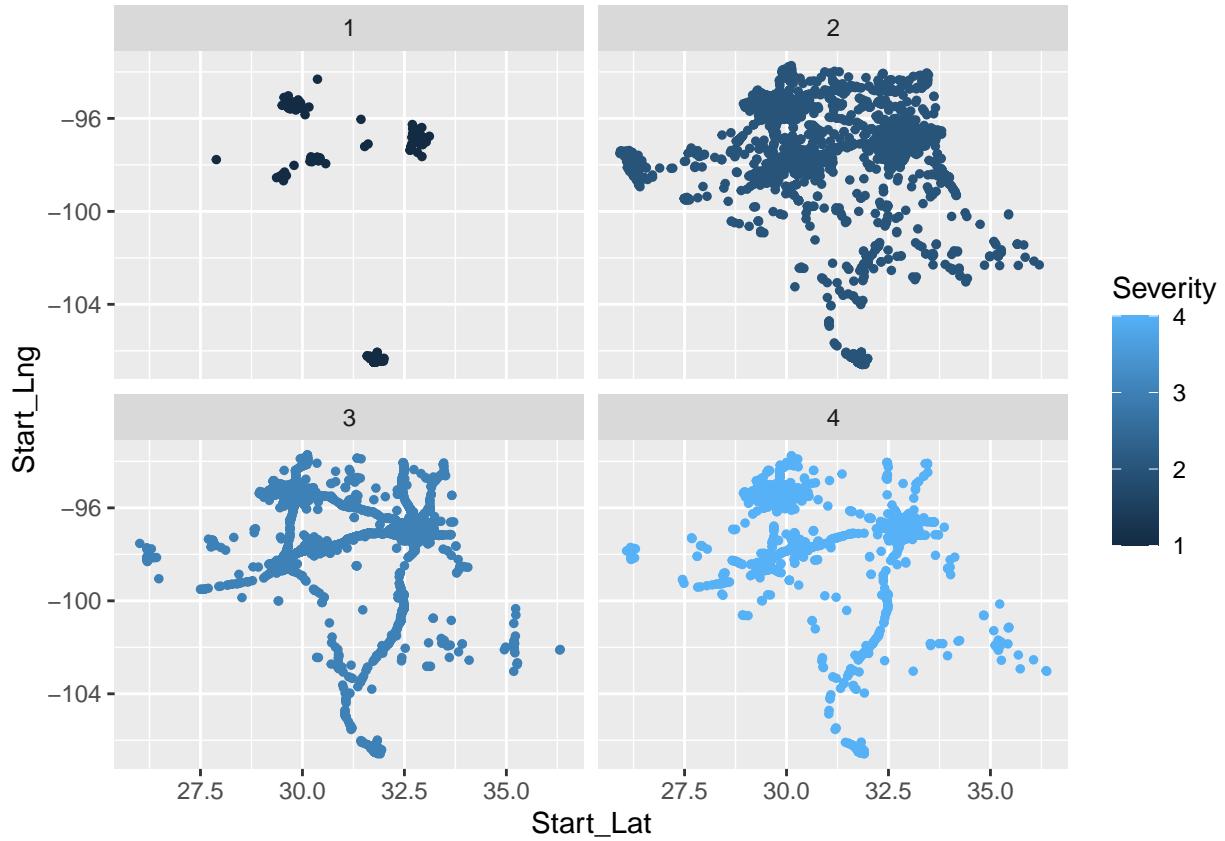
```
## [1] 20
```

The distribution plot shows that most of the accidents have a level 2 longevity. Accidents creating the least amount of chaos are also the fewest according to the plot.



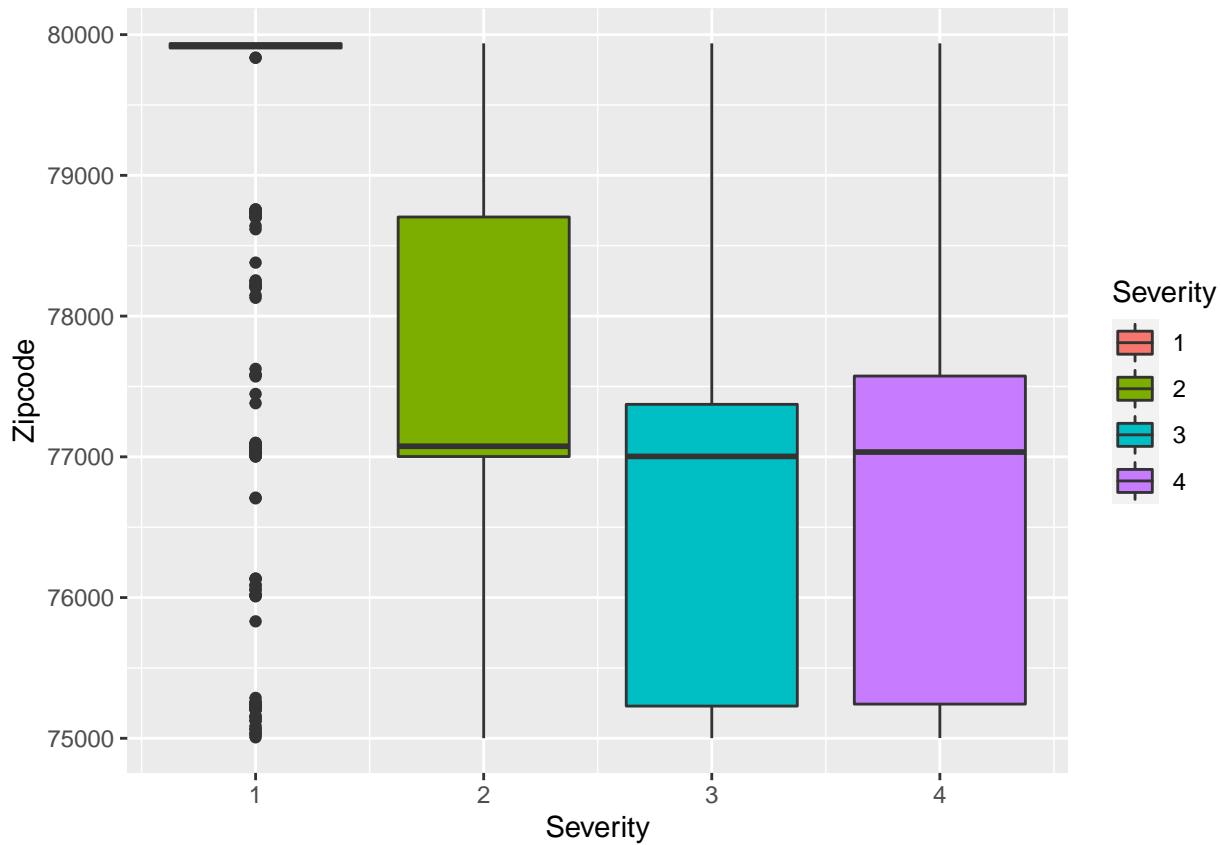
Latitude and longitude seem to be an indication for the level of severity. As expected from the distribution of severity, level 1 takes up the least space within the coordinates, but the locations of the accidents are also clustered around certain points. The areas for levels 3 and 4 are quite similar, while level 2 shows a more diffused picture.

```
my_data %>%
  ggplot(aes(Start_Lat, Start_Lng, color = Severity)) +
  geom_point(size = 1) +
  facet_wrap(~ Severity, nrow = 2)
```



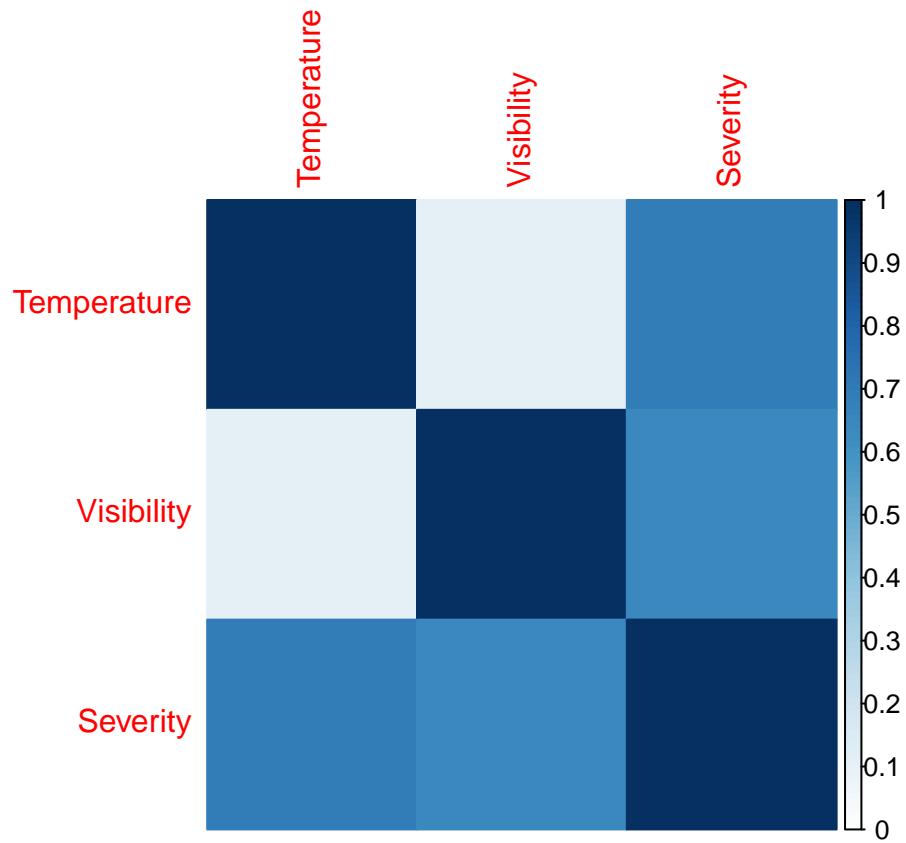
Zipcode is another location based variable, but provides a wider view. Level 1 severity has many outliers, which form some kind of clusters confirming the conclusion drawn from the above plot. Level 2 severity shows more variability than 3 and 4, but the most common zipcodes for all three are not far apart.

```
my_data %>%
  ggplot(aes(Severity, Zipcode)) +
  geom_boxplot(aes(fill = factor(Severity, levels = c(1, 2, 3, 4)))) +
  labs(fill = "Severity")
```



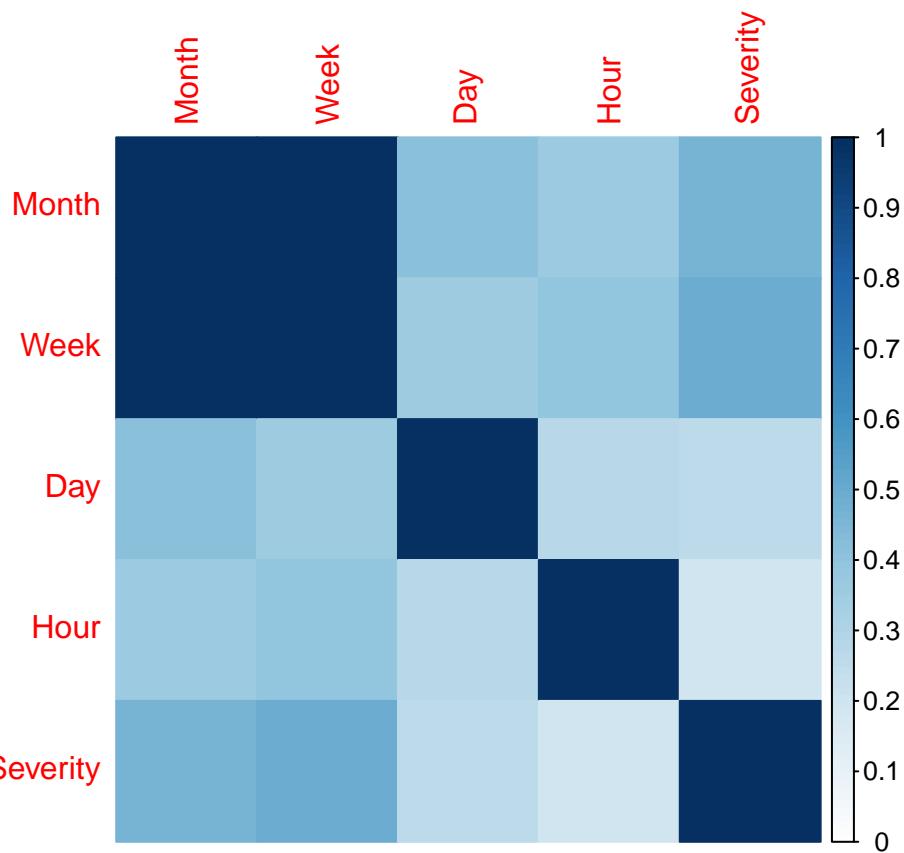
According to the correlation matrix, temperature at the time of the accident influences the severity more than visibility. Weather in Texas can be as hot as literal hell in the summer months and this can have an impact on the driver's alertness while driving. Also, it is uncommon to have poor visibility as cold weather conditions such as snow and ice are unusual.

```
#select wather variables
weather <- my_data %>% select(Temperature, Visibility, Severity)
#create correlation matrix
weather_cor <- cor(weather)
#display matrix
corrplot(abs(cor(weather_cor)), method="color", tl.pos="lt", cl.lim = c(0,1))
```



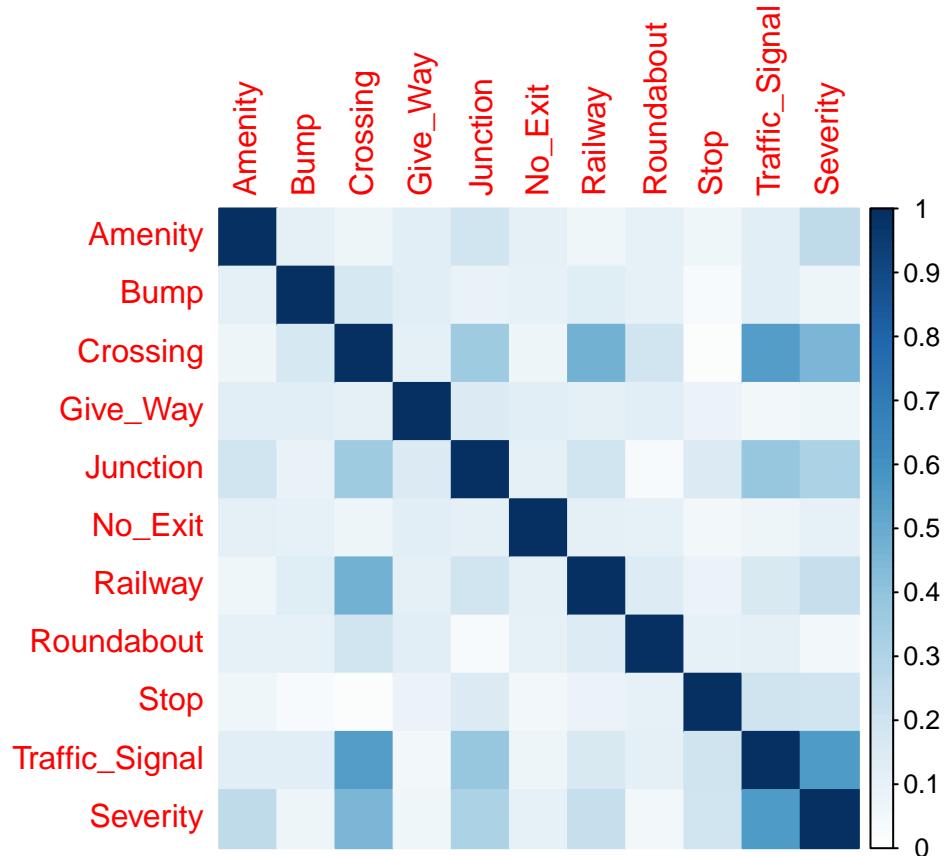
Week variable seems to have the most impact on the severity of the accident.

```
#select variables
datetime <- my_data %>% select(Month, Week, Day, Hour, Severity)
#create correlation matrix
datetime_cor <- cor(datetime)
#display matrix
corrplot(abs(cor(datetime_cor)), method="color", tl.pos="lt", cl.lim = c(0,1))
```



The presence of a traffic signal or traffic light at the vicinity of the accident correlates most with the severity of the accident.

```
#select variables
road <- my_data %>% select(Amenity, Bump, Crossing, Give_Way, Junction,
                               No_Exit, Railway, Roundabout, Stop, Traffic_Signal, Severity)
#create correlation matrix
road_cor <- cor(road)
#display matrix
corrplot(abs(cor(road_cor)), method="color", tl.pos="lt", cl.lim = c(0,1))
```



## Modelling

### Data splitting

Dataset is split into two parts: training set and test set. Models are trained on the training set to find the best prediction and this model is then run on test set. This is done to evaluate the model with data that was not used for training the model. Since the dataset is fairly big, splits 70/30 and 80/20 were tried on the models and 70/30 gave the best results.

```
#reproducible data
set.seed(1, sample.kind="Rounding")

#split the data
test_index <- createDataPartition(factor(my_data$Severity, levels = c(1, 2, 3, 4)), times = 1, p = 0.7,
train_set <- my_data[test_index, ]
test_set <- my_data[-test_index, ]

#separate features and label for preprocessing
x_train <- train_set[,-1]
y_train <- factor(train_set[,1])

x_test <- test_set[,-1]
y_test <- factor(test_set[,1])
```

### Preprocessing

Preprocessing is a step in machine learning, which directly influences the outcome of the model and reduces computation time. Here the standard deviation of each variable is computed and variables with zero or near-zero variance or standard deviation are excluded. Using these variables in models would create no benefit and are not descriptive of the value to be predicted.

```
#create a matrix of the data
x <- as.matrix(x_train)

#calculates the standard deviation for each column
sds <- colSds(x)

#calculate the variables, which have zero or near-zero variability
nzv <- nearZeroVar(x)
```

These are the variables excluded going forward.

```
#show the column names of zero or near-zero variability
colnames(as.data.frame(x)[nzv])

## [1] "Visibility" "Amenity"     "Bump"        "Give_Way"    "No_Exit"
## [6] "Railway"    "Roundabout"  "Stop"
```

These are the useful variables and are included in modelling.

```
#show the column names of the columns we will work with
col_index <- setdiff(1:ncol(x), nzv)
colnames(as.data.frame(x)[col_index])

## [1] "Start_Lat"      "Start_Lng"      "Zipcode"       "Temperature"
## [5] "Crossing"       "Junction"      "Traffic_Signal" "Month"
## [9] "Week"           "Day"           "Hour"

#change x_train and x_test to include only useful variables
x_train <- x_train[col_index]
x_test <- x_test[col_index]
```

## Evaluation Function

Loss function (RMSE) is used throughout the analysis in order to evaluate the performance of the model. The output of the function shows how much the prediction deviates from the actual result.

## Quadratic Discriminant Analysis

Firstly, this algorithm identifies the distribution of each variable for each level of severity. Then it flips the distribution so that it is possible to calculate the level of severity for each row of the observation. The result of this algorithm is 0.594, which means the prediction deviates from the actual result by ca 0.6 points.

```
#train with qda model
fit_qda <- train(x_train, y_train, method = "qda")
#predict with model
y_hat <- predict(fit_qda, x_test)
#show accuracy
qda_rmse <- rmse(y_test, y_hat)
qda_rmse
```

```

## [1] 0.5943467

#create a table for storing results
rmse_results <- tibble(method = "QDA", RMSE = qda_rmse)

```

## Decision Tree

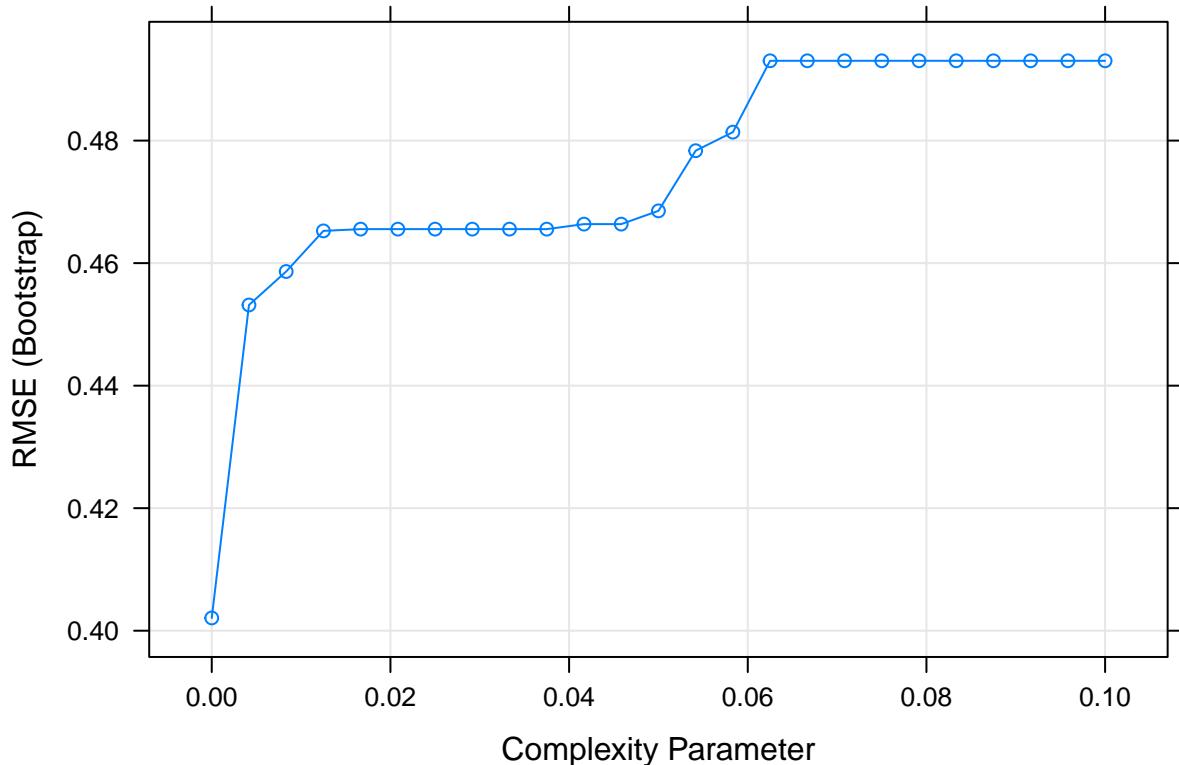
This algorithm partitions the data into regions with one variable at a time with nodes at the end of them that represent decision points. The data is split until the final node, which gives the prediction.

Cross-validation is used to choose the complexity parameter. The best value for a complecity parameter is 0.

```

fit_rpart <- train(x_train, as.numeric(y_train),
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)))
#plot rmse on each parameter
plot(fit_rpart)

```



```
fit_rpart$bestTune
```

```

##    cp
## 1  0

```

The decision tree algorithm gives the result of 0.377. This model performs better than QDA method.

```

#predict with model
y_hat <- predict(fit_rpart, x_test)
#show result
rpart_rmse <- rmse(y_test, y_hat)
rpart_rmse

## [1] 0.3767506

#add result to the table
rmse_results <- add_row(rmse_results, method = "RPART", RMSE = rpart_rmse)

```

## Decision Trees with Stochastic Gradient Boosting

In this method, the decision trees are grown sequentially: each successive tree is grown using information from previously grown trees, with the aim to minimize the error of the previous models. Cross-validation with 10 folds is used to find the best hyperparameters for the model.

```

set.seed(123, sample.kind = "Rounding")

xgb_fit <- train(x_test, as.numeric(y_test),
                  method = "xgbTree",
                  trControl = trainControl("cv", number = 10))

```

```

## [20:33:01] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:13] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:18] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:24] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:30] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:36] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:43] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:50] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:33:57] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:05] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:14] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:21] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:30] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:39] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:47] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:34:57] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:16] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:22] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:28] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:33] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:39] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:45] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:51] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:35:58] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:36:06] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:36:12] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [20:36:20] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated

```













```

## [21:16:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:16:14] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:16:23] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:16:32] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:16:40] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:16:50] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:17:00] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:17:10] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated

#show best tune
xgb_fit$bestTune

```

```

##      nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 108      150          3 0.4     0             0.8                 1               1

```

The result is 0.386, which is worse than the previous model.

```

#predict with model
y_hat <- predict(xgb_fit, x_test)
#show result
xgb_rmse <- rmse(y_test, y_hat)
xgb_rmse

## [1] 0.3860944

#add result to the table
rmse_results <- add_row(rmse_results, method = "XBG", RMSE = xgb_rmse)

```

## Generalized Linear Model with Gradient Descent and Regularization

This model builds generalized linear model and optimizes it usin regularization and gradient descent. Gradient descent tries to optimize the loss function by tuning different values of coefficients to minimize the error. In this algorithm, the subsequent models are built on residuals (actual - prediction) generated by previous examples. Firstly, the parameters are tuned with cross validation and then the model is fitted.

```

xgb_trcontrol = trainControl(
  method = "cv",
  number = 5,
  allowParallel = TRUE,
  verboseIter = FALSE,
  returnData = FALSE
)

xgb_l_fit <- train(x_test, as.numeric(y_test),
                     method = "xgbLinear",
                     trControl = xgb_trcontrol)

```

```

## [21:17:22] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:17:28] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:17:34] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:17:40] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated

```





```

## [21:38:11] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:38:16] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:38:22] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:38:28] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:38:33] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:38:39] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:38:50] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:39:01] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:39:12] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:39:23] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:39:34] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:39:45] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:39:57] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:40:08] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:40:18] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:40:35] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:40:51] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:41:23] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:41:40] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:41:56] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:42:12] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:42:28] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated
## [21:42:45] WARNING: amalgamation/../src/objective/regression_obj.cu:174: reg:linear is now deprecated

y_hat <- predict(xbg_l_fit, x_test)

xbg_l_rmse <- rmse(y_test, y_hat)
xbg_l_rmse

## [1] 0.3199985

rmse_results <- add_row(rmse_results, method = "XBG_L", RMSE = xbg_l_rmse)

```

The result is 0.32, which is the best performing model.

## Conclusion

In this report, 4 different models were fitted to find the best model.

```
rmse_results
```

```

## # A tibble: 4 x 2
##   method    RMSE
##   <chr>    <dbl>
## 1 QDA      0.594
## 2 RPART    0.377
## 3 XBG      0.386
## 4 XBG_L    0.320

```

The best model was found using Generalized Linear Model with Gradient Descent and Regularization. Decision Tree model and Boosted Decision Trees model could have performed better if with further tuned parameters to fit the best model. Also, Principal Component Analysis could be run with missing values to provide an estimates for them and therefore not lose any possible statistical characteristics of variables.

## Acknowledgements

US-Accidents: A Countrywide Traffic Accident Dataset [https://smoosavi.org/datasets/us\\_accidents](https://smoosavi.org/datasets/us_accidents)

Beginners Tutorial on XGBoost and Parameter Tuning in R <https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/>

Statistical Machine Learning Essentials <http://www.sthda.com/english/articles/35-statistical-machine-learning-essentials/139-gradient-boosting-essentials-in-r-using-xgboost/>