Accident Severity Prediction Model

Sarthak Pant

6/16/2021

Introduction

The goal of this project is to predict the worst accident severities based on a combination of variables in the United Kingdom by using the UK traffic collision dataset from Kaggle. There were three different datasets which had split up the data by years. The rbind function was used to create a cumulative dataset titled total_accidents_2005_to_2014. The dataset is made up of 1,504,150 accident reports from 2005 to 2014 (except for 2008) across 33 different variables. The prediction model will be built using cross validation and regularization. In order to determine the accuracy of the model, the residual mean squared error (RMSE) is calculated with the target of achieving a score below 0.45.

Loading Data

As previously mentioned, the datasets were sourced from "1.6 million UK traffic accidents" dataset on kaggle: https://www.kaggle.com/daveianhickey/2000-16-traffic-flow-england-scotland-wales. The complete accident data was split across three csv files: accidents_2005_to_2007.csv, accidents_2009_to_2011.csv, and accidents_2012_to_2014.csv.The three datasets were combined into one large dataset which consists of the accident information from 2005 to 2014 (excluding 2008)

```
# knitr::knit_global()
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(latexpdf)

accidents_05_to_07 <- read.csv(file = './CYO_DATASETS/accidents_2005_to_2007.csv')
accidents_09_to_11 <- read.csv(file = './CYO_DATASETS/accidents_2009_to_2011.csv')
accidents_12_to_14 <- read.csv(file = './CYO_DATASETS/accidents_2012_to_2014.csv')

total_accidents_2005_to_2014 <- rbind(accidents_05_to_07, accidents_09_to_11, accidents_12_to_14)
```

Data Preparation

Before building the algorithm, the data was explored and prepped. It was first split into two subsets with the CYO set consisting of 90% of the data and the Validation set consisting of the remaining 10% of the data. The purpose of splitting the data was to have one of the sets for training and the other for testing. The Validation set is only to be used when running the final model which is why the CYO set was further split into two subsets with the training set consisting of 80% of the data and the testing set consisting of the remaining 20% of the data.

```
#split the total_accidents_2005_to_2014 data set into a 90% CYO dataset and 10% VALIDATION dataset
set.seed(1)
test_index <- createDataPartition(y = total_accidents_2005_to_2014$Accident_Severity,
times = 1, p = 0.1, list = FALSE)
CYO <- total_accidents_2005_to_2014[-test_index,]
temp <- total_accidents_2005_to_2014[test_index,]
Validation <- temp
```

```
#Split the CYO dataset into an 80% training set and 20% testing set#
set.seed(1)
test_index <- createDataPartition(y = CYO$Accident_Severity , times = 1, p = 0.2, list = FALSE)
testing_set <- total_accidents_2005_to_2014[-test_index,]
training_set <- total_accidents_2005_to_2014[test_index,]
```

Data Exploration

In order to familiarize myself with the data I looked at the summary and the distribution of each of the variables. Out of the 33 different variables seven variables were used which included Light Conditions, Day of Week, Road Surface Conditions, Speed Limit, Road Type, Weather Conditions, and Urban or Rural.

```
#Data set summary
summary(CYO)
```

```
##  Accident_Index     Location_Easting_OSGR Location_Northing_OSGR
##  Length:1353735     Min.   : 64950        Min.   :  10290
##  Class :character   1st Qu.:375030        1st Qu.: 178250
##  Mode  :character   Median :439930        Median : 269030
##                     Mean   :439618        Mean   : 300189
##                     3rd Qu.:523050        3rd Qu.: 398190
```

```
##                    Max.   :655370        Max.   :1205100
##                    NA's   :94            NA's   :94
##    Longitude          Latitude      Police_Force  Accident_Severity
##  Min.   :-7.5162   Min.   :49.91   Min.   : 1.0   Min.   :1.000
##  1st Qu.:-2.3741   1st Qu.:51.49   1st Qu.: 6.0   1st Qu.:3.000
##  Median :-1.4039   Median :52.31   Median :30.0   Median :3.000
##  Mean   :-1.4367   Mean   :52.59   Mean   :30.2   Mean   :2.838
##  3rd Qu.:-0.2215   3rd Qu.:53.48   3rd Qu.:45.0   3rd Qu.:3.000
##  Max.   : 1.7594   Max.   :60.72   Max.   :98.0   Max.   :3.000
##  NA's   :94        NA's   :94
##  Number_of_Vehicles Number_of_Casualties     Date            Day_of_Week
##  Min.   : 1.000     Min.   : 1.000       Length:1353735     Min.   :1.000
##  1st Qu.: 1.000     1st Qu.: 1.000       Class :character   1st Qu.:2.000
##  Median : 2.000     Median : 1.000       Mode  :character   Median :4.000
##  Mean   : 1.832     Mean   : 1.351                          Mean   :4.118
##  3rd Qu.: 2.000     3rd Qu.: 1.000                          3rd Qu.:6.000
##  Max.   :67.000     Max.   :93.000                          Max.   :7.000
##
##      Time           Local_Authority_.District. Local_Authority_.Highway.
##  Length:1353735     Min.   :  1.0              Length:1353735
##  Class :character   1st Qu.:110.0              Class :character
##  Mode  :character   Median :322.0              Mode  :character
##                     Mean   :347.6
##                     3rd Qu.:518.0
##                     Max.   :941.0
##
##  X1st_Road_Class X1st_Road_Number  Road_Type          Speed_limit
##  Min.   :1.000   Min.   :  -1     Length:1353735     Min.   :10.00
##  1st Qu.:3.000   1st Qu.:   0     Class :character   1st Qu.:30.00
##  Median :4.000   Median : 129     Mode  :character   Median :30.00
##  Mean   :4.087   Mean   :1009                        Mean   :39.01
##  3rd Qu.:6.000   3rd Qu.: 724                        3rd Qu.:50.00
##  Max.   :6.000   Max.   :9999                        Max.   :70.00
##
##  Junction_Detail Junction_Control   X2nd_Road_Class  X2nd_Road_Number
##  Mode:logical    Length:1353735     Min.   :-1.000   Min.   :  -1
##  NA's:1353735    Class :character   1st Qu.:-1.000   1st Qu.:   0
##                  Mode  :character   Median : 3.000   Median :   0
##                                     Mean   : 2.674   Mean   : 381
##                                     3rd Qu.: 6.000   3rd Qu.:   0
##                                     Max.   : 6.000   Max.   :9999
##
##  Pedestrian_Crossing.Human_Control Pedestrian_Crossing.Physical_Facilities
##  Length:1353735                    Length:1353735
##  Class :character                  Class :character
##  Mode  :character                  Mode  :character
##
##
##
##
##  Light_Conditions   Weather_Conditions Road_Surface_Conditions
##  Length:1353735     Length:1353735     Length:1353735
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

```
##
##
##
##
##  Special_Conditions_at_Site Carriageway_Hazards Urban_or_Rural_Area
##  Length:1353735             Length:1353735      Min.   :1.000
##  Class :character           Class :character    1st Qu.:1.000
##  Mode  :character           Mode  :character    Median :1.000
##                                                 Mean   :1.354
##                                                 3rd Qu.:2.000
##                                                 Max.   :3.000
##
##  Did_Police_Officer_Attend_Scene_of_Accident LSOA_of_Accident_Location
##  Length:1353735                              Length:1353735
##  Class :character                            Class :character
##  Mode  :character                            Mode  :character
##
##
##
##
##       Year
##  Min.   :2005
##  1st Qu.:2006
##  Median :2010
##  Mean   :2009
##  3rd Qu.:2012
##  Max.   :2014
##
```

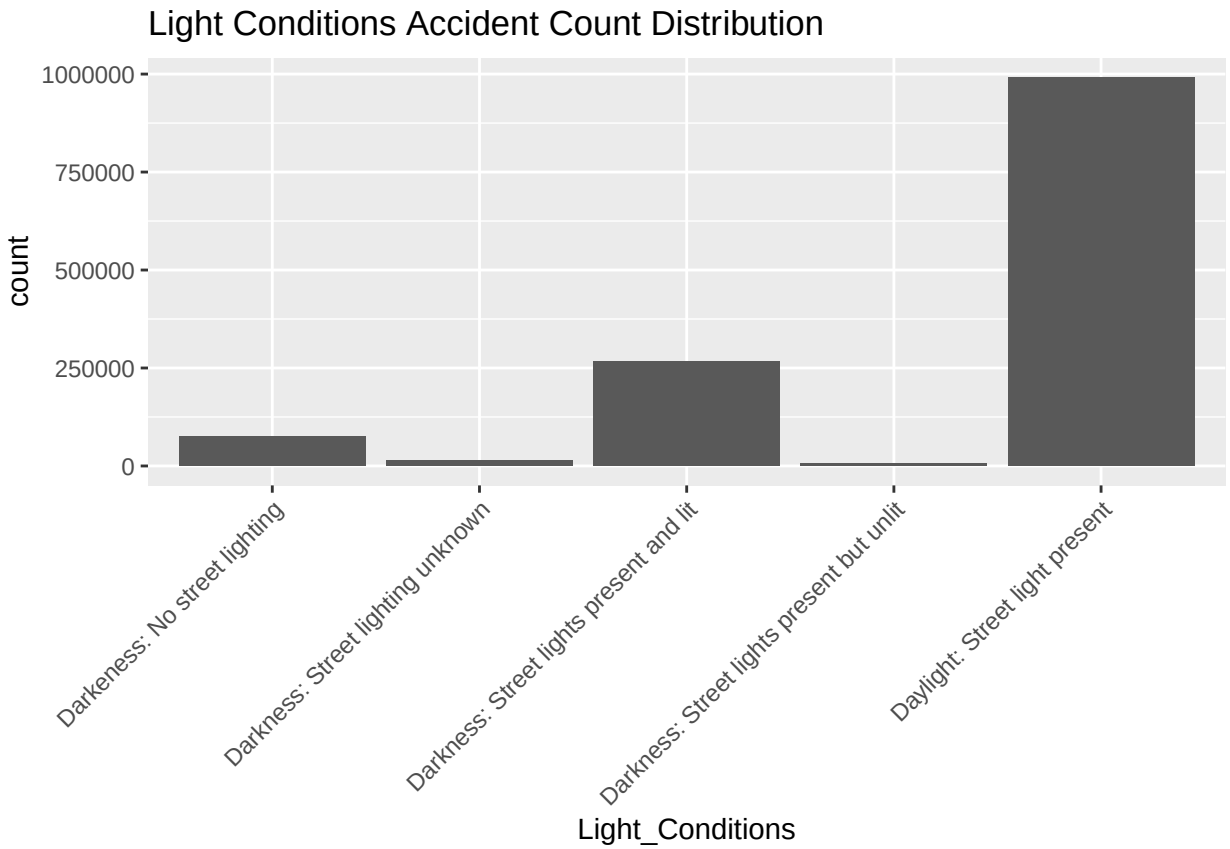
Summary of the variables

As mentioned above, we will be looking seven variables for the purpose of this model. In this section we are looking at the summary of each of the variables in order to understand the distribution of the data and see how it could potentially affect our model.

```
#Light Conditions Summary
CYO %>% group_by(Light_Conditions) %>% summarise(n=n())
```

```
## # A tibble: 5 x 2
##   Light_Conditions                          n
##   <chr>                                 <int>
## 1 Darkeness: No street lighting         74342
## 2 Darkness: Street lighting unknown     14506
## 3 Darkness: Street lights present and lit   266556
## 4 Darkness: Street lights present but unlit   6272
## 5 Daylight: Street light present        992059
```

```
#Light Conditions Distribution
CYO %>% ggplot(aes(Light_Conditions)) +
geom_bar() +
theme(axis.text.x = element_text(angle = 45,
vjust = 1, hjust = 1)) + ggtitle("Light Conditions Accident Count Distribution")
```

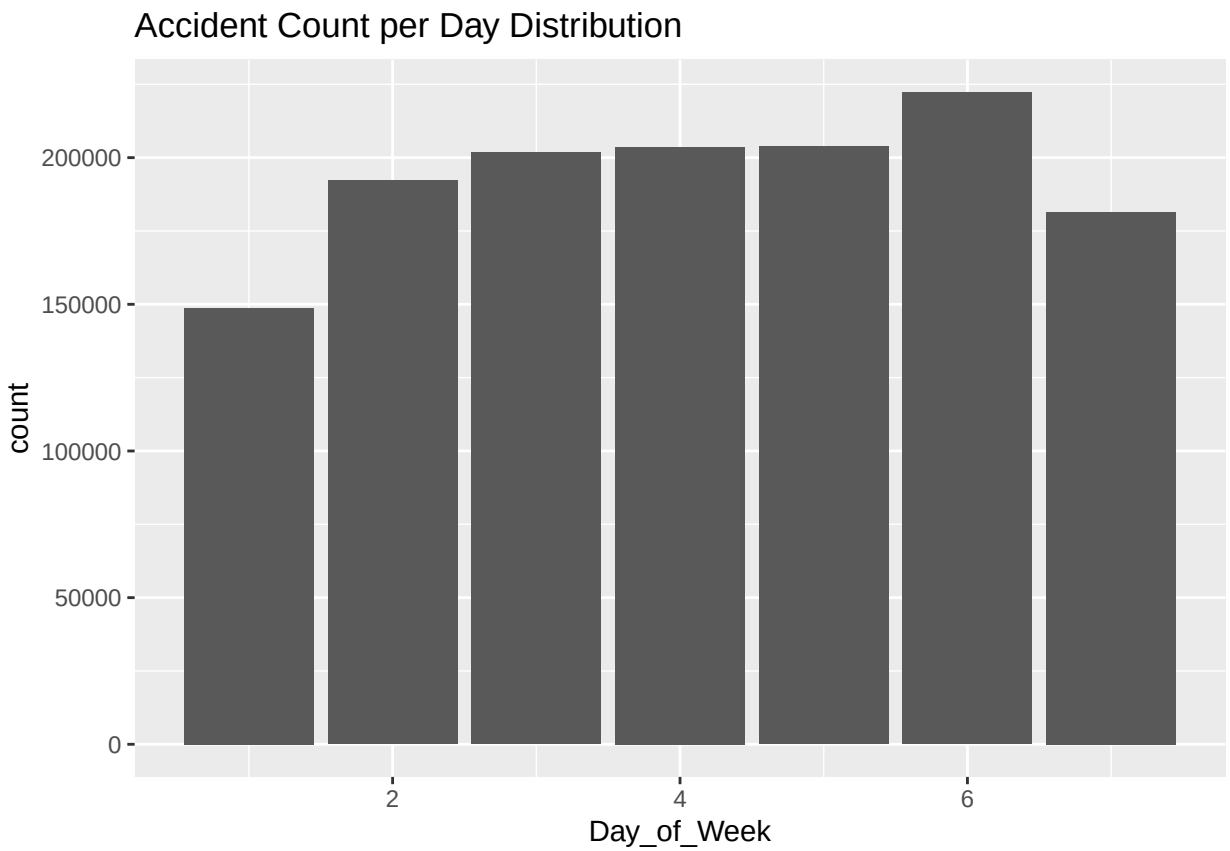


Looking at the data for the distribution of accidents based on lighting conditions, the most number of accidents occurred when there was daylight with street lights being present.

```
#Day of the Week Summary
CYO %>% group_by(Day_of_Week) %>% summarise(n=n())
```

```
## # A tibble: 7 x 2
##   Day_of_Week      n
##         <int>  <int>
## 1           1 148661
## 2           2 192268
## 3           3 201682
## 4           4 203678
## 5           5 203764
## 6           6 222263
## 7           7 181419
```

```
#Day of the Week Distribution
CYO %>% ggplot(aes(Day_of_Week)) + geom_bar() + ggtitle("Accident Count per Day Distribution")
```

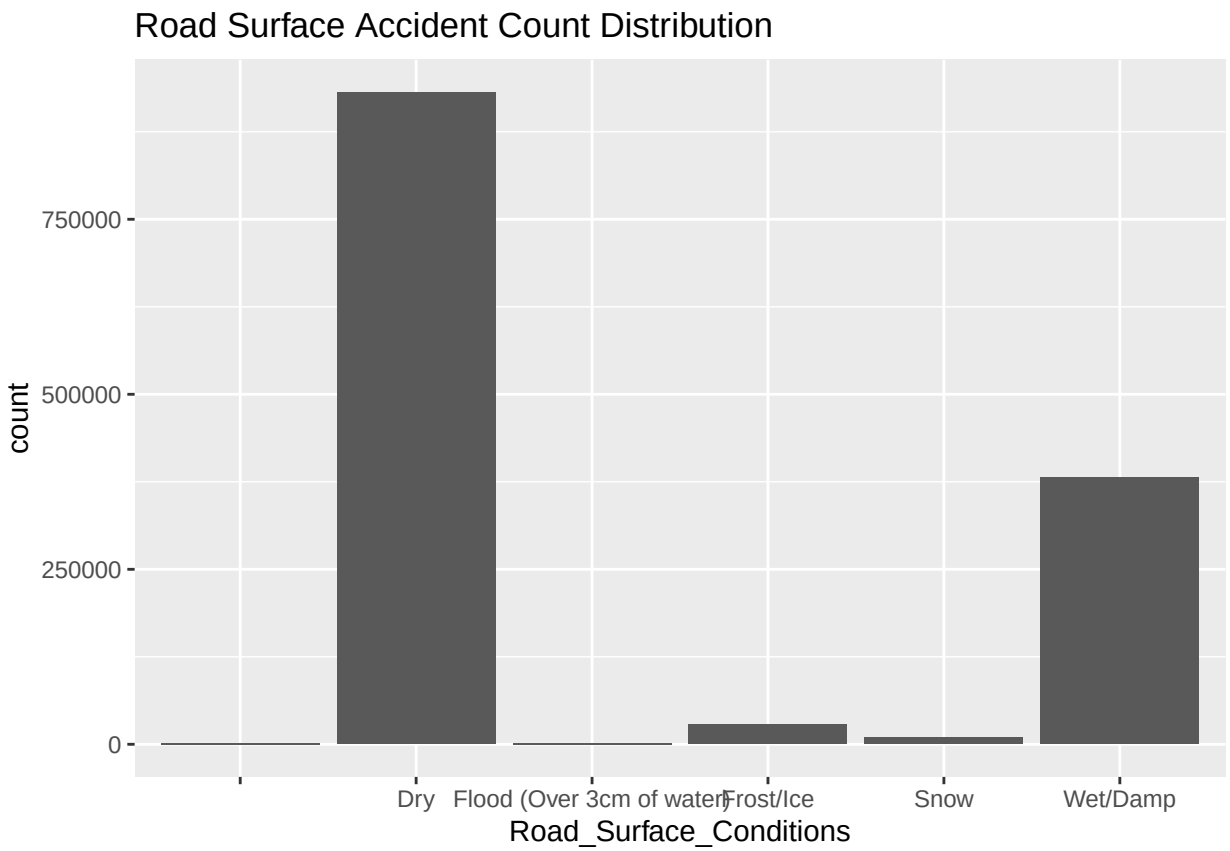


Looking at the data for the distribution of accidents over each day of the week, the most number of accidents occurred on the sixth day of the week.

```
#Road Surface Conditions
CYO %>% group_by(Road_Surface_Conditions) %>% summarise(n=n())
```

```
## # A tibble: 6 x 2
##   Road_Surface_Conditions          n
##   <chr>                        <int>
## 1 ""                            1753
## 2 "Dry"                       931344
## 3 "Flood (Over 3cm of water)"   1938
## 4 "Frost/Ice"                  28251
## 5 "Snow"                        9474
## 6 "Wet/Damp"                  380975
```

```
#Road Surface Conditions Distribution
CYO %>% ggplot(aes(Road_Surface_Conditions)) + geom_bar() +
ggtitle("Road Surface Accident Count Distribution") + scale_x_discrete(labels=c("Flood
(Over 3cm of water)"= "Flood"))
```

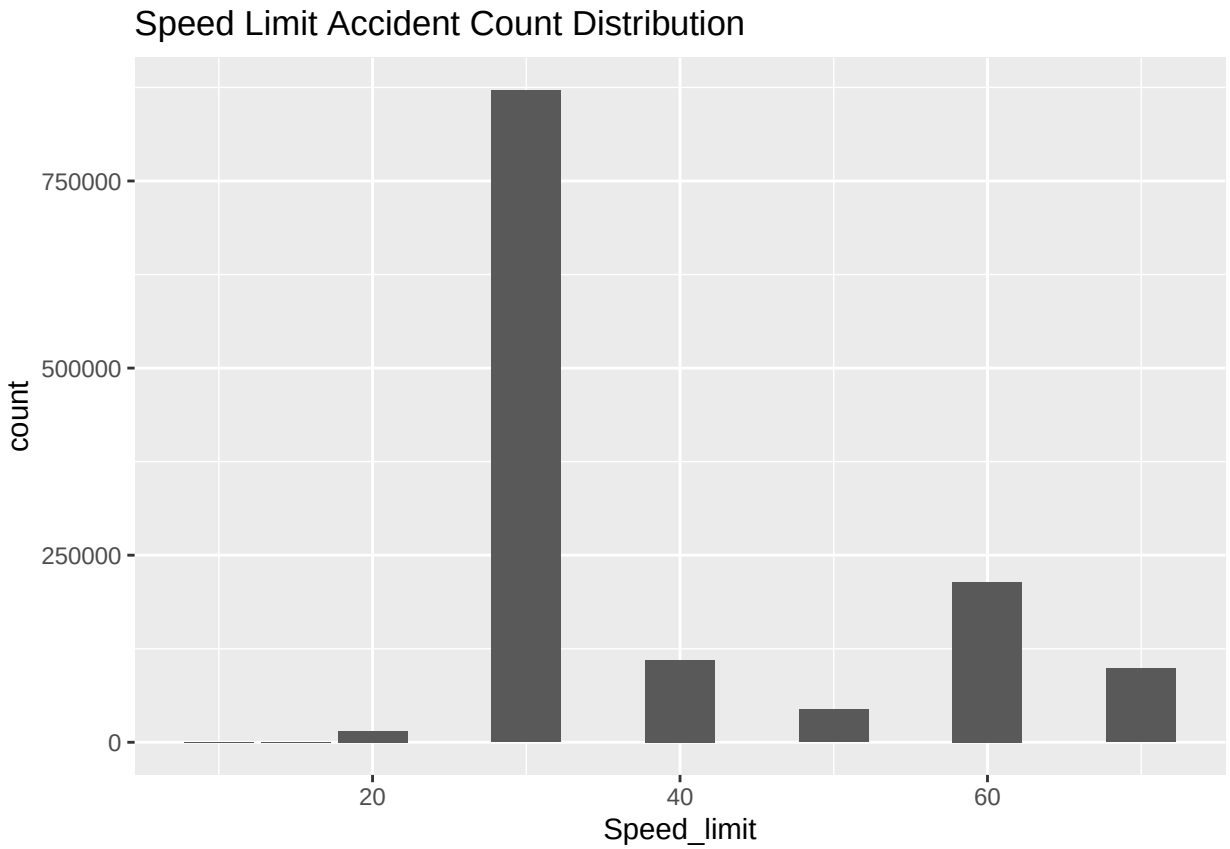


Looking at the data for the distribution of accidents over the different road surface conditions, the most number of accidents occurred when the roads were dry.

```
#Speed Limit
CYO %>% group_by(Speed_limit) %>% summarise(n=n())
```

```
## # A tibble: 8 x 2
##   Speed_limit      n
##         <int>  <int>
## 1          10     12
## 2          15     10
## 3          20  15455
## 4          30 871098
## 5          40 110254
## 6          50  43930
## 7          60 214487
## 8          70  98489
```

```
#Speed Limit Distribution
CYO %>% ggplot(aes(Speed_limit)) + geom_bar() + ggtitle("Speed Limit Accident Count Distribution")
```

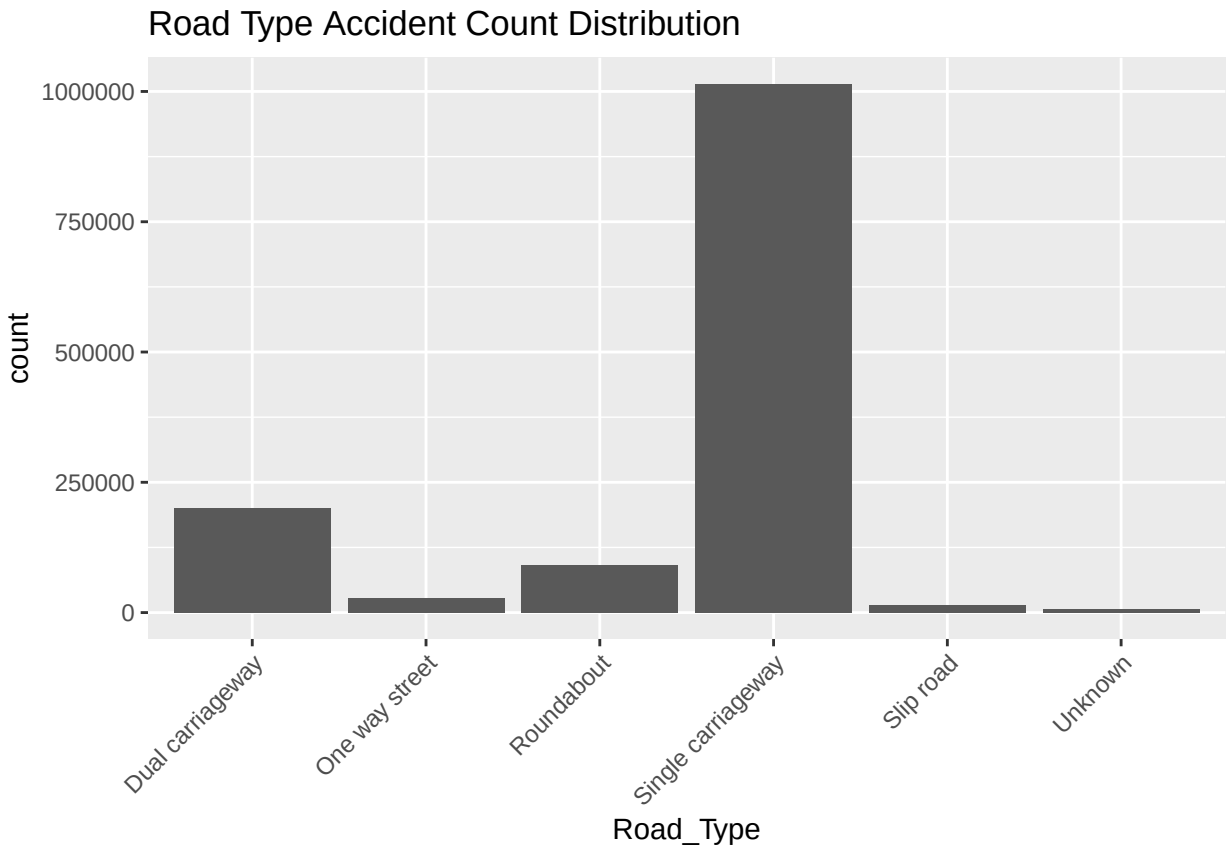


Looking at the data for the distribution of accidents over the different speed limit markers, the most number of accidents occurred on roads with a 30mph speed limit.

```
#Road Type
CYO %>% group_by(Road_Type) %>% summarise(n=n())
```

```
## # A tibble: 6 x 2
##   Road_Type               n
##   <chr>               <int>
## 1 Dual carriageway    199952
## 2 One way street       27847
## 3 Roundabout           90339
## 4 Single carriageway 1014031
## 5 Slip road            14078
## 6 Unknown               7488
```

```
#Road Type Distribution
CYO %>% ggplot(aes(Road_Type)) + geom_bar() +
theme(axis.text.x = element_text(angle = 45,
vjust = 1, hjust = 1)) +
ggtitle("Road Type Accident Count Distribution")
```

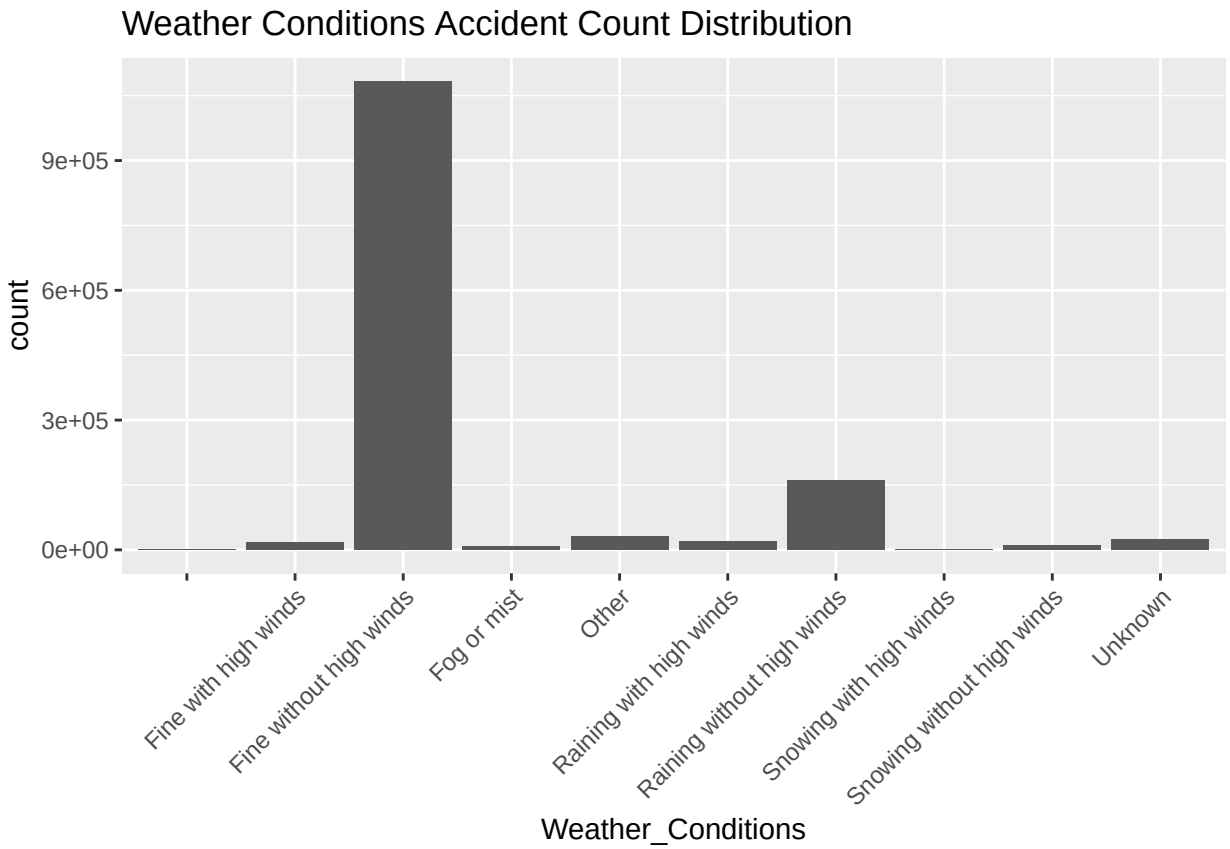


Looking at the data for the distribution of accidents over the different road types, the most number of accidents occurred on on single carriageways.

```
#Weather Conditions
CYO %>% group_by(Weather_Conditions) %>% summarise(n=n())
```

```
## # A tibble: 10 x 2
##    Weather_Conditions                n
##    <chr>                         <int>
##  1 ""                              106
##  2 "Fine with high winds"        16473
##  3 "Fine without high winds"   1083418
##  4 "Fog or mist"                  7361
##  5 "Other"                       30266
##  6 "Raining with high winds"     18723
##  7 "Raining without high winds" 160056
##  8 "Snowing with high winds"      1757
##  9 "Snowing without high winds"  10175
## 10 "Unknown"                     25400
```

```
#Weather Conditions Distribution
CYO %>% ggplot(aes(Weather_Conditions)) + geom_bar() +
theme(axis.text.x = element_text(angle = 45,
vjust = 1, hjust = 1)) +
ggtitle("Weather Conditions Accident Count Distribution")
```

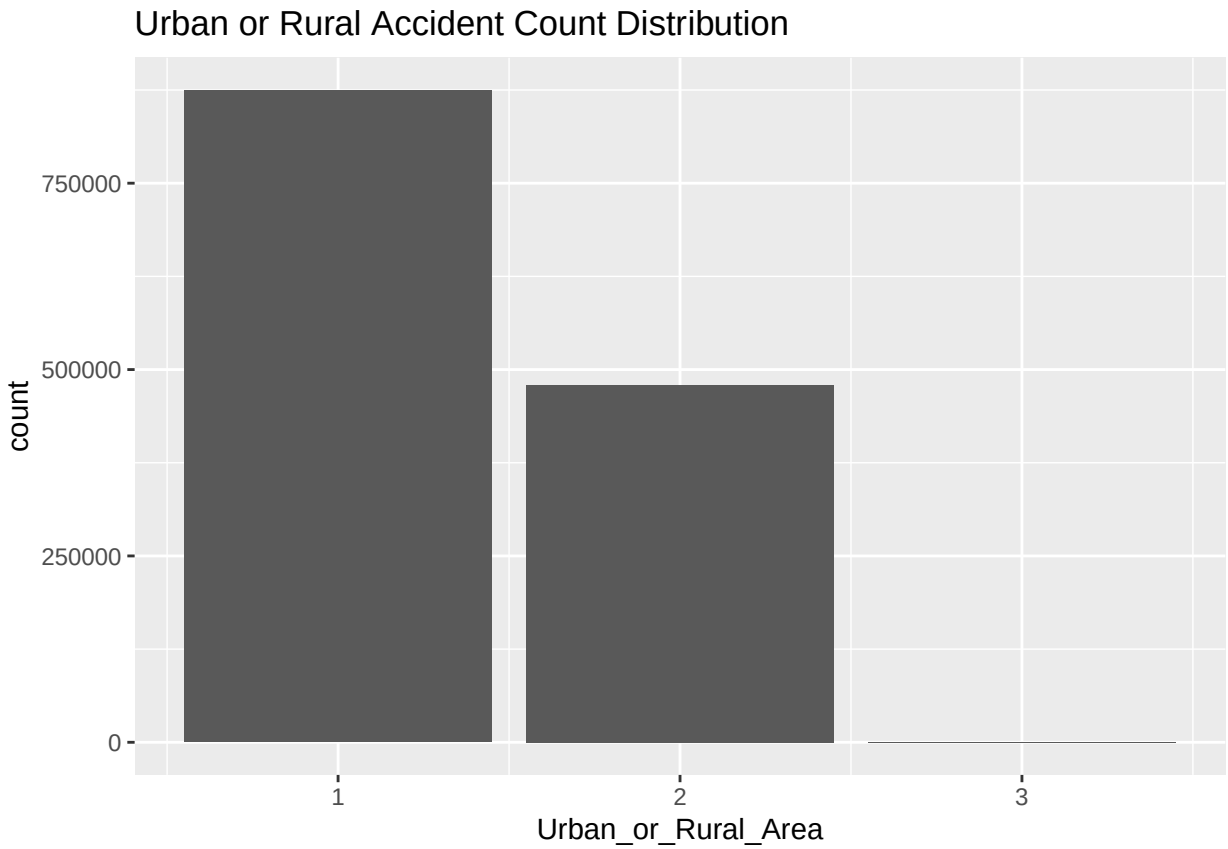


Looking at the data for the distribution of accidents across the different weather conditions, the most number of accidents occurred when the weather was fine with no high winds.

```
#Urban or Rural
CYO %>% group_by(Urban_or_Rural_Area) %>% summarise(n=n())
```

```
## # A tibble: 3 x 2
##   Urban_or_Rural_Area      n
##                 <int>  <int>
## 1                   1 874548
## 2                   2 479067
## 3                   3    120
```

```
#Urban or Rural Distribution
CYO %>% ggplot(aes(Urban_or_Rural_Area)) + geom_bar() +
ggtitle("Urban or Rural Accident Count Distribution")
```



Looking at the data for the distribution of accidents between urban and rural areas, the most number of accidents occurred in urban areas.

Results

Initial Model

```
mu1 <- mean(training_set$Accident_Severity)
mu1
```

```
## [1] 2.838584
```

Light Condition Effect Model

```
light_conditions <- training_set %>%
  group_by(Light_Conditions) %>%
  summarize(lc = mean(Accident_Severity - mu1))

predictions_lighting_conditions <- mu1 + testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  pull(lc)
```

```
rmse_lighting_conditions <- RMSE(testing_set$Accident_Severity, predictions_lighting_conditions)
rmse_lighting_conditions
```

```
## [1] 0.4007661
```

Light Condition + Day of the Week Effect model

```
day_of_week <- training_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  group_by(Day_of_Week) %>%
  summarize(dw = mean(Accident_Severity - mu1 - lc))

predictions_day_of_week <- testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  mutate(predict = mu1 + lc + dw) %>%
  pull(predict)

rmse_day_of_week <- RMSE(testing_set$Accident_Severity, predictions_day_of_week)
rmse_day_of_week
```

```
## [1] 0.4004761
```

Light Condition + Day of the Week + Road Surface Conditions Effect model

```
road_surface_conditions <- training_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  group_by(Road_Surface_Conditions) %>%
  summarize(rc = mean(Accident_Severity - mu1 - lc - dw))

predictions_road_surface_conditions <- testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  mutate(predict = mu1 + lc + dw + rc) %>%
  pull(predict)

rmse_road_surface_conditions <-
RMSE(testing_set$Accident_Severity, predictions_road_surface_conditions)

rmse_road_surface_conditions
```

```
## [1] 0.4003265
```

Light Condition + Day of the Week + Road Surface Conditions + Speed Limit Effect model

```
speed_limit <- training_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  group_by(Speed_limit) %>%
  summarize(sl = mean(Accident_Severity - mu1 - lc - dw - rc))

predictions_speed_limit <- testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by ="Speed_limit") %>%
  mutate(predict = mu1 + lc + dw + rc +sl) %>%
  pull(predict)

rmse_speed_limit <- RMSE(testing_set$Accident_Severity, predictions_speed_limit)
rmse_speed_limit
```

```
## [1] 0.3988829
```

Light Condition + Day of the Week + Road Surface Conditions + Speed Limit + Road Type Effect model

```
road_type <- training_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by = "Speed_limit") %>%
  group_by(Road_Type) %>%
  summarize(rt = mean(Accident_Severity - mu1 - lc - dw - rc - sl))

predictions_road_type <- testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by ="Speed_limit") %>%
  left_join(road_type, by = "Road_Type") %>%
  mutate(predict = mu1 + lc + dw + rc +sl + rt) %>%
  pull(predict)

rmse_road_type <- RMSE(testing_set$Accident_Severity, predictions_road_type)
rmse_road_type
```

```
## [1] 0.398425
```

Light Condition + Day of the Week + Road Surface Conditions + Speed Limit + Road Type + Weather Conditions Effect model

```
weather_conditions <- training_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by = "Speed_limit") %>%
  left_join(road_type, by = "Road_Type") %>%
  group_by(Weather_Conditions) %>%
  summarize(wc = mean(Accident_Severity - mu1 - lc - dw - rc - sl + rt))

predictions_weather_conditions <- testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by ="Speed_limit") %>%
  left_join(road_type, by = "Road_Type") %>%
  left_join(weather_conditions, by = "Weather_Conditions") %>%
  mutate(predict = mu1 + lc + dw + rc +sl + rt + wc) %>%
  pull(predict)

rmse_weather_conditions <- RMSE(testing_set$Accident_Severity, predictions_weather_conditions)
rmse_weather_conditions
```

```
## [1] 0.3982923
```

Light Condition + Day of the Week + Road Surface Conditions + Speed Limit + Road Type + Weather Conditions + Urban or Rural Effect model

```
urban_or_rural <- training_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by = "Speed_limit") %>%
  left_join(road_type, by = "Road_Type") %>%
  left_join(weather_conditions, by = "Weather_Conditions") %>%
  group_by(Urban_or_Rural_Area) %>%
  summarize(ur = mean(Accident_Severity - mu1 - lc - dw - rc - sl + rt + wc))

predictions_urban_or_rural <- testing_set %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by ="Speed_limit") %>%
  left_join(road_type, by = "Road_Type") %>%
  left_join(weather_conditions, by = "Weather_Conditions") %>%
  left_join(urban_or_rural, by = "Urban_or_Rural_Area") %>%
  mutate(predict = mu1 + lc + dw + rc +sl + rt + wc + ur) %>%
  pull(predict)
```

```
rmse_urban_or_rural <- RMSE(testing_set$Accident_Severity, predictions_urban_or_rural)
rmse_urban_or_rural
```

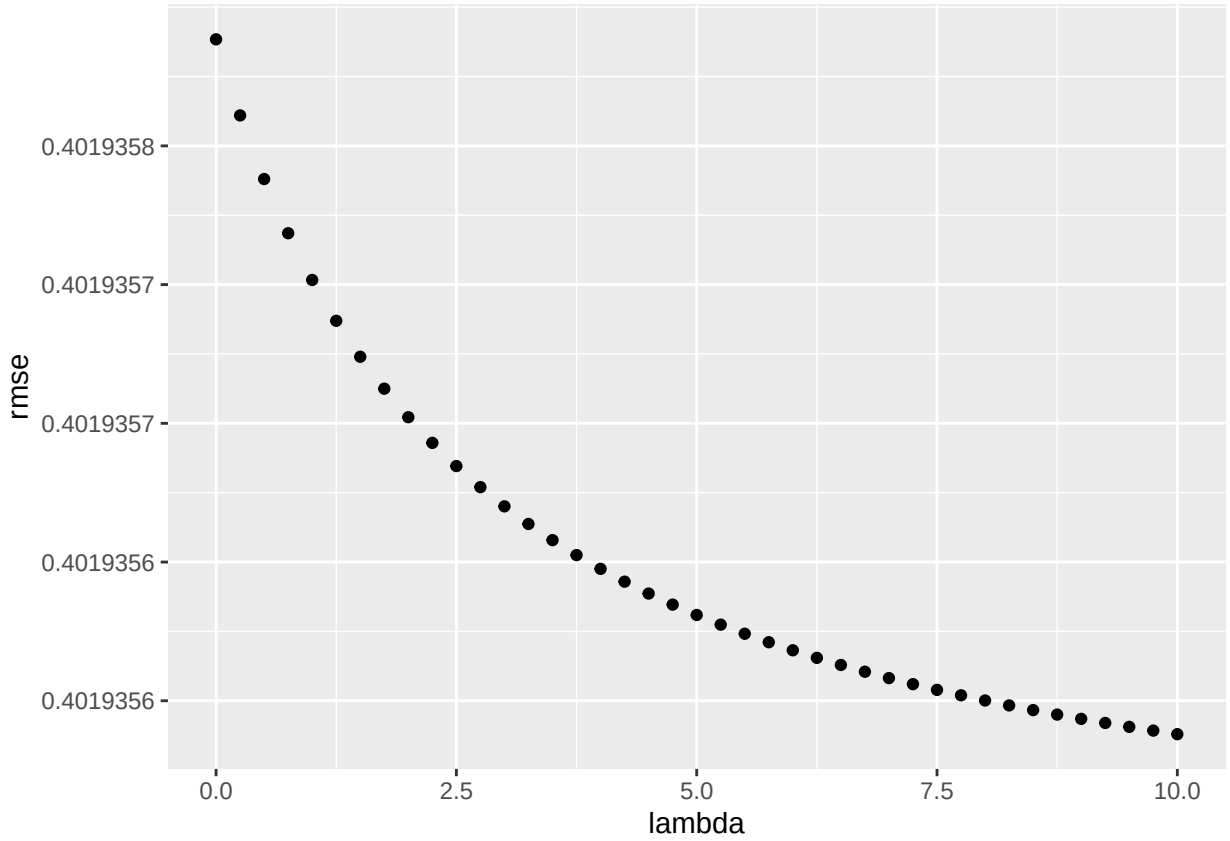
```
## [1] 0.3982103
```

Regularization Model

From our data exploration we found that a lot of these variables have unimodal or bimodal distributions which indicate non-normality and a significant skewness in the data. In order to create a model which avoids over-fitting, the model must be regularized for higher accuracy.

```
mu1 <- mean(training_set$Accident_Severity)

lambda <- seq(0, 10, 0.25)

rmse <- sapply(lambda, function(lmd){

  light_conditions <- training_set %>%
    group_by(Light_Conditions) %>%
    summarize(lc = mean(Accident_Severity - mu1)/(n()+lmd))

  day_of_week <- training_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    group_by(Day_of_Week) %>%
    summarize(dw = mean(Accident_Severity - mu1 - lc)/(n()+lmd))

  road_surface_conditions <- training_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    left_join(day_of_week, by = 'Day_of_Week') %>%
    group_by(Road_Surface_Conditions) %>%
    summarize(rc = mean(Accident_Severity - mu1 - lc - dw)/(n()+lmd))

  speed_limit <- training_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    left_join(day_of_week, by = 'Day_of_Week') %>%
    left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
    group_by(Speed_limit) %>%
    summarize(sl = mean(Accident_Severity - mu1 - lc - dw - rc)/(n()+lmd))

  road_type <- training_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    left_join(day_of_week, by = 'Day_of_Week') %>%
    left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
    left_join(speed_limit, by = "Speed_limit") %>%
    group_by(Road_Type) %>%
    summarize(rt = mean(Accident_Severity - mu1 - lc - dw - rc - sl)/(n()+lmd))

  weather_conditions <- training_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    left_join(day_of_week, by = 'Day_of_Week') %>%
    left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
    left_join(speed_limit, by = "Speed_limit") %>%
```

```
    left_join(road_type, by = "Road_Type") %>%
    group_by(Weather_Conditions) %>%
    summarize(wc = mean(Accident_Severity - mu1 - lc - dw - rc - sl + rt)/(n()+lmd))

  urban_or_rural <- training_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    left_join(day_of_week, by = 'Day_of_Week') %>%
    left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
    left_join(speed_limit, by = "Speed_limit") %>%
    left_join(road_type, by = "Road_Type") %>%
    left_join(weather_conditions, by = "Weather_Conditions") %>%
    group_by(Urban_or_Rural_Area) %>%
    summarize(ur = mean(Accident_Severity - mu1 - lc - dw - rc - sl + rt + wc)/(n()+lmd))


  predictions_total <- testing_set %>%
    left_join(light_conditions, by = "Light_Conditions") %>%
    left_join(day_of_week, by = 'Day_of_Week') %>%
    left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
    left_join(speed_limit, by ="Speed_limit") %>%
    left_join(road_type, by = "Road_Type") %>%
    left_join(weather_conditions, by = "Weather_Conditions") %>%
    left_join(urban_or_rural, by = "Urban_or_Rural_Area") %>%
    mutate(predict = mu1 + lc + dw + rc +sl + rt + wc + ur) %>%
    pull(predict)

  RMSE(predictions_total, testing_set$Accident_Severity)
})
qplot(lambda, rmse)
```




```
lowest_rmse <- rmse[which.min(rmse)]
lowest_rmse
```

```
## [1] 0.4019356
```

```
lowest_lambda <- lambda[which.min(rmse)]
lowest_lambda
```

```
## [1] 10
```

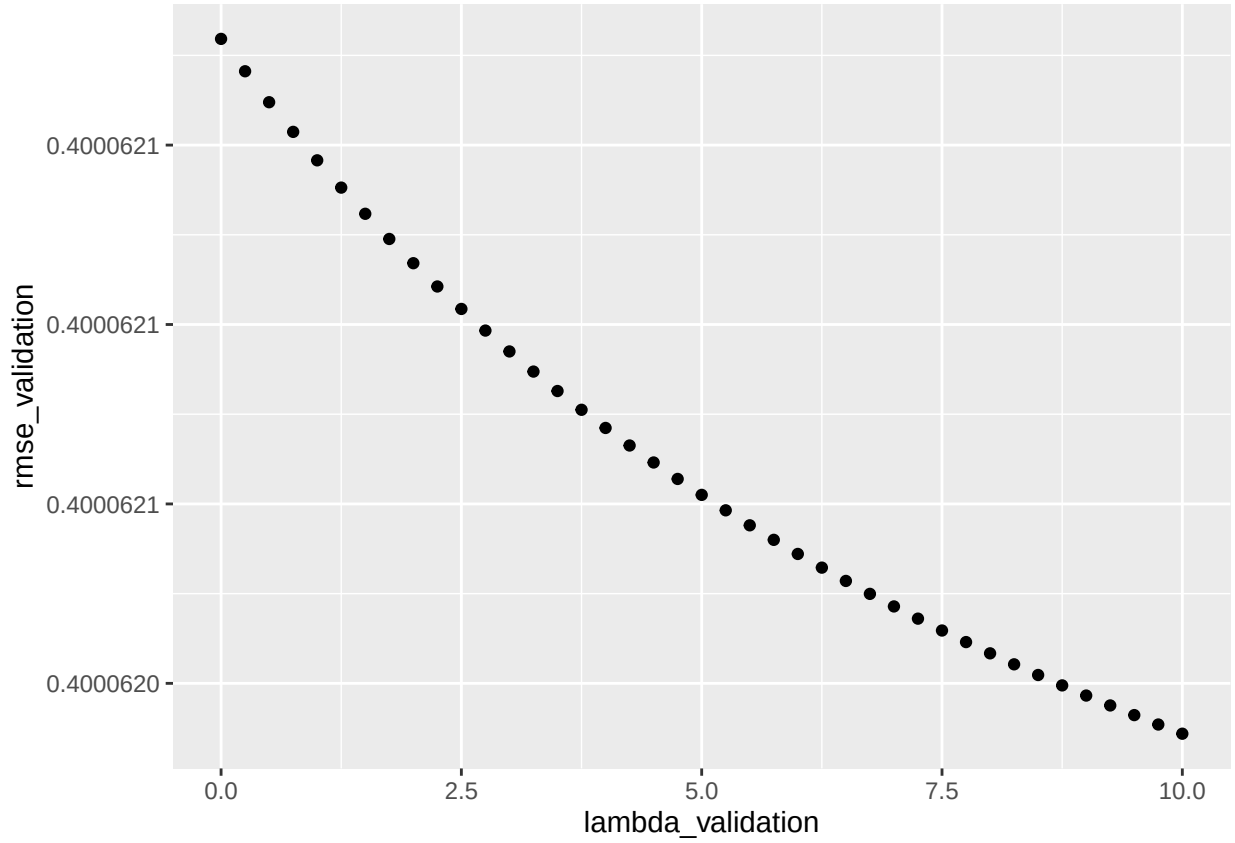
Since the RMSE value is below our target of 0.45 a final run is done with the Validation set

Validation run

```
mu1 <- mean(CYO$Accident_Severity)

lambda_validation <- seq(0, 10, 0.25)

rmse_validation <- sapply(lambda_validation, function(lmd){

  light_conditions_validation <- CYO %>%
    group_by(Light_Conditions) %>%
```

```
  summarize(lc = mean(Accident_Severity - mu1)/(n()+lmd))

day_of_week_validation <- CYO %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  group_by(Day_of_Week) %>%
  summarize(dw = mean(Accident_Severity - mu1 - lc)/(n()+lmd))

road_surface_conditions_validation <- CYO %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  left_join(day_of_week_validation, by = 'Day_of_Week') %>%
  group_by(Road_Surface_Conditions) %>%
  summarize(rc = mean(Accident_Severity - mu1 - lc - dw)/(n()+lmd))

speed_limit_validation <- CYO %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  left_join(day_of_week_validation, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions_validation, by = "Road_Surface_Conditions") %>%
  group_by(Speed_limit) %>%
  summarize(sl = mean(Accident_Severity - mu1 - lc - dw - rc)/(n()+lmd))

road_type_validation <- CYO %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  left_join(day_of_week_validation, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions_validation, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit_validation, by = "Speed_limit") %>%
  group_by(Road_Type) %>%
  summarize(rt = mean(Accident_Severity - mu1 - lc - dw - rc - sl)/(n()+lmd))

weather_conditions_validation <- CYO %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  left_join(day_of_week_validation, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions_validation, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit_validation, by = "Speed_limit") %>%
  left_join(road_type_validation, by = "Road_Type") %>%
  group_by(Weather_Conditions) %>%
  summarize(wc = mean(Accident_Severity - mu1 - lc - dw - rc - sl + rt)/(n()+lmd))

urban_or_rural_validation <- CYO %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  left_join(day_of_week_validation, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions_validation, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit_validation, by = "Speed_limit") %>%
  left_join(road_type_validation, by = "Road_Type") %>%
  left_join(weather_conditions_validation, by = "Weather_Conditions") %>%
  group_by(Urban_or_Rural_Area) %>%
  summarize(ur = mean(Accident_Severity - mu1 - lc - dw - rc - sl + rt + wc)/(n()+lmd))


predictions_total_validation <- Validation %>%
  left_join(light_conditions_validation, by = "Light_Conditions") %>%
  left_join(day_of_week_validation, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions_validation, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit_validation, by ="Speed_limit") %>%
```

```
    left_join(road_type_validation, by = "Road_Type") %>%
    left_join(weather_conditions_validation, by = "Weather_Conditions") %>%
    left_join(urban_or_rural_validation, by = "Urban_or_Rural_Area") %>%
    mutate(predict_validation = mu1 + lc + dw + rc +sl + rt + wc + ur) %>%
    pull(predict_validation)

  RMSE(predictions_total_validation, Validation$Accident_Severity)
})

qplot(lambda_validation, rmse_validation)
```




```
lowest_rmse_validation <- rmse_validation[which.min(rmse_validation)]
lowest_rmse_validation
```

```
## [1] 0.400062
```

```
lowest_lambda_validation <- lambda_validation[which.min(rmse_validation)]
lowest_lambda_validation
```

```
## [1] 10
```

Prediction list of the worst accidents

```
#Prediction list of the top 15 most severe accidents
Final_List <- Validation %>%
  left_join(light_conditions, by = "Light_Conditions") %>%
  left_join(day_of_week, by = 'Day_of_Week') %>%
  left_join(road_surface_conditions, by = "Road_Surface_Conditions") %>%
  left_join(speed_limit, by ="Speed_limit") %>%
  left_join(road_type, by = "Road_Type") %>%
  left_join(weather_conditions, by = "Weather_Conditions") %>%
  left_join(urban_or_rural, by = "Urban_or_Rural_Area") %>%
  mutate(prediction = mu1 + lc + dw + rc +sl + rt + wc + ur) %>%
  arrange(-prediction) %>%
  group_by(Accident_Index) %>%
  select(Accident_Index) %>%
  head(15)

Final_List
```

```
## # A tibble: 15 x 1
## # Groups:   Accident_Index [11]
##    Accident_Index
##    <chr>
##  1 20053102C3569
##  2 200732B062207
##  3 200604EA06326
##  4 200506B039723
##  5 20073102C4382
##  6 200540D006390
##  7 2.01E+12
##  8 2.01E+12
##  9 200720L025901
## 10 2.01E+12
## 11 20103102D0657
## 12 2.01E+12
## 13 201001RG40008
## 14 2.01E+12
## 15 201004EA10004
```

Conclusion

The purpose of this project was to develop an accident severity prediction model using data from the United Kingdom. Based on the results of the Regularized Cross Validation model the final RMSE achieved, with the Validation set, was 0.400062 which is well below our target RMSE of 0.45 and we were able to generate a list of the top 15 most severe accidents with the conditions added in the model. While the target was achieved a more accurate model could have been constructed by utilizing more variables in order to further decrease the RMSE. In the future, there is potential to apply this model to data from other nations aside from the United Kingdom. It would be interesting to see the similarities and differences when comparing the model across different nations but the end goal would be to use the information to try to prevent more accidents from occurring.