Analysis of a Countrywide Traffic Accident Dataset

Harsh Sugandh (suganh@rpi.edu)

Rensselaer Polytechnic Institute, Tetherless World Constellation, Troy, NY, United States

Abstract

Reducing traffic accidents is an important public safety challenge. However, the majority of studies on traffic accident analysis and prediction have used small-scale datasets with limited coverage, which limits their impact and applicability; and existing large-scale datasets are either private, old, or do not include important contextual information such as environmental stimuli (weather, points-of-interest, etc.). US-Accidents dataset currently contains data about 2.25 million instances of traffic accidents that took place within the contiguous United States, and over the last three years. Each accident record consists of a variety of intrinsic and contextual attributes such as location, time, natural language description, weather, period-of-day, and points-of-interest.

In this poster, US-Accidents dataset is used for applications such as studying accident hotspot locations; casualty analysis (extracting cause and effect rules to predict accidents); or studying the impact of precipitation or other environmental stimuli on accident occurrence.

Motivation

Reducing traffic accidents is an important public safety challenge around the world. A global status report on traffic safety, notes that there were 1.25 million traffic deaths in 2013 alone, with deaths increasing in 68 countries when compared to 2010. Accident prediction is important for optimizing public transportation, enabling safer routes, and cost-effectively improving the transportation infrastructure, all in order to make the roads safer.

The current presentation is to analyse the dataset and apply different predictive models for predicting the severity of the accidents on the basis of the environmental factors.

Dataset

The data is collected from February 2016 to March 2019, using several data providers, including two APIs which provide streaming traffic event data. These APIs broadcast traffic events captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks.

The severity of the accidents are on the scale from 1 to 4, 1 being the least severe and 4 being the most severe.

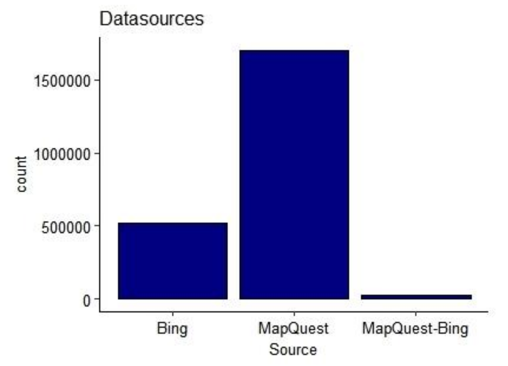

Fig 1: Datasource count

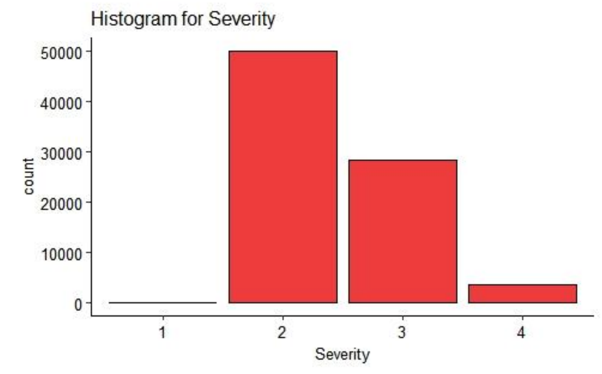

Fig 1: Severity count

Analyzing the Dataset

1. Importing & inspecting data:

```
library(readr)
dataset<-read_csv("US_Accidents_May19.csv")
view(dataset)
summary(dataset)
```

2. Cleaning and maipulating the data:

```
attach(dataset)
#removing null values
dataset <- dataset[!is.na(`Temperature(F)`) & !is.na (`Wind_Chill(F)`) &
!is.na(`Humidity(%)`) & !is.na (`Pressure(in)`)& !is.na(`Visibility(mi)`) &
!is.na (`Wind_Speed(mph)`) & !is.na(`Precipitation(in)`)]

#Converting categorical value to factors
dataset[,c("Wind_Direction","Weather_Timestamp", "Severity")]<-
lapply(dataset[,c("Wind_Direction","Weather_Timestamp","Severity" )],
factor)
```

3. Exploring the data:

```
dataset$Weather_Timestamp <- weekdays
(dataset$Weather_Timestamp)

#Ploting Accidents by weekday
ggplot(aes(x=Weekday),data=dataset)+ geom_histogram (stat =
"count", fill='orangered2')+ ggtitle(" Accidents by weekday") + theme
(plot.margin = unit(c(1, 1, 1, 1), "cm"),text = element_text(size=10))

#Plotting Frequency distribution of US Accidents
ggplot(aes(x=State),data=dataset)+geom_histogram(stat="count",fill=
'plum3')+ggtitle("Frequency distribution of US Accidents")+Theme
(text=element_text(size=8),axis.text.x=element_text(angle=90,
hjust=0.1 ,vjust=0.5))
```

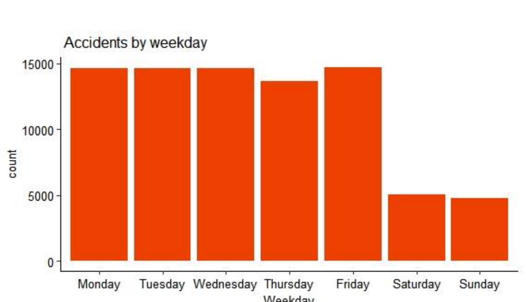

Fig 3: Accidents by weekday

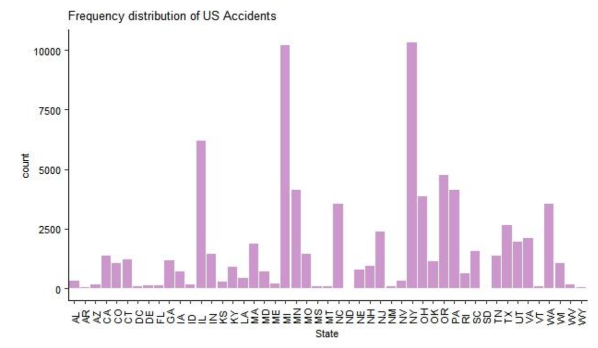

Fig 4: Frequency distribution of US Accidents

```
#Histograms and boxplots
g1<-ggplot(aes(x=Temperature),data=dataset)+geom_histogram
(binwidth=10,fill='brown2',color='black') + ggtitle("Histogram for
Temperature (F)")+ theme(text = element_text(size=8))
g2<-ggplot(data=dataset, aes(x = "", y = Temperature))+geom_boxplot
(fill='cyan3',color='black')+ theme(text = element_text(size=8))
grid.arrange(g1,g2,nrow=1)

g1<-ggplot(aes(x=Pressure), data=dataset)+geom_histogram
(binwidth=0.1,fill='brown2',color='black')+ggtitle
("Histogram for Pressure(in)") + theme(text = element_text(size=8))
g2<-ggplot(data = dataset, aes(x = "", y = Pressure)) + geom_boxplot
(fill='cyan3',color='black')+ theme(text = element_text(size=8))
grid.arrange(g1,g2,nrow=1)
```

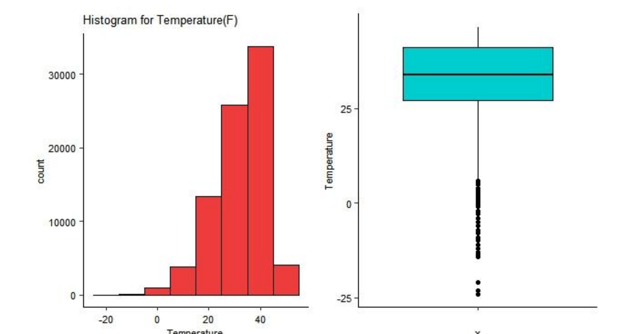

Fig 5: Histogram and Boxplot for Temperature(F)

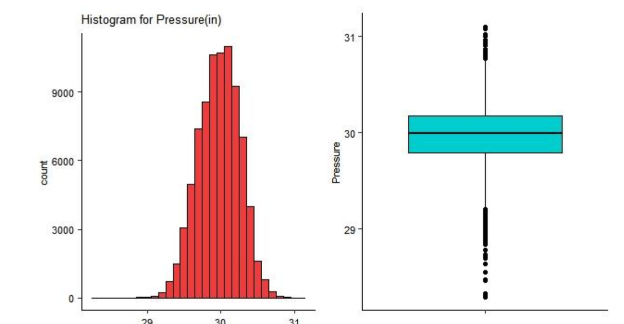

Fig 6: Histogram and Boxplot for Pressure(in)

```
g1<-ggplot(aes(x=Wind_Chill), data=dataset)+geom_histogram
(binwidth = 10, fill='brown2',color='black')+ggtitle ("Histogram for
Wind_Chill(F)")+theme(text = element_text(size=8))
g2<-ggplot(data = dataset, aes(x = "", y = Wind_Chill))+geom_boxplot
(fill='cyan3',color='black')+theme(text = element_text(size=8))
grid.arrange(g1,g2,nrow=1)

g1<-ggplot(aes(x=Visibility),data=dataset)+ geom_histogram
(binwidth=1, fill='brown2',color='black')+ggtitle("Histogram for
Visibility(mi)") +theme (text = element_text(size=8))
g2<-ggplot(data = dataset, aes(x = "", y = Visibility))+ geom_boxplot
(fill='cyan3',color='black')+theme(text = element_text(size=8))
grid.arrange(g1,g2,nrow=1)
```

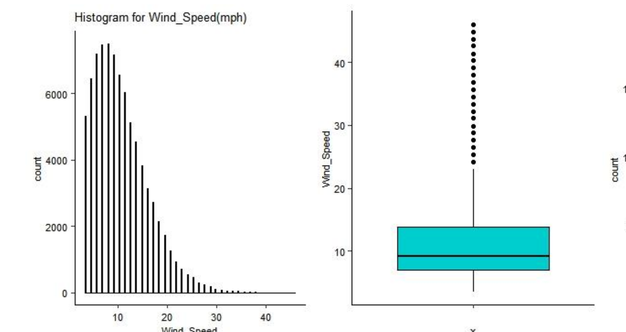

Fig 7: Histogram and Boxplot for Wind_Speed(mph)

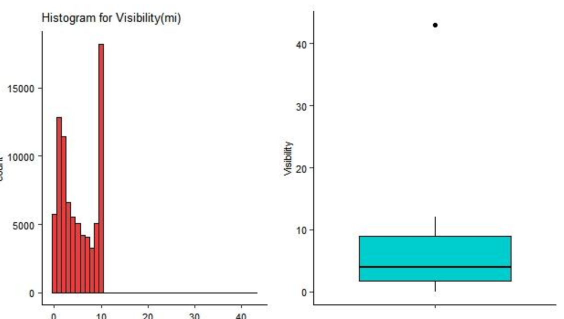

Fig 8: Histogram and Boxplot for Visibilty(mi)

4. Applying Predictive Models:

```
#Splitting the dataset into traindata and testdata
bound <- floor((nrow(dataset)/4)*3) and test set
dataset <- dataset[sample(nrow(dataset)), ]
traindata <- dataset[1:bound, ]
testdata <- dataset[(bound+1):nrow(dataset), ]
```

A. Multinomial regression

```
severity_mult <- multinom(Severity ~ Temperature+Wind_Chill+
Humidity+Pressure+Visibility+Wind_Speed +Precipitation,
data = traindata)
summary(severity_mult)
```

```
> summary(severity_mult)
Call:
multinom(formula = Severity ~ Temperature + Wind_Chill + Humidity +
    Pressure + Visibility + Wind_Speed + Precipitation, data = traindata)

Coefficients:
Call:
multinom(formula = Severity ~ Temperature + Wind_Chill + Humidity +
    Pressure + Visibility + Wind_Speed + Precipitation, data = traindata)

Coefficients:
  (Intercept) Temperature Wind_Chill    Humidity  Pressure Visibility Wind_Speed
2   0.1999308   -1.062017  0.8446095 0.007308396 0.5556124 0.06258538  0.3962536
3   1.8941958   -1.075298  0.8428080 0.008408281 0.4931536 0.04244740  0.4038640
4   9.3241664   -1.226231  0.9553217 0.013108188 0.2076249 0.06840265  0.4648672
  Precipitation
2     0.8049890
3     0.8806836
4     0.7480285

Std. Errors:
  (Intercept) Temperature Wind_Chill   Humidity  Pressure Visibility Wind_Speed
2 0.003397969  0.02949331 0.01944358 0.03545875 0.1060983  0.1037762 0.07317765
3 0.003404469  0.02972406 0.01966414 0.03546340 0.1061167  0.1037885 0.07319538
4 0.003418431  0.03289698 0.02274809 0.03553083 0.1063912  0.1039690 0.07342096
  Precipitation
2    0.04819404
3    0.04827164
4    0.09053712

Residual Deviance: 98407.73
AIC: 98455.73
```

```
#constructing confusin matrix
prediction_mult<-predict(severity_mult,newdata=testdata,
type= 'class')
confusionMatrix(prediction_mult, testdata[["Severity"]])
```

```
> confusionMatrix(prediction_mult, testdata[["Severity"]])
Confusion Matrix and Statistics

          Reference
Prediction     1     2     3     4
         1     0     0     0     0
         2     4 12450  7075   887
         3     0    28    40     8
         4     0     0     0     0

Overall Statistics

               Accuracy : 0.6095
                 95% CI : (0.6028, 0.6162)
    No Information Rate : 0.6089
    P-Value [Acc > NIR] : 0.4348

                  Kappa : 0.004

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity          0.0000000  0.99776 0.005622  0.00000
Specificity          1.0000000  0.00599 0.997309  1.00000
Pos Pred Value             NaN  0.60982 0.526316      NaN
Neg Pred Value       0.9998048  0.63158 0.653458  0.95632
Prevalence           0.0001952  0.60892 0.347209  0.04368
Detection Rate       0.0000000  0.60755 0.001952  0.00000
Detection Prevalence 0.0000000  0.99629 0.003709  0.00000
Balanced Accuracy    0.5000000  0.50187 0.501465  0.50000
```

B. Random Forest

```
set.seed(1)

rf_model <- randomForest(Severity ~ Temperature+Wind_Chill+
                    Humidity+Pressure+Visibility+
                    Wind_Speed+Precipitation,traindata)
rf_model
```

```
> rf_model

Call:
 randomForest(formula = Severity ~ Temperature + Wind_Chill +      Humidity + Pressur
e + Visibility + Wind_Speed + Precipitation,      data = traindata)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 35.91%
Confusion matrix:
  1     2    3   4 class.error
1 0     9    3   0   1.0000000
2 2 31417 5954 161   0.1629722
3 1 14078 7029 133   0.6690834
4 0  1333  404 950   0.6464459
```

```
rf_result <- predict(rf_model, newdata = testdata[,!colnames(testdata)
                                  %in% c("Severity")])

# constructing confusin matrix
confusionMatrix(rf_result, testdata$Severity)
```

```
> confusionMatrix(rf_result, testdata$Severity)
Confusion Matrix and Statistics

          Reference
Prediction     1     2     3     4
         1     0     0     0     0
         2     6 10613  4768   474
         3     0  1893  2259   126
         4     0    47    29   277

Overall Statistics

               Accuracy : 0.6417
                 95% CI : (0.6351, 0.6482)
    No Information Rate : 0.6126
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2094

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity          0.0000000   0.8455   0.3202  0.31585
Specificity          1.0000000   0.3390   0.8497  0.99613
Pos Pred Value             NaN   0.6691   0.5281  0.78470
Neg Pred Value       0.9997072   0.5811   0.7041  0.97021
Prevalence           0.0002928   0.6126   0.3443  0.04280
Detection Rate       0.0000000   0.5179   0.1102  0.01352
Detection Prevalence 0.0000000   0.7740   0.2088  0.01723
Balanced Accuracy    0.5000000   0.5922   0.5849  0.65599
```

5. Conclusion:

- Frequency of accidents is highest in New York State followed by Michigan State.
- Accuracy of Multinomial Logistic Regression model is 61%.
- Accuracy of Random Forest model is 65%.

Sponsors:

Glossary:

R – A program to process data and perform statistical analysis
Library (R): software package to be loaded to perform extra tasks
Df– Data manipulation structure in R
Levels in R dataframe – for factor data, the possible number of choices are levels

Resources:

ScrapeR Package in R: https://cran.r-project.org/web/packages/scrapeR/scrapeR.pdf
R deal with missing data: https://www.statmethods.net/input/missingdata.html
R visualization: https://www.analyticsvidhya.com/blog/2015/07/guide-data-visualization-r/
http://r-statistics.co/ggplot2-Tutorial-With-R.html
https://www.rdocumentation.org/packages/ggplot2/versions/3.2.1/topics/ggplot