Final Write-Up

Azka Javaid & Caleb Ki

December 19, 2016

Introduction

In 2015, 35092 people died in motor vehicle accidents according to the National Highway Traffic Safety Association (NHTSA). This is increase of 2348 fatalities from 2014. Part of the reason that there was an increase in the number of fatal traffic accidents overall is that the number of fatal traffic accidents involving alcohol impaired drivers increased by 322. The CDC reports that around 31% of traffic fatalities include a drunk driver and that alcohol impaired vehicle accidents costs around $44 billion.

Identifying the risk factors and the behavior of drunk drivers could go a long way in reducing drunk driving. Knowing who is more likely to drunk drive gives information about who to target with intervention methods. In addition, if we are able to identify the times that driver are more likely to be drunk, the general public could be informed (e.g. if there are more alcohol immpaired drivers during weekends, the general public should be aware and would then be able to be more viligant while driving during the weekend).

Furthermore, although the dangers of and facts concerning drunk driving are well documented, the known risk factors are generally only known at the individual level. What we mean is that there is relatively little literature exploring why there are large discrepancies in rate of drunk driving between different U.S. counties. For example, in 2015, Fremont County, Wyoming, had 27 fatal traffic accidents involving alcohol per 100,000 people whereas the same statistic for Morris County, New Jersey is only 0.2. The difference between the two counties is massive. But what makes Fremont County so different from Wyoming? Why is the rate of traffic fatalities including drunk drivers much higher in county than in another? What economic and demographic differences of the two counties explain why?

Determining the factors that could help explain the discrepancy in the rates of fatal traffic accidents involving a drunk driver could pinpoint the risk factors of drunk driving at the county level. This in turn would indicate which geographic areas would benefit the most from different types of intervention.

For our project we are looking to determine what predictors/factors at the individual and county level are good predictors for whether a driver involved in a fatal traffic accident was alcohol impaired or not. In addition we would also like to see what behaviors are good predictors as well. Specifically, if a fatal accident occurs during the night or the weekend, is it more likely that the drivers were drinking?

Data

In order to answer our questions we used data from two different sources. The first and primary source of data came from the Fatality Accident Report Service (FARS) managed by the NHTSA. FARS provides information concerning every vehicle and person involved in a fatal traffic accident. The data is provided at an accident, vehicle, and person level. Each accident is given a unique identifier, and each row of the vehicle and person datasets includes the unique ID indicating which accident that the vehicle or person was a part of. This allowed us to seamlessly join the data together. The manual provided in this link contains information about each variable within the datasets.

The second and supplementary source of data came from the American Community Survey (ACS) conducted by the U.S. Census Bureau. The ACS provides social, economic, and demographic data at the county, state, and national level. The FARS data contains geographic information like latitude, longitude, county, and state. By calculating FIPS codes, we were able to match the county level data we were looking for to each accident, vehicle, and person. Information about the ACS can be found here.

Data Wrangling

BigQuery

We used Google's BigQuery as a hosting platform for our datasets. BigQuery as a service is an analytics warehouse with the ability to process data on the petabyte scale. It provides a serverless and infrastructure-less environment since it deviates the need of a database administrator given that the data is processed and stored on the cloud. Its features include the ability to ingest data from different sources including Google Cloud Storage, Cloud Datastore, and livestream. Data can be read and written via Cloud Dataflow, Spark, and Hadoop and exported out in the Cloud. A key feature of BigQuery is the ability to collaborate and share queries as well as data by adding members to a project. Since we used Person, Vehicle and Accident level data, BigQuery provided a cohesive and structured environment for managing all three. Additionally the user-friendly interface was conducive to basic exploratory analysis with SQL as well as for performing variety of joins.

Most basic and preliminary use of BigQuery entails navigating between two environments: Google BigQuery and Google Cloud Platform. Data can be uploaded in BigQuery by first creating a project from the ProjectsPage and enable billing as well as the BigQuery API. Once a project has been created, it can be selected on the Google BigQuery platform and datasets can be added from the 'create a new dataset drop down option' on the highlighted project (available on the left side of the inteface). After specifying a dataset, it is populated by tables of interest. The specification of the table entails defining a schema (structure or data skeleton), which involves defining the variable names as well the data types for each variable as shown by the image.

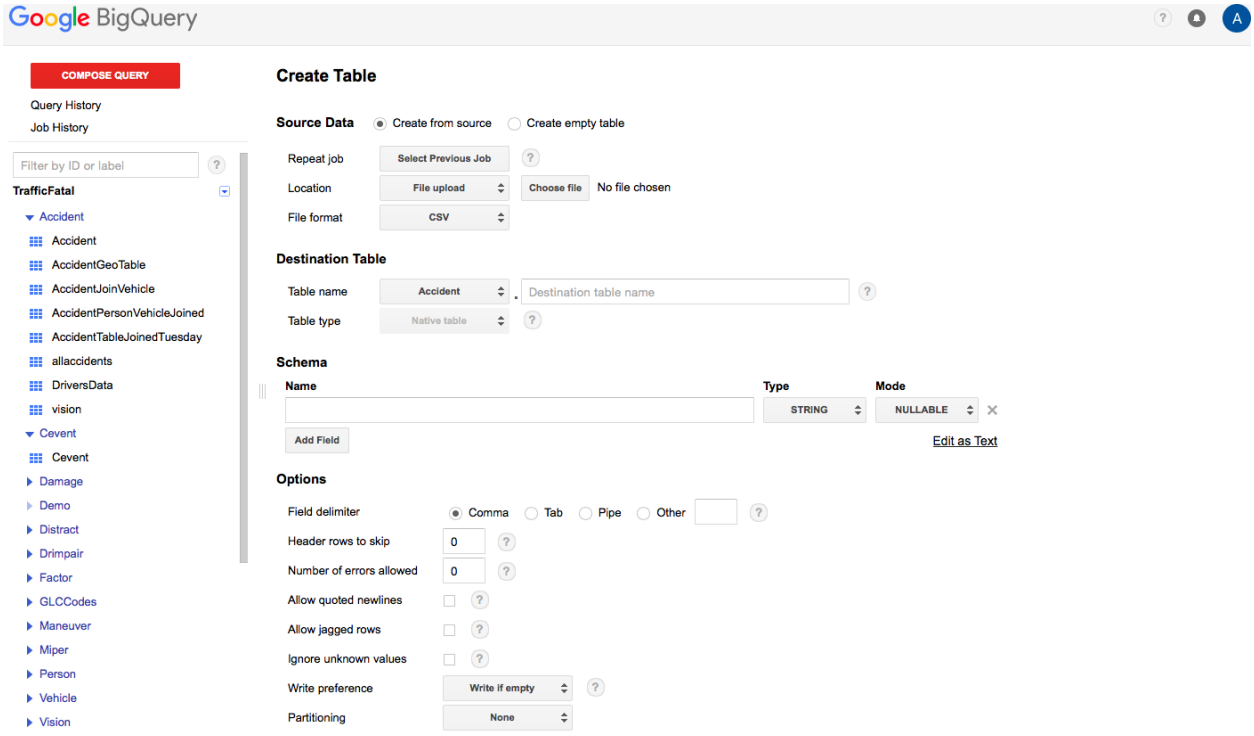


Figure 1:

Once the data is exported in a table, it can be previewed and queried through the 'compose query tab' on the left.

After the data is queried, the resulting dataset can be exported out to the Cloud. This export can be achieved by first creating a bucket from the Cloud console. Buckets can be created by selecting the Storage option

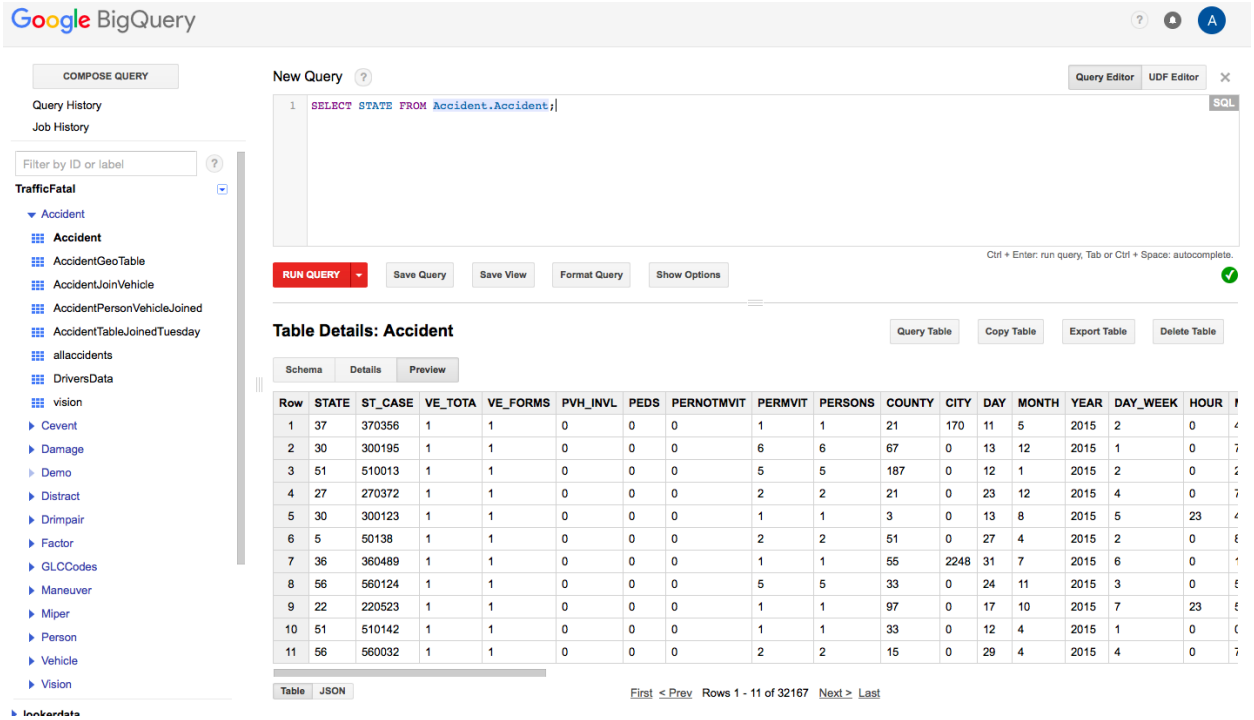


Figure 2:

from the tabbed main Cloud Platform console page. After a bucket is created, the queried data can be exported to that bucket with file name and format specified.

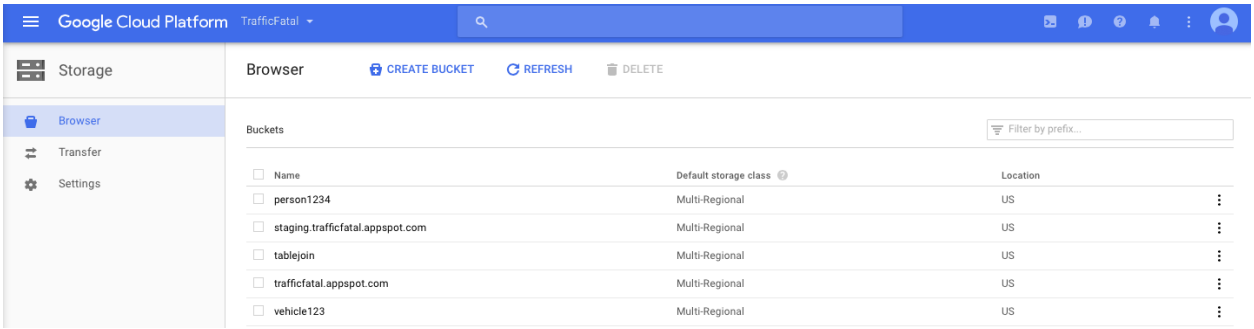


Figure 3:

BigQuery has additional features worth highlighting like the publicly available datasets, which include the National Oceanic and Atmospheric Administration (NOAA) global data obtained from the USAF Climatology Center, US Disease Surveillance data from the CDC, NYC Taxi and Limousine Commission (TLC) Trip Data and GDELT Internet Archive Book Data.

FARS/ACS datasets

The FARS dataset was cleaned to remove any observations with unreported sex, drug use and death scence status. A FIPS code column was created and used as the basis for joining the FARS and the ACS datasets so that each driver is matched with the county and state aggregate population and income to poverty ratio measures. We also created weekday and day status predictors. The data given from FARS already had

information about the day of the week the accident occurred as well as the time. When dayweek, the variable for day of the week equaled 1 it meant Sunday and when it equaled 7 it meant Saturday so the weekend variable was coded simply to be a binary indicator of weekend using the day of week variable. We coded night for if the accident occurred between the hours of 6pm and 6am and day otherwise. Final predictors of interest as a measure of driver's drunk incidence included the driver's sex (Sex), age (Age), history of previous DWI convictions, previous speeding convictions and suspensions and revocations (PrevDWIConvictions, PrevSpeeding, PrevSuspensions), driver's police reported drug use and death scene status (ReportedDrugs, DeathSceneStatus), vehicle level attributes like speed before crash (VehicleSpeed), accident level atributes like number of fatalities, weekday status and say status (NumFatalities, WeekdayStatus, DayStatus) and county level variables like total population and income to poverty ratio in the past 12 months of the driver's state and county (TotalPopulation, IncomeToPovRatio).

Results/Analysis

The dataset was then split in test and training sets with a 70/30 split (70% data in training and 30% in test set). Logistic regression was used first to assess whether drunk driving incidence can be predicted by the above mentioned predictors followed by random forest. Both models were built on the training data and then assessed on the test data via accuracy and ROC curves.

Logistic Regression

Predictors associated with increased drunk driving incidence Predictors accounting for the driver's previous suspensions and revocations, previous DWI convictions, previous speeding convictions, reported drug use, vehicle speed, death enroute to a medical facility, weekend, nighttime and total population all increased the odds of driver being drunk.

The coefficient for PrevSuspensions indicates that holding other variables constant, the odds of a driver involved in a fatal accident being drunk increase by 1.07 given an increase in the previous suspension record by one. Coefficient for PreDWIConvictions indicates that an increase in the previous DWI convictions by one produces an increase in the odds of the driver being drunk by about 1.05 holding other variables fixed. In regards to police reported drug involvement, the odds of a driver being drunk given police reported drug use by the driver is about 2.32 higher than the odds of the driver being drunk given no police reported drug use.

The odds of a driver being drunk are about 1.00 higher given a one unit increase in the vehicle speed holding other variables constant. There is about a 1.45 higher odds of a driver being drunk if the driver dies enroute to a medical facility than if the driver dies at scene of accident. If the day of crash is a weekend, then there is about a 1.93 odds of a drunk driver incidence in comparison to is the crash occurred on a weekday. If the time occurred during night, then there is about a 5.08 odds of the driver being drunk than if the crash occurred during the day. Lastly the odds of a drunk driver increase by about 1.00 given a one unit increase in the total population holding other variables fixed.

Predictors associated with decreased drunk driving incidence Predictors like the driver's sex, age, history of previous speeding suspensions, number of fatalities and income to poverty ratio all decreased the odds of a driver being drunk.

The coefficient for Sex indicates that holding other variables constant, if the driver is female, then there is about a 0.64 odds of a driver being drunk than if the driver was a male. Age indicates that the odds of a driver being drunk are about 0.98 higher given a one year increase in age. Variable for PrevSpeeding indicates that holding other variables constant, the odds of a driver being drunk are 0.90 higher given an increase in previous speeding suspensions of one holding other variables constant. In regards to NumFatalities, the odds of a driver being drunk are about 0.79 higher iwith an increase in the number of fatalities by one. Lastly an increase in income to poverty ratio of one is associated with a 0.99 increased chance of odds of the driver being drunk.

Logistic Regression Assessment Logistic regression was assessed with model accuracy. Initial model accuracy was about 0.76 and then cross validation with k = 100 (data is iteratively split in training and test sets 100 times and accuracy is computed and averaged) was used. The accuracy from cross validation was about 0.72.

Random Forest

Variable Importance Random forest model was also built in addition to logistic regression. Random forest is an ensemble modeling approach where output from many bootstrapped decision trees is aggregated to provide an estimate. About 100 trees were grown for the random forest model and the same predictors were regressed to predict drunk driving incidence as used for the logistic regression (Sex, Age, PrevDWIConvictions, PrevSpeeding, PrevSuspensions, ReportedDrugs, DeathSceneStatus, VehicleSpeed, NumFatalities, WeekdayStatus, DayStatus, TotalPopulation, IncomeToPovRatio).

Variable importance plot was constructed to assess which predictors are most important in predicting drunk driving incidence. Mean decrease in accuracy and mean decrease in gini index (measure of the model misclassification rate) was used as a criterion for importance. Day Status was the most important predictor according to the mean decrease accuracy index which means that exclusion of the day status predictor from the model would result in the highest decrease in accuracy. In comparison, age was the most important predictor according to the mean decrease gini index indicating that age results in the highest node purity, which signifies a low rate that a driver was mistakenly classified as drunk. Additionally IncomeToPovRatio and TotalPopulation were the next important predictors according to both the mean decrease accuracy and the mea decrease gini indices.

Least important predictors in predicting the drunk driving incident included the driver's Sex, DeathSceneStatus and NumFatalities according to both mean decrease accuracy and mean decrease gini indices. Additionally driver's previous speeding and DWI convictions also do not appear to be important in predicting drunk driving incidence.

Random Forest Assessment Random Forest was also assessed via accuracy. The accuracy for the initial test and training split was 0.74. Cross-validation was then used to validate accuracy. K value of 100 was specified in cross-validation, indicating that the data was split in a test and training set 100 times and accuracy was calculated each time, producing an aggregate estimate of accuracy of about 0.73.

Discussion

According to the results above, accident level attributes like the day status of crash and county level predictors like income to poverty ratio in the past 12 months and total population appear to be the most important predictors in predicting whether or not the driver is drunk. The day status predictor makes intuitive sense since people usually indulge in alcohol consumption at night, which could result in a higher incidence of drunk driving during the night. In regards to county level predictors, a more affluent county as indicated by a high income to poverty ratio may be more able to purchase and consume alcohol. High total population may be a confounding variable since there might naturally be an increase in the number of accidents as the population increases, irrelevant to the driver's drinking status. Age also appears to be an important predictor in drunk driving incidence such that an increase in age appears to decrease the odds of a driving being drunk. This result makes sense since older adults may engage less in risky behaviors like drunk driving. Police reported drug consumption is also a predictor of drunk driving since one may expect that individuals who consume alcohol and drunk drive may also pursue other risky behaviors like drug consumption.

Driver's death scene status (whether the driver died at scene or enroute to a medical facility) does not seem to be a relevant predictor in predicting drunk driving incidence which indicates that the severity of the driver's injury as measured by the DeathSceneStatus predictor, does not appear to be correlated with the amount of alcohol consumed. Driver's sex also appears to be irrelevant in predicting drunk driving incidence. Number

of fatalities is not an important variable indicating that the intensity of the crash as measured by the number of deaths in an accident does not predict the drunk driving incidence. Similalry previous speeding convictions and DWI convictions does not appear to be highly relevant indicating that individuals are not affected by their past actions and function in an unpredictable manner.

```
data1 <- read.csv("DriversData.csv") #only data for the drivers selected

#should get around 30000 observations: if more than person involved in an accident

small <- data1 %>% select(V_V_CONFIG, V_TRAV_SP, V_DEFORMED, V_DR_DRINK, V_PREV_ACC,
                                V_PREV_SUS, V_PREV_DWI, V_PREV_SPD, V_VALIGN, V_VPAVETYP, V_VSURCOND,
                                A_LGT_COND, A_FATALS, A_DRUNK_DR, A_WEATHER, A_WEATHER1, A_WEATHER2,
                                A_WRK_ZONE, A_MAN_COLL, P_AGE, P_SEX, P_INJ_SEV, P_SEAT_POS,
                                P_AIR_BAG, P_EJECTION, P_EJ_PATH, P_DRINKING, P_DRUGS,
                                P_LAG_HRS, P_DOA, A_COUNTY, A_STATE, A_CITY, A_DAY_WEEK, A_HOUR)

filter <- unique(small)
small <- filter

#Counting the county frequency by fips code

small$A_STATE <- as.numeric(small$A_STATE)
small$A_COUNTY <- as.numeric(small$A_COUNTY)
small <- small %>% mutate(FIPSCode = ((1000*A_STATE) + A_COUNTY)) #fips code


countiesL <- tally(small$FIPSCode) #tally by FIPSCode
data4 <- data.frame(countiesL)
data6 <- rename(data4, FIPSCode = X)
data7 <- merge(data6, small)
data7 <- unique(data7)

#create new variable here: only counties
data8 <- data7 %>%
  filter(Freq > 10)

data8$P_SEX = as.factor(data8$P_SEX)
data8$P_DRUGS = as.factor(data8$P_DRUGS)
data8$P_DOA = as.factor(data8$P_DOA)
data8$V_DR_DRINK = as.factor(data8$V_DR_DRINK)

data8 <- rename(data8, state = A_STATE)
data8 <- rename(data8, county = A_COUNTY)
data8$state = as.factor(data8$state)
data8$county = as.factor(data8$county)

data8 <- data8 %>% mutate(TravSpeed = as.numeric(V_TRAV_SP),
                          Age = as.numeric(P_AGE),
                          PDWI = as.numeric(V_PREV_DWI),
                          PSuspension = as.numeric(V_PREV_SUS),
                          PCrash = as.numeric(V_PREV_ACC),
                          PSpeed = as.numeric(V_PREV_SPD))

data9 <- data8
```

```r
ACS <- load("CensusFinalD1.Rda")
CensusFinalD1$FIPSCode <- as.factor(CensusFinalD1$FIPSCode)
FinalMerge2<- CensusFinalD1 %>% right_join(data9, by = "FIPSCode") #40422 observations
```

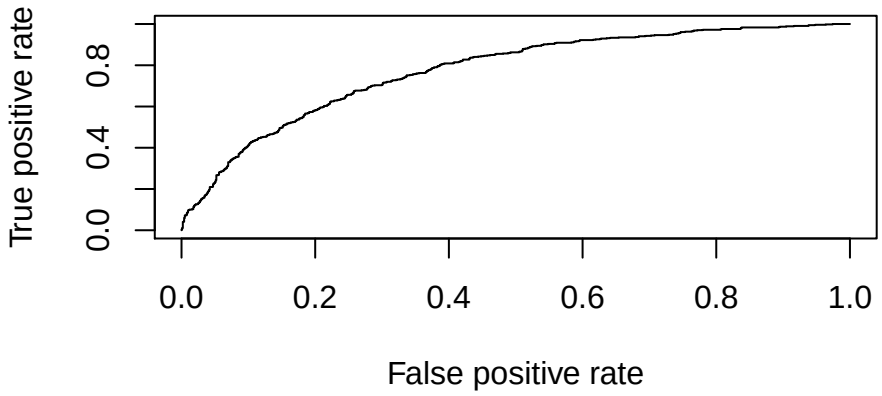
```
## Warning in right_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```r
FinalMerge2 <- na.omit(FinalMerge2)

########Build training and test models here
FinalMerge2 <- rename(FinalMerge2, IncomeToPovRatio = C17002_001) #Quant
FinalMerge2 <- rename(FinalMerge2, TotalPopulation = B01003_001) #Quant
FinalMerge2 <- rename(FinalMerge2, Sex = P_SEX) #Cat
FinalMerge2 <- rename(FinalMerge2, PrevDWIConvictions = PDWI) #Quantitative
FinalMerge2 <- rename(FinalMerge2, PrevSpeeding = PSpeed) #Quantitative
FinalMerge2 <- rename(FinalMerge2, PrevSuspensions = PSuspension) #Quant
FinalMerge2 <- rename(FinalMerge2, VehicleSpeed = TravSpeed) #Quant
FinalMerge2 <- rename(FinalMerge2, ReportedDrugs = P_DRUGS) #Cat
FinalMerge2 <- rename(FinalMerge2, NumFatalities = A_FATALS) #Quant
FinalMerge2 <- rename(FinalMerge2, DeathSceneStatus = P_DOA) #Cat
FinalMerge2 <- rename(FinalMerge2, DriverDrinking = V_DR_DRINK) #Cat

FinalMerge2 <- FinalMerge2 %>% filter(Sex == '1' | Sex == '2')
FinalMerge2$Sex <- droplevels(FinalMerge2$Sex)
levels(FinalMerge2$Sex) <- c("Male", "Female")

FinalMerge2 <- FinalMerge2 %>% filter(ReportedDrugs == '0' | ReportedDrugs == '1')
FinalMerge2$ReportedDrugs <- droplevels(FinalMerge2$ReportedDrugs)
levels(FinalMerge2$ReportedDrugs) <- c("No", "Yes")

FinalMerge2 <- FinalMerge2 %>% filter(DeathSceneStatus == '7' | DeathSceneStatus == '8')
FinalMerge2$DeathSceneStatus <- droplevels(FinalMerge2$DeathSceneStatus)
levels(FinalMerge2$DeathSceneStatus) <- c("DiedAtScence", "DiedAtEnroute")

#Creating weekend/weekday predictor
FinalMerge2 <- FinalMerge2 %>%
  mutate(WeekdayStatus = ifelse(A_DAY_WEEK == '1' | A_DAY_WEEK == '7', "Weekend", "Weekday"))
FinalMerge2$WeekdayStatus <- as.factor(FinalMerge2$WeekdayStatus)

FinalMerge2 <- FinalMerge2 %>%
  mutate(DayStatus = ifelse(A_HOUR < 6 | A_HOUR > 18, "Night", "Day"))
FinalMerge2$DayStatus <- as.factor(FinalMerge2$DayStatus)

test <- FinalMerge2 %>% select(A_HOUR, DayStatus, A_DAY_WEEK, WeekdayStatus) #6908 observations in tota


#Creating training and test results
mergeData <- FinalMerge2
n <- nrow(mergeData)
shuffled <- mergeData[sample(n),]
train <- shuffled[1:round(0.7 * n),]
test <- shuffled[(round(0.7 * n) + 1):n,]
```

```
logmod <- glm(formula = DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions + PrevSpeeding
              + ReportedDrugs + VehicleSpeed + DeathSceneStatus
              + NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio + TotalPopulation,
              family=binomial(link='logit'), data = train)

logTable <- tidy(logmod)
logTable <- logTable %>% mutate(ExpEstimate = exp(estimate))
logTable <- logTable %>% dplyr::select(term, estimate, ExpEstimate, std.error, p.value)
logTable <- rename(logTable, Estimate = estimate)
logTable <- rename(logTable, Term = term)
logTable <- rename(logTable, StdError = std.error)
logTable <- rename(logTable, PValue = p.value)
logTable  #added exponentiated estimate column in the model summary
```

```
##                            Term      Estimate ExpEstimate     StdError
## 1                   (Intercept) -7.253758e-01   0.4841426 1.692989e-01
## 2                     SexFemale -4.785086e-01   0.6197070 9.826630e-02
## 3                           Age -1.773228e-02   0.9824240 2.421277e-03
## 4               PrevSuspensions  3.475621e-02   1.0353673 1.668976e-02
## 5            PrevDWIConvictions  1.339228e-01   1.1433045 4.754336e-02
## 6                 PrevSpeeding -1.714357e-01   0.8424544 4.641545e-02
## 7               ReportedDrugsYes  7.884288e-01   2.1999372 9.988071e-02
## 8                   VehicleSpeed  3.517380e-05   1.0000352 8.434746e-05
## 9  DeathSceneStatusDiedAtEnroute  1.425594e-01   1.1532216 2.875946e-01
## 10                NumFatalities -2.493007e-01   0.7793456 7.952486e-02
## 11         WeekdayStatusWeekend  6.259414e-01   1.8700056 8.092152e-02
## 12               DayStatusNight  1.615117e+00   5.0284746 8.032345e-02
## 13             IncomeToPovRatio -9.140000e-06   0.9999909 6.624783e-06
## 14              TotalPopulation  9.036115e-06   1.0000090 6.529543e-06
##          PValue
## 1  1.830786e-05
## 2  1.118764e-06
## 3  2.415415e-13
## 4  3.729801e-02
## 5  4.849641e-03
## 6  2.211836e-04
## 7  2.933460e-15
## 8  6.766706e-01
## 9  6.201091e-01
## 10 1.719259e-03
## 11 1.032676e-14
## 12 6.323792e-90
## 13 1.676889e-01
## 14 1.663947e-01
```

```
prob <- predict(logmod, newdata=test, type="response")
pred <- prediction(prob, test$DriverDrinking)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf)
```




```
#Accuracy assessment
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc #75% accurate
```

```
## [1] 0.7745816
```

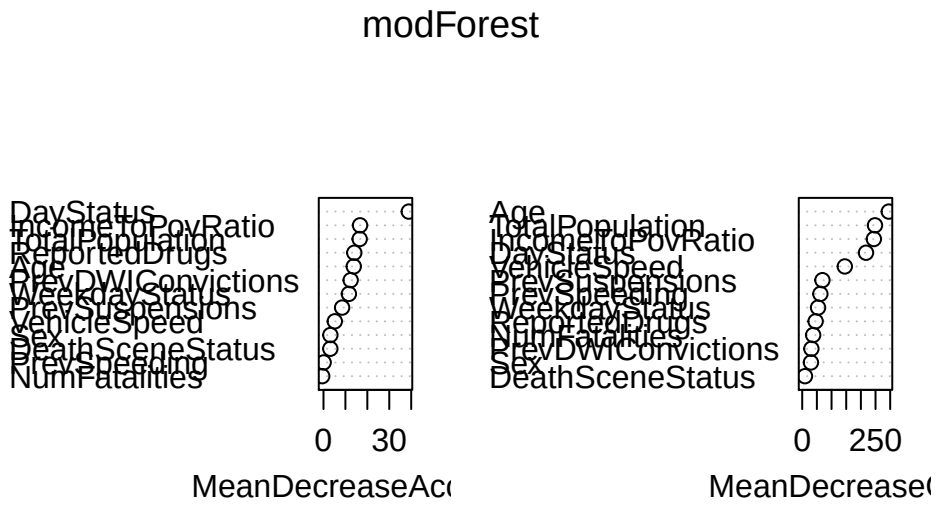
```
#Cross-validation for model assessment
dat <- mergeData
k <- 100
acc <- NULL
set.seed(123)
for(i in 1:k)
{
  smp_size <- floor(0.70 * nrow(dat))
  index <- sample(seq_len(nrow(dat)),size=smp_size)
  train <- dat[index, ]
  test <- dat[-index, ]
  model <- glm(formula = DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions + PrevSpeedi
              + ReportedDrugs + VehicleSpeed + DeathSceneStatus
              + NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio + TotalPopulation,
              family=binomial(link='logit'), data = train)
  results_prob <- predict(model,newdata=test,type='response')
  results <- ifelse(results_prob > 0.5,1,0)
  answers <- test$DriverDrinking
  misClasificError <- mean(answers != results)
  acc[i] <- 1-misClasificError
}

mean(acc)
```

```
## [1] 0.7237058
```

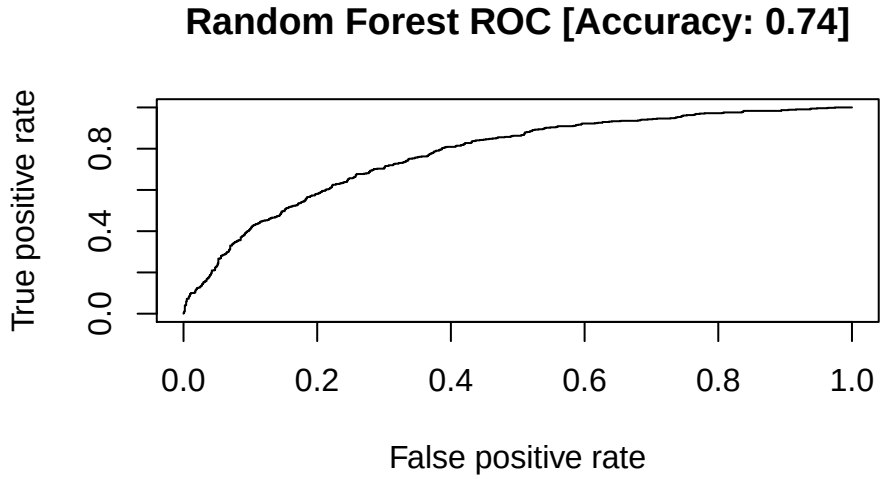
Random Forest

```
set.seed(1000)
modForest <- randomForest(DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions +
```

```
                          PrevSpeeding + ReportedDrugs + VehicleSpeed + DeathSceneStatus +
                          NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio +
                          TotalPopulation, data = train, ntree = 100, mtry = 4,
                        keep.forest = FALSE, importance = TRUE)
#Importance plots
varImpPlot(modForest)
```




```
#Assessing model accuracy
modForest1 <- randomForest(DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions +
                           PrevSpeeding + ReportedDrugs + VehicleSpeed + DeathSceneStatus +
                           NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio +
                           TotalPopulation, data = train, ntree = 100, mtry = 4,
                         keep.forest = TRUE, importance = FALSE)

test.forest = predict(modForest1, type = "prob", newdata = test)
forestpred = prediction(test.forest[,2], test$DriverDrinking)
forestperf = performance(forestpred, "tpr", "fpr")
plot(perf, main="Random Forest ROC [Accuracy: 0.74]")
```



```
conf <- modForest1$confusion
print(sum(diag(conf)) / sum(conf))
```

```
## [1] 0.7310047
```

```
dat <- mergeData
k <- 2
acc2 <- NULL
set.seed(123)
for(i in 1:k)
{
  smp_size <- floor(0.70 * nrow(dat))
  index <- sample(seq_len(nrow(dat)),size=smp_size)
  train <- dat[index, ]
  test <- dat[-index, ]
  modForest1 <- randomForest(DriverDrinking ~ Sex + Age + PrevSuspensions + PrevDWIConvictions +
                             PrevSpeeding + ReportedDrugs + VehicleSpeed + DeathSceneStatus +
                             NumFatalities + WeekdayStatus + DayStatus + IncomeToPovRatio +
                             TotalPopulation, data = train, ntree = 100, mtry = 4,
                           keep.forest = TRUE, importance = FALSE)
  conf <- modForest1$confusion
  acc2[i] <- (sum(diag(conf)) / sum(conf))
}
mean(acc2)
```

```
## [1] 0.7363795
```

Check conditions for logistic regression

Include screenshots to BigQuery

Change cross validation to 100 above