Uber Take Home

Jeffrey Uslan

November 1, 2015

library(dplyr)

##   
## Attaching package: 'dplyr'  
##   
## The following object is masked from 'package:stats':  
##   
## filter  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(lubridate)  
library(rpart)  
library(randomForest)

## randomForest 4.6-10  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(zoo)

##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(caret)

## Loading required package: lattice  
## Loading required package: ggplot2

library(pander)

load(file="uber\_test\_data.rda")

data=tbl\_df(uber\_unlist)  
data$trips\_in\_first\_30\_days=as.numeric(as.character(data$trips\_in\_first\_30\_days))  
data$signup\_date=as.Date(data$signup\_date)  
data$avg\_rating\_of\_driver=as.numeric(as.character(data$avg\_rating\_of\_driver))  
data$avg\_surge=as.numeric(as.character(data$avg\_surge))  
data$last\_trip\_date=as.Date(data$last\_trip\_date)  
data$surge\_pct=as.numeric(as.character(data$surge\_pct))  
data$weekday\_pct=as.numeric(as.character(data$weekday\_pct))  
data$avg\_dist=as.numeric(as.character(data$avg\_dist))  
data$avg\_rating\_by\_driver=as.numeric(as.character(data$avg\_rating\_by\_driver))

numeric\_data\_inds=sapply(data,is.numeric)  
cat\_data\_inds=!sapply(data,is.numeric)  
  
#tabulations of categorical data  
pander(data %>% group\_by(city) %>% summarise(Count=n()))

|  |  |
| --- | --- |
| city | Count |
| King's Landing | 10130 |
| Astapor | 16534 |
| Winterfell | 23336 |

pander(data %>% group\_by(phone) %>% summarise(Count=n()))

|  |  |
| --- | --- |
| phone | Count |
| iPhone | 34582 |
| Android | 15022 |
| NA | 396 |

pander(data %>% group\_by(uber\_black\_user) %>% summarise(Count=n()))

|  |  |
| --- | --- |
| uber\_black\_user | Count |
| TRUE | 18854 |
| FALSE | 31146 |

#generating retained variable  
data$retained=0  
data$retained[which(data$trips\_in\_first\_30\_days>0)]=1  
mean(data$retained,na.rm=TRUE)

## [1] 0.6922

data$retained=as.factor(data$retained)  
data %>% group\_by(retained) %>% summarise(Count=n())

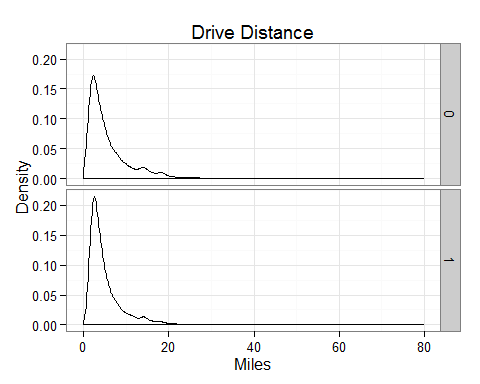
## Source: local data frame [2 x 2]  
##   
## retained Count  
## 1 0 15390  
## 2 1 34610

# numeric exploration

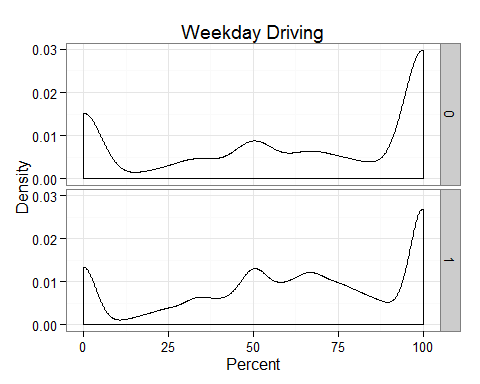
data=na.omit(data)  
  
  
  
pander(data %>% group\_by(retained) %>% summarise(Distance=mean(avg\_dist),Weekday=mean(weekday\_pct), Passenger=mean(avg\_rating\_by\_driver),Driver=mean(avg\_rating\_of\_driver),  
 Surges=mean(avg\_surge)))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| retained | Distance | Weekday | Passenger | Driver | Surges |
| 0 | 6.005 | 61.72 | 4.79 | 4.634 | 1.07 |
| 1 | 5.25 | 61.08 | 4.771 | 4.589 | 1.075 |

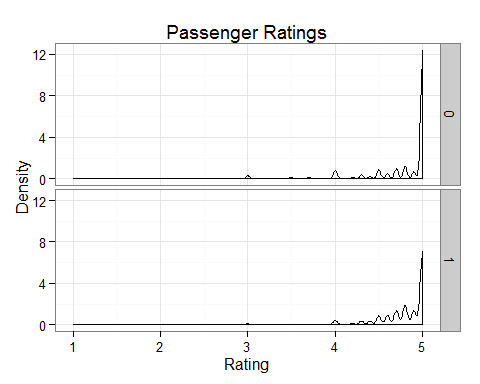
ggplot(data)+geom\_density(aes(x=avg\_dist))+theme\_bw()+facet\_grid(retained~.)+  
 ggtitle("Drive Distance")+xlab("Miles")+ylab("Density")



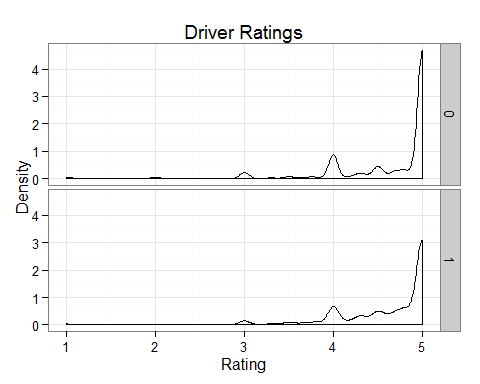
ggplot(data)+geom\_density(aes(x=weekday\_pct))+theme\_bw()+facet\_grid(retained~.)+  
 ggtitle("Weekday Driving")+xlab("Percent")+ylab("Density")



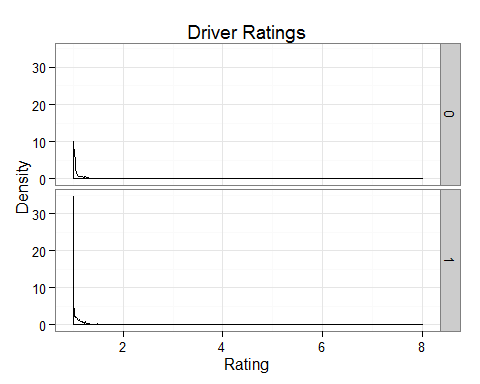
ggplot(data)+geom\_density(aes(x=avg\_rating\_by\_driver))+ theme\_bw()+facet\_grid(retained~.)+ggtitle("Passenger Ratings")+xlab("Rating")+ylab("Density")



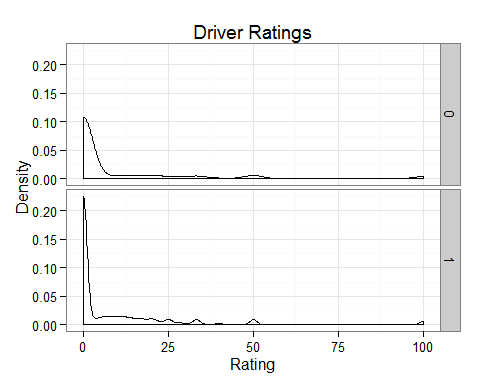
ggplot(data)+geom\_density(aes(x=avg\_rating\_of\_driver))+ theme\_bw()+facet\_grid(retained~.)+ggtitle("Driver Ratings")+xlab("Rating")+ylab("Density")



ggplot(data)+geom\_density(aes(x=avg\_surge))+   
 theme\_bw()+facet\_grid(retained~.)+ggtitle("Driver Ratings")+xlab("Rating")+ylab("Density")



ggplot(data)+geom\_density(aes(x=surge\_pct))+   
 theme\_bw()+facet\_grid(retained~.)+ggtitle("Driver Ratings")+xlab("Rating")+ylab("Density")



# covariates

inTrain = createDataPartition(y=data$retained, p = .6)[[1]]  
training = data[ inTrain,]  
testing = data[-inTrain,]  
  
modFit <- randomForest(retained ~city + phone +uber\_black\_user + avg\_rating\_of\_driver +  
 avg\_surge+ surge\_pct + weekday\_pct +avg\_dist +avg\_rating\_by\_driver ,data=training)  
  
  
train\_pred=predict(modFit,newdata= training, type = "response")  
confusionMatrix(training$retained,train\_pred )

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 2674 4241  
## 1 60 17893  
##   
## Accuracy : 0.827   
## 95% CI : (0.8223, 0.8317)  
## No Information Rate : 0.8901   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.4709   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9781   
## Specificity : 0.8084   
## Pos Pred Value : 0.3867   
## Neg Pred Value : 0.9967   
## Prevalence : 0.1099   
## Detection Rate : 0.1075   
## Detection Prevalence : 0.2781   
## Balanced Accuracy : 0.8932   
##   
## 'Positive' Class : 0   
##

test\_pred=predict(modFit,newdata= testing, type = "response")  
confusionMatrix(testing$retained,test\_pred)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 236 4373  
## 1 306 11662  
##   
## Accuracy : 0.7177   
## 95% CI : (0.7108, 0.7246)  
## No Information Rate : 0.9673   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0352   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.43542   
## Specificity : 0.72728   
## Pos Pred Value : 0.05120   
## Neg Pred Value : 0.97443   
## Prevalence : 0.03270   
## Detection Rate : 0.01424   
## Detection Prevalence : 0.27804   
## Balanced Accuracy : 0.58135   
##   
## 'Positive' Class : 0   
##

varImpPlot(modFit,main="Variable Importance")

