

Mva_Project7.R

thindprateek

2020-11-12

```
##### Applying Linear Discriminant Analysis #####
```

```
#Getting working directory  
getwd()
```

```
## [1] "/Users/thindprateek/Desktop/Multivariate Analysis"
```

```
#Setting directory to load data set  
setwd("/Users/thindprateek/Desktop/Multivariate Analysis")
```

```
#Reading the data into a data frame  
#df <- read.csv(file = 'US_Acc_June20.csv')  
num <- read.csv(file = 'num.csv')
```

```
# Performing clustering on the first 500 records for now to achieve easy and quick results and test the  
attach(num)  
# Printing first few columns of data set for inference  
#head(df)
```

```
## Setting random seed to shuffle data before splitting  
set.seed(23)
```

```
#Checking number of rows  
#rows<-sample(nrow(df))
```

```
#Shuffling the data  
#mva<-df[rows,]
```

```
#Taking the required number of instances from the shuffled data to reduce any biases  
#mva<-mva[950000:1000000,]
```

```
#Checking the structure of the data set  
#str(mva)
```

```
# Checking the number of rows and columns in the current uncleaned dataset  
#ncol(mva)  
#nrow(mva)
```

```
# Printing all the column names to find and filter the relevant and irrelevant attributes  
#names<-names(mva)  
#names
```

```
## DATA CLEANING ##

#Dropping the surplus attributes which do not contribute to the analysis
#mva <- mva[-c(1:3,7:10,13,14,19,21:23,33,47:49)]

#Checking for any null values in the present data set
# is.na(mva[,])

#Checking which rows have all the values filled and complete
# complete.cases(mva)

#Making a new dataframe with only the rows that have complete information and all values filled
#Mva<-na.omit(mva)
#Mva<-Mva[!(is.na(Mva$Sunrise_Sunset) | Mva$Sunrise_Sunset==""), ]
#Mva<- Mva[complete.cases(Mva),]
#Verifying for missing values in the new dataframe
#complete.cases(Mva)
#unique(Mva$Sunrise_Sunset)

# Creating new dataframe with only the numerical attributes to perform statistical functions
#num<-Mva[,c(1,4,11:15,17,18)]
#write.csv(num,"/Users/mihikagupta/Desktop/SEM_2/MVA/num.csv", row.names = FALSE)

# Scaling the new data set for better accuracies
# num<-scale(num)

# Checking the dimensions of the data
nrow(num)
```

```
## [1] 18250
```

```
ncol(num)
```

```
## [1] 9
```

```
names(num)
```

```
## [1] "Severity"           "Distance.mi."       "Temperature.F."
## [4] "Wind_Chill.F."     "Humidity..."      "Pressure.in."
## [7] "Visibility.mi."    "Wind_Speed.mph."    "Precipitation.in."
```

```
dim(num)
```

```
## [1] 18250      9
```

```
names(num)[names(num) == "Distance.mi."] <- "dist"
names(num)[names(num) == "Temperature.F."] <- "temp"
names(num)[names(num) == "Wind_Chill.F."] <- "windchill"
names(num)[names(num) == "Humidity..."] <- "humidity"
names(num)[names(num) == "Pressure.in."] <- "pressure"
names(num)[names(num) == "Visibility.mi."] <- "visibility"
```

```
names(num)[names(num) == "Wind_Speed.mph."] <- "windspeed"
names(num)[names(num) == "Precipitation.in."] <- "precip"
names(num)
```

```
## [1] "Severity" "dist" "temp" "windchill" "humidity"
## [6] "pressure" "visibility" "windspeed" "precip"
```

```
num$Severity<- factor(num$Severity)
str(num)
```

```
## 'data.frame': 18250 obs. of 9 variables:
## $ Severity : Factor w/ 4 levels "1","2","3","4": 2 2 3 2 2 2 2 3 2 3 ...
## $ dist : num 0 0 0 0 0 ...
## $ temp : num 78 96 89 68 53 37 8 78 40 46 ...
## $ windchill : num 78 96 89 68 53 30 -4 78 40 42 ...
## $ humidity : int 58 33 59 88 59 96 58 54 88 44 ...
## $ pressure : num 29.2 29.2 30 29.4 29.5 ...
## $ visibility: num 10 10 10 6 10 2 10 10 10 10 ...
## $ windspeed : num 12 7 6 5 12 10 8 12 3 8 ...
## $ precip : num 0 0 0 0.04 0 0.02 0 0 0 0 ...
```

```
library(MASS)
library(ggplot2)
# Lets cut the data into two parts
smp_size_raw <- floor(0.75 * nrow(num))
train_ind_raw <- sample(nrow(num), size = smp_size_raw)
train_raw.df <- as.data.frame(num[train_ind_raw, ])
test_raw.df <- as.data.frame(num[-train_ind_raw, ])
str(train_raw.df)
```

```
## 'data.frame': 13687 obs. of 9 variables:
## $ Severity : Factor w/ 4 levels "1","2","3","4": 2 2 2 2 2 2 2 3 2 2 ...
## $ dist : num 0 0.405 0 0.152 0 0 0 0 0 0 ...
## $ temp : num 98 80 50 65 93 62 45 54 84 57 ...
## $ windchill : num 98 80 50 65 93 62 37 54 84 57 ...
## $ humidity : int 28 50 71 56 42 70 90 28 63 94 ...
## $ pressure : num 29.1 29.7 30 29.7 29.2 ...
## $ visibility: num 10 10 10 10 10 10 10 10 10 7 ...
## $ windspeed : num 9 3 0 16 22 3 18 9 14 0 ...
## $ precip : num 0 0 0 0 0 0 0 0 0 0.02 ...
```

```
# We now have a training and a test set. Training is 75% and test is 25%
num.lda <- lda(formula = train_raw.df$Severity ~ ., data = train_raw.df)
#Prior probability is high for Severity 2 and then Severity 3
#Precipitation seems to be the most important independent variable followed by distance
summary(num.lda)
```

```
##      Length Class  Mode
## prior      4  -none- numeric
## counts     4  -none- numeric
```

```
## means 32 -none- numeric
## scaling 24 -none- numeric
## lev 4 -none- character
## svd 3 -none- numeric
## N 1 -none- numeric
## call 3 -none- call
## terms 3 terms call
## xlevels 0 -none- list
```

```
num.lda$counts
```

```
##      1      2      3      4
## 312 9767 3144 464
```

```
#Severity 2 and 3 have most of the distribution
```

```
num.lda$means
```

```
##      dist      temp windchill humidity pressure visibility windspeed
## 1 0.1923013 70.29135 69.98205 52.07372 29.10593 9.505769 8.452564
## 2 0.1341171 60.63874 59.23024 65.47466 29.30508 8.913619 7.315491
## 3 0.5084113 61.14205 59.45506 65.87214 29.28253 8.827020 8.165522
## 4 1.9723297 59.33341 57.27996 67.70259 29.13328 8.690625 7.919612
##      precip
## 1 0.002916667
## 2 0.006346882
## 3 0.012051527
## 4 0.007866379
```

```
num.lda$scaling
```

```
##      LD1      LD2      LD3
## dist 0.643362227 0.02488782 0.16025644
## temp 0.090654983 -0.05312258 -0.17813300
## windchill -0.078760570 0.07382747 0.14930871
## humidity 0.007780552 -0.02571859 -0.02086413
## pressure -0.092455612 -0.11695430 -0.09004520
## visibility 0.008178223 -0.05361433 -0.01702252
## windspeed 0.014939005 0.06707019 -0.11789759
## precip 0.427821895 0.19555958 -3.54006661
```

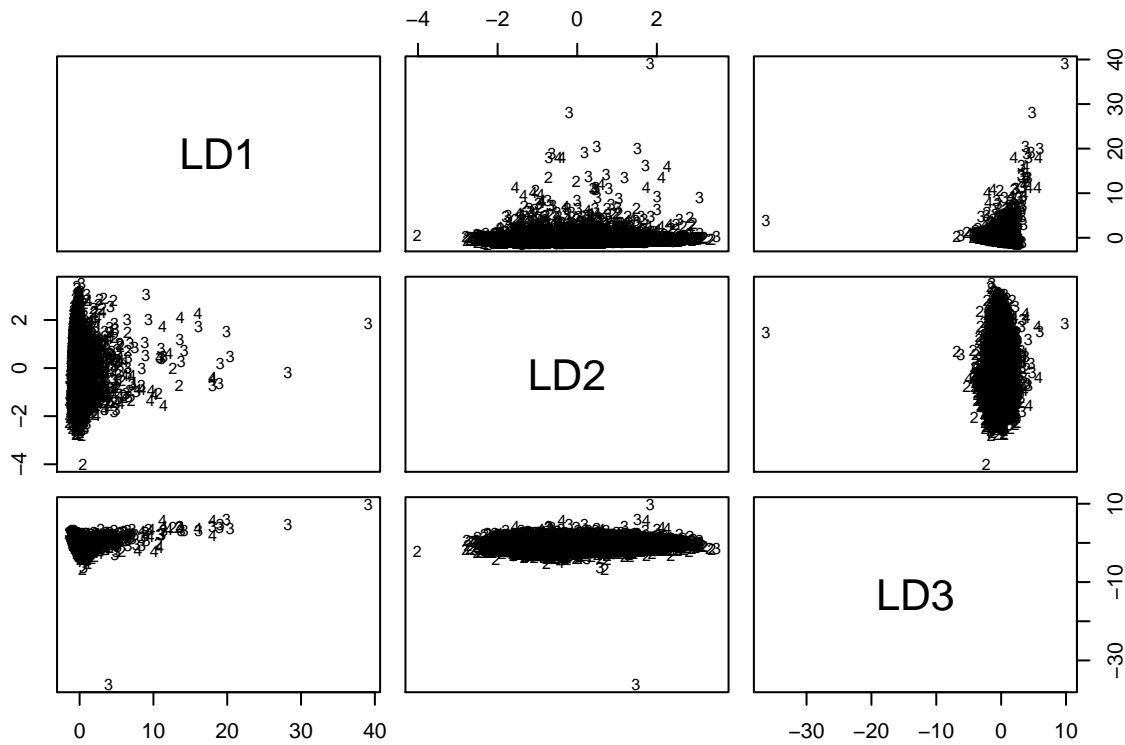
```
num.lda$prior
```

```
##      1      2      3      4
## 0.02279535 0.71359684 0.22970702 0.03390078
```

```
num$lev
```

```
## NULL
```

```
plot(num.lda)
```



```
num.lda.predict <- predict(num.lda)
num.lda.predict$class
```

[illegible]

[illegible]

##	[4393]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4429]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4465]	2	2	2	2	2	2	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4501]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4537]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4573]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4609]	2	2	2	2	2	2	2	2	2	2	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4645]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4681]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4717]	2	2	2	2	2	2	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4753]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4789]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4825]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4861]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4897]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4933]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[4969]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4	2	2
##	[5005]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[5041]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[5077]	2	2	2	2	2	2	2	2	2	2	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[5113]	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
##	[5149]	2	2	2	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4	2	2	2	2	2			

[illegible]

[illegible]

[illegible]


```
## [4219] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4256] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4293] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4330] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4367] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4404] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4441] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4478] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4515] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4552] 2 2 2 2 2 2 2 2 2 2 2 2
## Levels: 1 2 3 4
```

```
table(num.lda.predict$class, train_raw.df$Severity)
```

```
##
##      1      2      3      4
## 1      0      0      0      0
## 2 310 9721 3070 416
## 3      0      0      1      0
## 4      2     46     73     48
```

```
table(num.lda.predict.test$class, test_raw.df$Severity )
```

```
##
##      1      2      3      4
## 1      0      0      0      0
## 2  72 3227 1065 130
## 3      0      0      0      0
## 4      1     20     29     19
```

```
#you can see through tables, ratios are close to correct prediction
```