

main ▾

...

CIND820_Capstone / Capstone_DataAssessment_EDA.R



kmault Add files via upload

[History](#)

1 contributor

557 lines (421 sloc) | 21.4 KB

...

```
1 #CIND820 Captstone Project 'ASSESSMENT OF TORONTO CRIME DATA THROUGH EXPLORATORY DATA ANALYSIS AND CLASSIFICATION METHODS'
2
3 #Load all the libraries, won't necessarily use them all but the are frequently used
4 library(ggplot2)
5 library(ggmap)
6 library(tidyverse)
7 library(rgdal)
8 library(readxl)
9 library(sf)
10 library(data.table)
11 library(plyr)
12 library(dplyr)
13 library(hablar)
14 library(ggthemes)
15 library(tidyr)
16 library(ggrepel)
17 library(lubridate)
18 library(mapproj)
19 library(RColorBrewer)
20 library(viridisLite)
21
22
23 #Load Toronto Major Crime Indicator dataset
24 crime<-read.csv(file ="D:/Ryerson Big Data/CIND820 Big Data Analytics Project/TorontoCrime/Major_Crime_Indicators.csv", header = T, na.
25
26 #check structure
27 str(crime)
28
29 #summary of data
30 summary(crime)
31
32 #summary of datatypes
33 table(sapply(crime, class))
34
35 #confirm number of complete records
36 sum(complete.cases(crime))
37
38 #check for occurrence dates before 2014 in the MCI
39 sum(crime$occurrenceyear < 2014)
40
41 #find records with the same ID and offense type
42 dupRow <- crime[which(duplicated(crime[,c('event_unique_id', 'offence')])==TRUE),]
43 nrow(dupRow)
44
45 #Start cleaning dataset
```

```

46 #feature reduction - remove redundant variables
47 MCI_cln <- crime[ -c(1:3,7,8,11,13:18,21, 22, 30) ]
48
49 #check occurrence year values
50 range(MCI_cln$occurrenceyear) #can see the column contains occurrences before 2014
51
52 #filter for occurrence dates between 2014 and 2021
53 MCI_cln<-filter(MCI_cln,occurrenceyear=='2014' | occurrenceyear=='2015' | occurrenceyear=='2016' | occurrenceyear=='2017' | occurrenceyear=='2018' | occurrenceyear=='2019' | occurrenceyear=='2020' | occurrenceyear=='2021')
54
55 range(MCI_cln$occurrenceyear) #confirm range
56
57 #remove records with same ID and offense type, rename dataset
58 MCI_cln<-MCI_cln %>%
59   distinct(event_unique_id, offence, .keep_all = TRUE)
60
61
62 #convert selected categorical columns to factor variables
63 MCI_cln$Division<-as.factor(MCI_cln$Division)
64 MCI_cln$premises_type<-as.factor(MCI_cln$premises_type)
65 MCI_cln$offence<-as.factor(MCI_cln$offence)
66 MCI_cln$occurrencemonth<-factor(MCI_cln$occurrencemonth, levels=c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))
67 MCI_cln$occurrencedayofweek = gsub(" ", "", MCI_cln$occurrencedayofweek)
68 MCI_cln$occurrencedayofweek<-factor(MCI_cln$occurrencedayofweek, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
69 MCI_cln$MCI<-as.factor(MCI_cln$MCI)
70 MCI_cln$Neighbourhood<-as.factor(MCI_cln$Neighbourhood)
71
72 #add a new column "Weight" via the matched UCR columns in the MCI and Weights dataframes, then move after ucr
73 MCI_cln$weight <- Weights$Weighting[match(MCI_cln$ucr, Weights$UCR)]
74
75 MCI_cln <- MCI_cln %>% relocate(weight, .before = offence)
76
77 #Add a season column based on months then convert to factor
78 MCI_cln$season <- ifelse(MCI_cln$occurrencemonth %in% c('December','January','February'), "Winter",
79   ifelse (MCI_cln$occurrencemonth %in% c('September','October', 'November'), "Autumn",
80     ifelse (MCI_cln$occurrencemonth %in% c('March','April','May'),
81       "Spring", "Summer")))
82 #Convert Season to a factor
83 MCI_cln$season<-factor(MCI_cln$season, levels=c("Winter", "Spring", "Summer", "Autumn"))
84
85
86 #Figure 1: structure of cleaned Toronto MCI database
87 str(MCI_cln)
88
89
90 #Load Toronto Neighbourhood Profiles and check structure, entries, and variables. Dataset was cumbersome and cleaned in another notebook
91
92 Nhoods<-read.csv(file = "D:/Ryerson Big Data/CIND820 Big Data Analytics Project/TorontoCrime/Neighbourhoods.csv", header = T, na.strings = c("NA", " "))
93
94 #check the structure
95 str(Nhoods)
96
97 #check datatypes
98 table(sapply(Nhoods, class))
99
100 #check for missing values
101 sum(is.na(Nhoods))
102
103 #check for duplicates
104 sum(duplicated(Nhoods$X_id))
105
106 #Transpose database
107 Hoods_cln<-t(Nhoods)
108
109 #Confirm the new file is a dataframe
110 str(Nhoods)

```

```

111
112 exists("Hoods_cln")&&is.data.frame(get("Hoods_cln"))
113
114 #Convert to a data frame & confirm updated structure
115
116 Hoods_cln <- as.data.frame(Hoods_cln)
117
118 exists("Hoods_cln")&&is.data.frame(get("Hoods_cln"))
119
120
121 #read CSI weights file
122 Weights<-read_excel("D:\\Ryerson Big Data\\CIND820 Big Data Analytics Project\\TorontoCrime\\CSI_weights2020.xlsx")
123
124 str(Weights)
125
126
127 #read police divisions shape file
128 Patrols <- st_read(
129   "D:/Ryerson Big Data/CIND820 Big Data Analytics Project/TorontoCrime/ShapeFiles/Police_Divisions.shp")
130
131 str(Patrols)
132
133
134 #Figure 2: Plot of Patrol Zones
135
136 ggplot() +
137   geom_sf(data = Patrols, size = 1, color = "black", fill = "white") +
138   ggtitle("Toronto Police Patrol Zones") +
139   xlab("Longitude") +
140   ylab("Latitude")+
141   coord_sf()
142
143 #read neighbourhoods shape file
144 Neighbourhoods <- st_read(
145   "D:/Ryerson Big Data/CIND820 Big Data Analytics Project/TorontoCrime/ShapeFiles/Neighbourhoods.shp")
146 str(Neighbourhoods)
147 head(Neighbourhoods)
148
149
150 #Figure 3: Plot of Toronto Neighbourhoods
151 ggplot() +
152   geom_sf(data = Neighbourhoods, size = 1, color = "black", fill = "white") +
153   ggtitle("Toronto Neighbourhoods") +
154   xlab("Longitude") +
155   ylab("Latitude")+
156   coord_sf()
157
158
159
160 #EDA & Descriptive Statistics Toronto MCI Dataset
161
162 #Figure 4: incidents per year
163 IncYear<-count(MCI_cln$occurrenceyear)
164 setnames(IncYear, "x", "Year")
165 setnames(IncYear, "freq", "IncidentCounts")
166
167 #Calc avg yearly crime count between 2014 and 2021 for Toronto
168 AvgCrime<-(sum(IncYear$IncidentCounts))/8
169
170 write.table(IncYear, file = "IncYear.txt", sep = ",", quote = FALSE, row.names = F)
171
172 ggplot(data=MCI_cln, aes(MCI_cln$occurrenceyear)) +
173   geom_histogram(binwidth = 1, color="black", fill="blue") + scale_x_continuous(breaks = 2014:2021) +
174   labs(title = "Toronto Crime - Incidents per Year", x = "Year", y = "Frequency") +
175   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),

```

```

176     panel.background = element_blank(), axis.line = element_line(colour = "black"))
177
178
179 #Figure 5: boxplot of incidents per year
180 BP_Yr <-MCI_cln %>% group_by(occurrenceyear) %>%
181   dplyr::summarise(N = n())
182
183 ggplot(BP_Yr, aes(x="", N, y=N)) +
184   geom_boxplot(width=0.6, outlier.size=3,outlier.colour="black", fill = 'blue') +
185   stat_summary(
186     aes(label=sprintf("%1.1f", ..y..)),
187     geom="text",
188     fun = function(y) boxplot.stats(y)$stats,
189     position=position_nudge(x=0.33),
190     size=3.5) +
191   theme_bw() +
192   stat_boxplot(geom = "errorbar", width = 0.5) +
193   xlab("Year") + ylab("Count") +
194   ggtitle("Range of Incident Counts per Year")
195
196
197
198 #Figure 6: incidents per month
199 IncMonth<-count(MCI_cln$occurrencemonth)
200 setnames(IncMonth, "x", "Month")
201 setnames(IncMonth, "freq", "IncidentCounts")
202
203 ggplot(MCI_cln, aes(x = occurrencemonth, fill=occurrencemonth)) +
204   geom_bar(width=0.8, stat="count") + scale_fill_brewer(palette="Set3") +
205   theme(legend.position="none") + scale_x_discrete()+
206   ggtitle("Incidents per Month") + xlab("Month") + ylab("Count") +
207   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
208     panel.background = element_blank(), axis.line = element_line(colour = "black"))
209
210 #Figure 7: boxplot of incidents per month
211 BP_M <-MCI_cln %>% group_by(occurrencemonth) %>%
212   dplyr::summarise(N = n())
213
214
215 ggplot(BP_M, aes(x="", N, y=N)) +
216   geom_boxplot(width=0.6, outlier.size=3,outlier.colour="black", fill = "aquamarine") +
217   stat_summary(
218     aes(label=sprintf("%1.1f", ..y..)),
219     geom="text",
220     fun = function(y) boxplot.stats(y)$stats,
221     position=position_nudge(x=0.33),
222     size=3.5) +
223   theme_bw() +
224   stat_boxplot(geom = "errorbar", width = 0.5) +
225   xlab("Month") + ylab("Count") +
226   ggtitle("Range of Incident Counts per Month")
227
228
229
230 #Figure 8: incidents per day
231 IncDay<-count(MCI_cln$occurrencedayofweek)
232 setnames(IncDay, "x", "Day")
233 setnames(IncDay, "freq", "IncidentCounts")
234
235 ggplot(MCI_cln, aes(x = occurrencedayofweek, fill=occurrencedayofweek)) +
236   geom_bar(width=0.8, stat="count") + scale_fill_brewer(palette="Set3") +
237   theme(legend.position="none") + scale_x_discrete()+
238   ggtitle("Incidents per Day") + xlab("Day") + ylab("Count") +
239   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
240     panel.background = element_blank(), axis.line = element_line(colour = "black"))

```

```

241
242 #Figure 9: boxplot of incidents per day
243 BP_dow <-MCI_cln %>% group_by(occurrencedayofweek) %>%
244   dplyr::summarise(N = n())
245
246
247 ggplot(BP_dow, aes(x="", N, y=N)) +
248   geom_boxplot(width=0.6, outlier.size=3,outlier.colour="black", fill = "cadetblue") +
249   stat_summary(
250     aes(label=sprintf("%1.1f", ..y..)),
251     geom="text",
252     fun = function(y) boxplot.stats(y)$stats,
253     position=position_nudge(x=0.33),
254     size=3.5) +
255   theme_bw() +
256   stat_boxplot(geom = "errorbar", width = 0.5) +
257   xlab("Day") + ylab("Count") +
258   ggtitle("Range of Incident Counts per Day")
259
260
261
262 #Figure 10: incidents per season
263 IncSeason<-count(MCI_cln$season)
264 setnames(IncSeason, "x", "Season")
265 setnames(IncSeason, "freq", "IncidentCounts")
266
267 ggplot(MCI_cln, aes(x = season, fill=season)) +
268   geom_bar(width=0.8, stat="count") + scale_fill_brewer(palette="Set3") +
269   theme(legend.position="none") + scale_x_discrete()+
270   ggtitle("Incidents per Season ") + xlab("Season") + ylab("Count") +
271   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
272     panel.background = element_blank(), axis.line = element_line(colour = "black"))
273
274 #Figure 11: incidents per premises type
275 IncPrem<-count(MCI_cln$premises_type)
276 setnames(IncPrem, "x", "Premises")
277 setnames(IncPrem, "freq", "IncidentCounts")
278
279
280 ggplot(MCI_cln, aes(x = premises_type, fill=premises_type)) +
281   geom_bar(width=0.8, stat="count") + scale_fill_brewer(palette="Set3") +
282   theme(legend.position="none") + scale_x_discrete()+
283   ggtitle("Incidents per Premises Type ") + xlab("Premises Type") + ylab("Count") +
284   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
285     panel.background = element_blank(), axis.line = element_line(colour = "black"))
286
287
288 #Figure 12: incidents per hour
289 IncHour<-count(MCI_cln$occurrencehour)
290 setnames(IncHour, "x", "Hour")
291 setnames(IncHour, "freq", "IncidentCounts")
292
293
294 ggplot(MCI_cln, aes(x = occurrencehour, fill=as.factor(occurrencehour))) +
295   geom_bar(width=0.8, stat="count", fill = plasma(24)) + theme(legend.position="none") + scale_x_continuous(breaks = 0:23)+
296   ggtitle("Incidents per Hour ") + xlab("Hour") + ylab("Count") +
297   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
298     panel.background = element_blank(), axis.line = element_line(colour = "black"))
299
300
301 #Figure 13: Day/time trends
302 daytime <-MCI_cln %>% group_by(occurrencedayofweek,occurrencehour) %>%
303   dplyr::summarise(N = n()) #sample code found here - https://stackoverflow.com/questions/22767893/count-number-of-rows-by-group-using
304
305 ggplot(daytime, aes(occurrencedayofweek, occurrencehour, fill = N)) +

```

```

306 geom_tile(size = 1, color = "white") +
307 scale_fill_gradient2('N', low = "darkslategray3", high = "darkslategray4", midpoint = 1750) +
308 scale_y_continuous(breaks = 0:23) +
309 ggtitle("Toronto Crimes by Day and Time") + xlab("Day") + ylab("Hour") +
310 theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
311        panel.background = element_blank(), axis.line = element_line(colour = "black"))
312
313 #Figure 14: Updated Table of incidents per Neighbourhood with NSA removed
314 IncHood2<-count(MCI_RemNSA$Neighbourhood)
315 setnames(IncHood2, "x", "Neighbourhood")
316 setnames(IncHood2, "freq", "IncidentCounts")
317 IncHood2_TopInc<-head(IncHood2,10) #some weirdness with ordering top = last 10 and last = top 10
318 IncHood2_LastInc<-head(IncHood2,10)
319
320 BP_IncHood <-MCI_RemNSA %>% group_by(Neighbourhood) %>%
321   dplyr::summarise(N = n())
322
323
324 ggplot(BP_IncHood, aes(x="", N, y=N)) +
325   geom_boxplot(width=0.6, outlier.size=3,outlier.colour="black", fill = "cadetblue") +
326   stat_summary(
327     aes(label=sprintf("%1.1f", ..y..)),
328     geom="text",
329     fun = function(y) boxplot.stats(y)$stats,
330     position=position_nudge(x=0.33),
331     size=3.5) +
332   theme_bw() +
333   stat_boxplot(geom = "errorbar", width = 0.5) +
334   xlab("Neighbourhood") + ylab("Count") +
335   ggtitle("Range of Incident Counts per Neighbourhood")
336
337
338 #Fig 15: incidents per neighbourhood & plot top 10 neighbourhood with most crime
339 IncHood<-count(MCI_cln$Neighbourhood)
340 setnames(IncHood, "x", "Neighbourhood")
341 setnames(IncHood, "freq", "IncidentCounts")
342
343 MCI_RemNSA<-MCI_cln[!grepl("NSA", MCI_cln$Neighbourhood),] #Unlabelled neighbourhoods listed as NSA, removed.
344 MCIhood <-MCI_RemNSA %>% group_by(Neighbourhood, MCI) %>%
345   dplyr::summarise(N = n())
346 MCIhood <- MCIhood[order(MCIhood$N),]
347 MCIhood_top10<-tail(MCIhood, 10)
348
349 ggplot(aes(x = reorder(Neighbourhood, N), y = N), data = MCIhood_top10) +
350   geom_bar(stat = 'identity', width = 0.6, fill = plasma(10)) +
351   geom_text(aes(label = N), stat = 'identity', data = MCIhood_top10, hjust = -0.1, size = 3) +
352   coord_flip() +
353   xlab('Neighbourhoods') +
354   ylab('Incident Counts') +
355   ggtitle('Top 10 Toronto Neighbourhoods with the Most Crime') +
356   theme_bw() +
357   theme(plot.title = element_text(size = 14),
358         axis.title = element_text(size = 12, face = "bold"))
359
360
361
362 #Figure 16: Neighbourhoods with the least crime
363 IncHood3 <-MCI_RemNSA %>% group_by(Neighbourhood) %>% #code from Sundar (2020), Li (2017)
364   dplyr::summarise(N = n())
365 IncHood3 <- IncHood3[order(IncHood3$N), ] #so much drama with these files
366 IncHood3_Last10 <- head(IncHood3, 10)
367 IncHood3_Top10 <- tail(IncHood3, 10)
368
369 ggplot(aes(x = reorder(Neighbourhood, N), y = N), data = IncHood3_Top10) +
370   geom_bar(stat = 'identity', width = 0.6, fill = plasma(10)) +

```

```

371 geom_text(aes(label = N), stat = 'identity', data = IncHood3_Top10, hjust = -0.1, size = 3) +
372 coord_flip() +
373 xlab('Neighbourhoods') +
374 ylab('Incident Counts') +
375 ggtitle('Top 10 Toronto Neighbourhoods with the Most Crime') +
376 theme_bw() +
377 theme(plot.title = element_text(size = 14),
378       axis.title = element_text(size = 12, face = "bold"))
379
380 #Figure 17: Listing of incident types and counts
381 MCICat <-MCI_cln %>% group_by(MCI) %>%
382   dplyr::summarise(N = n())
383 MCICat <- MCICat[order(MCICat$N), ]
384
385 ggplot(aes(x = reorder(MCI, N), y = N), data = MCICat) +
386   geom_bar(stat = 'identity', width = 0.5, fill = "blue") +
387   geom_text(aes(label = N), stat = 'identity', data = MCICat, hjust = -0.1, size = 3.5) +
388   coord_flip() +
389   xlab('Major Crime Indicators') +
390   ylab('Occurrence Count') +
391   ggtitle('Major Crime Indicators Toronto (2014 - 2021)') +
392   theme_bw() +
393   theme(plot.title = element_text(size = 14),
394         axis.title = element_text(size = 12, face = "bold"))
395
396 #Figure 18:Listing of all offence types
397 OffCat <-MCI_cln %>% group_by(offence) %>%
398   dplyr::summarise(N = n())
399 OffCat <- OffCat[order(OffCat$N),]
400
401 ggplot(aes(x = reorder(offence, N), y = N), data = OffCat) +
402   geom_bar(stat = 'identity', width = 0.5, fill = "blue") +
403   geom_text(aes(label = N), stat = 'identity', data = OffCat, hjust = -0.1, size = 3.5) +
404   coord_flip() +
405   xlab('Offence Types') +
406   ylab('Count') +
407   ggtitle('Toronto Criminal Offence Types (2014 - 2021)') +
408   theme_bw() +
409   theme(plot.title = element_text(size = 14),
410         axis.title = element_text(size = 12, face = "bold"))
411
412 #Figure 18: Inset with Top 10 Offenses
413 OffCat_Top10 <- tail(OffCat, 10)
414
415 ggplot(aes(x = reorder(offence, N), y = N), data = OffCat_Top10) +
416   geom_bar(stat = 'identity', width = 0.5, fill = "blue") +
417   geom_text(aes(label = N), stat = 'identity', data = OffCat_Top10, hjust = -0.1, size = 3.5) +
418   coord_flip() +
419   xlab('Offence Types') +
420   ylab('Count') +
421   ggtitle('Toronto Criminal Offence Types - Top 10') +
422   theme_bw() +
423   theme(plot.title = element_text(size = 14),
424         axis.title = element_text(size = 12, face = "bold"))
425
426 #Fig 19: MCI per Neighbourhood
427 MCI_RemNSA<-MCI_cln[!grepl("NSA", MCI_cln$Neighbourhood),]
428 MCIhood <-MCI_RemNSA %>% group_by(Neighbourhood, MCI) %>%
429   dplyr::summarise(N = n())
430 MCIhood <- MCIhood[order(MCIhood$N),]
431 MCIhood_top10<-tail(MCIhood, 10)
432
433 ggplot(aes(x = reorder(Neighbourhood, N), y = N), data = MCIhood_top10) +
434   geom_bar(stat = 'identity', width = 0.5, fill = "blue") +
435   geom_text(aes(label = N), stat = 'identity', data = MCIhood_top10, hjust = -0.1, size = 3.5) +

```

```

436 coord_flip() +
437 xlab('Offence Types') +
438 ylab('Count') +
439 ggtitle('Toronto Criminal Offence Types - Top 10') +
440 theme_bw() +
441 theme(plot.title = element_text(size = 14),
442       axis.title = element_text(size = 12, face = "bold"))
443
444
445
446 #Figure 19: MCI counts per Neighbourhood
447 MCI_N <-MCI_RemNSA %>% group_by(MCI, Neighbourhood) %>%
448   dplyr::summarise(N = n())
449 MCI_N <- MCI_N[order(MCI_N$N),]
450 MCI_N_Top10<-tail(MCI_N, 20)
451
452 ggplot(MCI_N_Top10, aes(x = Neighbourhood, y=N, fill = MCI)) +
453   geom_bar(stat = 'identity', width = 0.8) +
454   xlab('Neighbourhood') +
455   ylab('MCI Count') +
456   ggtitle('Top MCI Categories by Neighbourhood (2014 - 2021)') + theme_bw() +
457   theme(plot.title = element_text(size = 14),
458         axis.title = element_text(size = 12, face = "bold"),
459         axis.text.x = element_text(angle = 70, hjust = 1, vjust = 1))
460
461
462 #Figure 20: MCI counts per Premises
463 MCI_P <-MCI_c1n %>% group_by(MCI,premises_type) %>%
464   dplyr::summarise(N = n())
465
466 ggplot(MCI_P, aes(premises_type, MCI, fill = N)) +
467   geom_tile(size = 1, color = "white") +
468   scale_fill_gradient2('N', low = "cadetblue", mid = "white", high = "darkslategray", midpoint = 25000) +
469   ggtitle("Toronto MCI Categories by Premises Type (2014 - 2021)") + xlab("Premises Type") + ylab("MCI") +
470   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
471         panel.background = element_blank(), axis.line = element_line(colour = "black"))
472
473
474 #Figure 21: MCI counts per Year
475 MCI_Y <-MCI_c1n %>% group_by(occurrenceyear,MCI) %>%
476   dplyr::summarise(N = n())
477
478 ggplot(MCI_Y, aes(occurrenceyear, MCI, fill = N)) +
479   geom_tile(size = 1, color = "white") +
480   scale_fill_gradient2('N', low = "darkslategray4", mid = "yellow", high = "darkslategray", midpoint = 12000) +
481   ggtitle("Toronto MCI Categories by Year (2014 - 2021)") + xlab("Year") + ylab("MCI") +
482   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
483         panel.background = element_blank(), axis.line = element_line(colour = "black")) +
484   scale_x_continuous(breaks = 2014:2021)
485
486
487 #Figure 22: MCI counts per Month
488 MCI_M <-MCI_c1n %>% group_by(occurrencemonth,MCI) %>%
489   dplyr::summarise(N = n())
490
491 ggplot(MCI_M, aes(occurrencemonth, MCI, fill = N)) +
492   geom_tile(size = 1, color = "white") +
493   scale_fill_gradient2('N', low = "darkslategray4", mid = "yellow", high = "darkslategray", midpoint = 7000) +
494   ggtitle("Toronto MCI Categories by Month (2014 - 2021)") + xlab("Month") + ylab("MCI") +
495   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
496         panel.background = element_blank(), axis.line = element_line(colour = "black"))
497
498
499 #Figure 23: MCI counts per Day
500 MCI_D <-MCI_c1n %>% group_by(occurrencedayofweek,MCI) %>%

```



```

501   dplyr::summarise(N = n())
502
503 ggplot(MCI_D, aes(occurrencedayofweek, MCI, fill = N)) +
504   geom_tile(size = 1, color = "white") +
505   scale_fill_gradient2('N', low = "cyan2", mid = "white", high = "cyan4", midpoint = 11000) +
506   ggtitle("Toronto MCI Categories by Day (2014 - 2021)") + xlab("Day") + ylab("MCI") +
507   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
508         panel.background = element_blank(), axis.line = element_line(colour = "black"))
509
510
511
512 #Figure 24: MCI counts per Hour
513 MCI_H <-MCI_cln %>% group_by(occurrencehour,MCI) %>%
514   dplyr::summarise(N = n())
515
516 ggplot(MCI_H, aes(occurrencehour, MCI, fill = N)) +
517   geom_tile(size = 1, color = "white") +
518   scale_fill_gradient2('N', low = "cyan2", mid = "white", high = "cyan4", midpoint = 5000) +
519   ggtitle("Toronto MCI Categories by Hour (2014 - 2021)") + xlab("Hour") + ylab("MCI") +
520   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
521         panel.background = element_blank(), axis.line = element_line(colour = "black"))
522
523
524 #Combine the neighbourhood crime counts from MCI dataset with the Neighbourhoods dataframe
525 #start by adding a new column for avg crime counts per year and populate with AvgCrime (avg for Toronto)
526 IncHood2$AvgCount<-(IncHood2$IncidentCounts)/8
527 IncHood2$Number<-MCI_RemNSA$Hood_ID[match(IncHood2$Neighbourhood,MCI_RemNSA$Neighbourhood)]
528
529 #add the avg yearly crime counts per neighbourhood
530 Nhoods$AvgCrime<-IncHood2$AvgCount[match(Nhoods$Neighbourhood.Number, IncHood2$Number)]
531 #add the avg crime count for Toronto
532 Nhoods[1,25] = AvgCrime
533
534 #generate the average crime per 100k population
535 Nhoods$Avg100k<-(Nhoods$AvgCrime / Nhoods$Population)*100000
536
537 #Add column to designate neighbourhoods as high/low crime
538 #start by dividing each column by the city average, 1 = TO, >1 = high crime, <1 = low crime;
539 Nhoods$Ratio<-(Nhoods$Avg100k / 1190.5923)
540
541 Nhoods$Ratio[Nhoods$Ratio > 1] <- 1 #high crime area
542 Nhoods$Ratio[Nhoods$Ratio < 1] <- 0 #low crime
543
544 #Convert column to factor
545 Nhoods$Ratio<-as.factor(Nhoods$Ratio)
546
547 HoodOff <-MCI_RemNSA %>% group_by(offence, Neighbourhood) %>%
548   dplyr::summarise(N = n())
549
550 HoodOff$Wt<-MCI_cln$weight[match(HoodOff$offence,MCI_cln$offence)]
551 HoodOff$ucr<-MCI_cln$ucr_code[match(HoodOff$offence,MCI_cln$offence)]
552
553 #filter for ucr less than 1700
554 HoodViolent<-filter(HoodOff,ucr < 1700)
555 HoodViolent$Total<- HoodViolent$N * HoodViolent$Wt
556
557

```

main ▾

...

CIND820_Capstone / CIND820_Assignment3_FeatureSelection_InitialResults.R



kmault Add files via upload

[History](#)

1 contributor

233 lines (161 sloc) | 6.78 KB

...

```
1 #Feature Selection, Initial Results & Code
2
3 #Figure 27; final review of dataset, removal of redundant or non-predictive variables
4 str(MCI_db)
5
6 #FEATURE SELECTION
7
8 #Check for correlation among variables
9 set.seed(123)
10 correlationMatrix <- cor(MCI_db[,1:10])
11
12 #Figure 28
13 print(correlationMatrix)
14
15 #Figure 29; Display correlation visually
16 corrplot(correlationMatrix, method="circle")
17
18 #Feature Selection method 1: Boruta
19 #Boruta method of show variable importance; http://r-statistics.co/Variable-Selection-and-Importance-With-R.html;
20 #https://www.machinelearningplus.com/machine-learning/feature-selection/
21 #https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/
22 #https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a
23 #libraries: Boruta, mlbench, randomforest
24
25 set.seed(123)
26 boruta.train <- Boruta(as.factor(class)~., data = MCI_db, doTrace = 2)
27 print(boruta.train)
28
29 # Get significant variable
30 boruta_signif <- getSelectedAttributes(boruta.train, withTentative = TRUE)
31 print(boruta_signif)
32
33 # Variable Importance Scores
34 imps <- attStats(boruta.train)
35 imps2 = imps[imps$decision != 'Rejected', c('meanImp', 'decision')]
36 head(imps2[order(-imps2$meanImp), ]) # descending sort
37
38 # Figure 30: Plot variable importance
39 plot(boruta.train, cex.axis=.7, las=2, xlab="", main="Variable Importance")
40
41
42
43 #Feature Selection Method 2: Stepwise forward and backward elimination; caret and random forest libraries
44 #https://www.machinelearningplus.com/machine-learning/feature-selection/#4stepwiseforwardandbackwardselection
45 #http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/154-stepwise-regression-essentials-in-r/
```

```

46
47 #Stepwise
48 set.seed(123)
49
50 train.control <- trainControl(method = "cv", number = 10)
51
52 step.model <- train(MCI ~., data = MCI_db,
53                     method = "leapSeq",
54                     tuneGrid = data.frame(nvmax = 1:10),
55                     trControl = train.control
56 )
57
58 #summary results
59 step.model$results
60
61 #Display model has the lowest RMSE
62 step.model$bestTune
63
64 #Figure 31: Summary showing the optimal set of variables
65 summary(step.model$finalModel)
66
67
68 #Backwards
69 set.seed(123)
70
71 train.control2 <- trainControl(method = "cv", number = 10)
72
73 step.model2 <- train(MCI ~., data = MCI_db,
74                     method = "leapBackward",
75                     tuneGrid = data.frame(nvmax = 1:10),
76                     trControl = train.control
77 )
78
79 #summary results
80 step.model2$results
81
82 #Display model with the lowest RMSE
83 step.model2$bestTune
84
85 #Figure 32: Summary showing the optimal set of variables
86 summary(step.model2$finalModel)
87
88 #Further reduce features for final classification dataset
89 MCI_Final<-MCI_db[-c(1,3,7,10)]
90 str(MCI_Final)
91
92
93 #Table 2: check relative proportions of classes; data set quite imbalanced
94 MCI_prop<-table(MCI_Final$MCI)
95 MCI_prop
96 round(100*prop.table(MCI_prop))
97
98
99 #Addressing class imbalance
100 #Apply SMOTE method to balance the dataset. Use smote family library
101 set.seed(123)
102 smote<-SMOTE(MCI_Final[, -11], MCI_Final$MCI)
103 smote=smote$data
104 Smote_prop<-table(smote$MCI)
105 Smote_prop
106 round(100*prop.table(Smote_prop))
107
108 #write to csv
109 write.csv(smote,"D:/Ryerson Big Data/CIND820 Big Data Analytics Project/Assignment3/MCI_mod.csv", row.names = FALSE)
110

```

```

111 #Rename dataset then split the data: Train & Test
112 MCI_mod<-smote
113 #library(caTools)
114 set.seed(123)
115 TrainInd<-sample(1:nrow(MCI_mod), 0.8*nrow(MCI_mod))
116 Train<-MCI_mod[TrainInd,]
117 Test<-MCI_mod[-TrainInd,]
118
119 write.csv(Train,"D:/Ryerson Big Data/CIND820 Big Data Analytics Project/Assignment3/TrainingData.csv", row.names = FALSE)
120 write.csv(Test,"D:/Ryerson Big Data/CIND820 Big Data Analytics Project/Assignment3/TestingData.csv", row.names = FALSE)
121
122
123 #Model 1: Decision Tree Model (J48) with 10 fold cross validation; RWeka library
124 #Follow method as outlined here: https://cran.r-project.org/web/packages/RWeka/RWeka.pdf
125 #https://rdr.io/cran/RWeka/man/Weka\_control.html
126
127 #create training model and evaluate
128 set.seed(123)
129 DT <- J48(as.factor(class)~., data = Train, control=Weka_control(M=5))
130
131 #10 fold cross validation
132 EV<-evaluate_Weka_classifier(DT,numFolds = 10)
133 EV
134 EV$details
135
136 #predict using J48
137 predDT<-predict(DT, Test, type="class")
138
139 #DT confusion matrix
140 confDT<-table(Test$class, predDT, dnn=c("Actual", "Predicted"))
141 confDT
142
143 #evaluate model
144 confusionMatrix(as.factor(Test$class), as.factor(predDT))
145
146
147
148
149 #Model 2: multivariate logistic regression - same parameters as example
150 #https://stackoverflow.com/questions/39550118/cross-validation-function-for-logistic-regression-in-r
151 #https://www.youtube.com/watch?v=fDjKa7yWk1U; nnet library
152
153 #set up traint/test & 10 fold cross validation
154 LR_train<-Train
155 LR_test<-Test
156 set.seed(123)
157 tc <- trainControl(method = "cv", number = 10)
158
159 # Training the multinomial model
160 MN_model <- multinom(class ~ .,
161                       data = LR_train,
162                       method = 'glm',
163                       trControl = tc,
164                       family = binomial()
165 )
166
167 # Checking the model
168 summary(MN_model)
169
170 #convert coefficients to odds
171 exp(coef(MN_model))
172
173 #top observations
174 head(round(fitted(MN_model), 2))
175

```

```

176
177 #Prediction
178 # Predicting the values for train dataset
179 LR_train$Pred <- predict(MN_model, newdata = LR_train, "class")
180
181
182 # Building classification table
183 tab <- table(LR_train$class, LR_train$Pred)
184
185 #confusion matrix for training model
186 CM_train<-confusionMatrix(tab)
187 CM_train
188
189 #misclassification error
190 1-sum(diag(tab))/sum(tab)
191
192
193 # Calculating accuracy - sum of diagonal elements divided by total obs
194 round((sum(diag(tab))/sum(tab))*100,2)
195
196 # Predicting the class for test dataset
197 LR_test$Pred <- predict(MN_model, newdata = LR_test, "class")
198 # Building classification table
199 tab2 <- table(LR_test$class, LR_test$Pred)
200 tab2
201
202 CM_mod<-confusionMatrix(tab2)
203 CM_mod
204
205 # Calculating accuracy of predictive model - sum of diagonal elements divided by total obs
206 round((sum(diag(tab2))/sum(tab2))*100,2)
207
208
209
210 #Model 3: Naive Bayes with 10 fold cross validation:Balanced Data
211 #https://www.geeksforgeeks.org/naive-bayes-classifier-in-r-programming/
212 #https://rpubs.com/maulikpatel/224581
213 #https://www.analyticsvidhya.com/blog/2021/03/introduction-to-k-fold-cross-validation-in-r/; naive bayes package
214 NB_train<-Train
215 NB_test<-Test
216
217 set.seed(100)
218 trctrl <- trainControl(method = "cv", number = 10, savePredictions=TRUE)
219 nb_fit <- train(as.factor(class) ~., data = NB_train, method = "naive_bayes", trControl=trctrl, tuneLength = 0)
220 nb_fit
221
222
223 #predict based on the NB Model
224 y_predNB <- predict(nb_fit, newdata = NB_test)
225
226 #NB Confusion Matrix & evaluation
227 cmNB <- table(NB_test$class, y_predNB, dnn = c("Actual", "Predicted"))
228 cmNB
229
230
231 confusionMatrix(as.factor(NB_test$class), as.factor(y_predNB))
232
233

```