Machine Learning for Epi: Assignment 4

2023-02-14

## Part I: Implementing a Simple Prediction Pipeline

The New York City Department of Health administered a questionnaire on general health and physical activity among residents.

### Step 1: Load data and prepare for analysis

The code chunk below loads the Physical Activity and Transit (PAT) survey data, renames variables to meaningful names, and when appropriate, converts numeric variables to factors.

PAT\_df = readr::read\_csv("./class4\_p1.csv") %>%   
 dplyr::rename(id = `...1`,  
 htn = chronic1,  
 diabetes = chronic3,  
 asthma = chronic4,  
 tobacco = tobacco1,  
 alcohol = alcohol1,  
 pa\_min = gpaq8totmin,  
 walk\_days = gpaq11days,  
 physical\_act = habits5,  
 diet = habits7,  
 gender = dem3,  
 hisp\_lat = dem4,  
 us\_born = dem8) %>%   
 mutate\_at(vars(htn:asthma, tobacco:alcohol, physical\_act:povertygroup), ~ as.factor(.))   
  
summary(PAT\_df) %>% knitr::kable(digits = 2)

|  | id | htn | diabetes | asthma | bmi | tobacco | alcohol | pa\_min | walk\_days | physical\_act | diet | agegroup | gender | hisp\_lat | us\_born | povertygroup | healthydays |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Min. : 1.0 | 1 :1096 | 1 : 426 | 1 : 236 | Min. : 9.09 | 1 : 402 | 1 : 109 | Min. : 0.00 | Min. :0.000 | 1 :1163 | 1 : 324 | 1 : 277 | 1:1546 | 1 : 843 | 1 :2338 | 1 :690 | Min. : 0.00 |
|  | 1st Qu.: 953.5 | 2 :2687 | 2 :3368 | 2 :3567 | 1st Qu.:23.05 | 2 : 201 | 2 :1140 | 1st Qu.: 0.00 | 1st Qu.:2.000 | 2 :1772 | 2 : 695 | 2 :1177 | 2:2265 | 2 :2938 | 2 :1467 | 2 :607 | 1st Qu.:27.00 |
|  | Median :1906.0 | NA’s: 28 | NA’s: 17 | NA’s: 8 | Median :26.47 | 3 :3205 | 3 :2545 | Median : 0.00 | Median :5.000 | 3 : 661 | 3 : 886 | 3 :1451 | NA | NA’s: 30 | NA’s: 6 | 3 :611 | Median :30.00 |
|  | Mean :1906.0 | NA | NA | NA | Mean :27.32 | NA’s: 3 | NA’s: 17 | Mean : 43.77 | Mean :4.273 | 4 : 198 | 4 : 458 | 4 : 898 | NA | NA | NA | 4 :581 | Mean :26.34 |
|  | 3rd Qu.:2858.5 | NA | NA | NA | 3rd Qu.:30.23 | NA | NA | 3rd Qu.: 60.00 | 3rd Qu.:7.000 | NA’s: 17 | 5 : 113 | NA’s: 8 | NA | NA | NA | 5 :758 | 3rd Qu.:30.00 |
|  | Max. :3811.0 | NA | NA | NA | Max. :99.83 | NA | NA | Max. :960.00 | Max. :7.000 | NA | NA’s:1335 | NA | NA | NA | NA | 6 :320 | Max. :30.00 |
|  | NA | NA | NA | NA | NA’s :73 | NA | NA | NA’s :5 | NA’s :5 | NA | NA | NA | NA | NA | NA | NA’s:244 | NA’s :88 |

Based on the summary of the output, we can see that the features BMI, pa\_min (minutes of physical activity on chores), and walk\_days are continuous. Meanwhile, the remaining features are factor variables (htn, diabetes, asthma, tobacco, alcohol, physical\_act, diet, gender, hisp\_lat, us\_born, povertygroup).

### Step 2: Pre-process the data

Next, we will preprocess the data. The code chunk below removes the id feature, omits NA observations. We will also want to center and scale the data, however, the caret package will do this for us within the model training step.

set.seed(123)  
  
#Drop category and ID variable and remove missings  
PAT\_df = PAT\_df %>%   
 select(-id) %>%   
 na.omit()

### Step 3: Partition the data

The code chunk below partitions the data into training and testing sets, using a 70/30 split.

#Creating balanced partitions in the data  
train\_index = createDataPartition(PAT\_df$healthydays, p = 0.7, list = FALSE)  
  
PAT\_train = PAT\_df[train\_index,]  
PAT\_test = PAT\_df[-train\_index,]  
  
#Check distribution of the outcome between train and test data  
summary(PAT\_test$healthydays)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 27.0 30.0 26.5 30.0 30.0

summary(PAT\_train$healthydays)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 27.00 30.00 26.39 30.00 30.00

We can see that the training and testing sets have similar median, mean, and IQR ranges, indicating that the data were successfully partitioned.

### Step 4: Construct linear regression models to predict healthy days

We will fit two linear models to predict the number of days in a month an individual reported having good physical health (feature name: healthydays).

* Model 1 (lm\_all): Include all features
* Model 2 (lm\_habits): Include only health habits and physical activity variables (tobacco + alcohol + pa\_min + walk\_days + physical\_act + diet)

These models will be trained on the training dataset using 10-fold cross validation. The data will be centered and scaled within the preProc option within train().

#Perform 10-fold cross-validation  
control.settings = trainControl(method = "cv", number = 10)  
  
#Fit models using caret  
lm\_all = train(healthydays ~., data = PAT\_train, method = "lm", preProc = c("center", "scale"), trControl = control.settings)  
  
lm\_habits = train(healthydays ~ tobacco + alcohol + pa\_min + walk\_days + physical\_act + diet, data = PAT\_train, method = "lm", preProc = c("center", "scale"), trControl = control.settings)  
  
# Output results  
rbind(lm\_all$results, lm\_habits$results) %>%   
 mutate(Model = c("All", "Habits")) %>%   
 relocate(Model, .before = intercept) %>%   
 knitr::kable(digits = 3)

| Model | intercept | RMSE | Rsquared | MAE | RMSESD | RsquaredSD | MAESD |
| --- | --- | --- | --- | --- | --- | --- | --- |
| All | TRUE | 7.248 | 0.128 | 4.602 | 0.454 | 0.036 | 0.372 |
| Habits | TRUE | 7.424 | 0.086 | 4.782 | 0.903 | 0.065 | 0.512 |

Results of the cross-validated training shows the model with all variables, lm\_all, has a slightly lower RMSE compared to the model with only health habits. However, we want to confirm this by running each model on the testing data.

### Step 5: Model Evaluation

Next, to determine the preferred prediction model, we will apply both models to the test data, and compare their performance based on the RMSE, which is an evaluation metric used for linear regression models.

test\_outcome1 = predict(lm\_all, PAT\_test)  
All\_RMSE = RMSE(test\_outcome1, PAT\_test$healthydays)  
  
test\_outcome2 = predict(lm\_habits, PAT\_test)  
Habits\_RMSE = RMSE(test\_outcome2, PAT\_test$healthydays)  
  
cbind(All\_RMSE, Habits\_RMSE) %>%   
 as\_tibble() %>%   
 knitr::kable(digits = 3)

| All\_RMSE | Habits\_RMSE |
| --- | --- |
| 7.172 | 7.387 |

The table shows that the lm\_all model performs better on the test data, with an RMSE of 7.172, compared to the lm\_habits model’s RMSE of 7.413. Therefore, if I were only interested in prediction performance, I would select the linear model with all features over the linear model with health habit features to predict the number of healthy days in a month.

One setting where the lm\_all model would be useful would be in a program evaluation setting. For instance, in a health program that promotes increased physical activity, when enrolling new subjects, we could predict the number of healthy days they are predicted to have in a month at baseline, and then compare the number of observed healthy days after program participation. This provides evidence for evaluating the success of the physical activity program in improving the number of healthy days a person may have.

## Part II: Conducting an Unsupervised Analysis

In Part II, we will use a data-driven hierarchical clustering approach to identify the optimal number of clusters in the R dataset, USArrests. The data includes the crime statistics for each of the 50 US states in 1973. Incidence of arrest, per 100,000 residents for assault, murder and rape are included along with the proportion of the population that lives in urban communities.

### Transform data

First, we will load the data and check the means and SD to determine whether scaling and centering our data is necessary:

USArrests = USArrests %>% na.omit()  
summary(USArrests) %>% knitr::kable()

|  | Murder | Assault | UrbanPop | Rape |
| --- | --- | --- | --- | --- |
|  | Min. : 0.800 | Min. : 45.0 | Min. :32.00 | Min. : 7.30 |
|  | 1st Qu.: 4.075 | 1st Qu.:109.0 | 1st Qu.:54.50 | 1st Qu.:15.07 |
|  | Median : 7.250 | Median :159.0 | Median :66.00 | Median :20.10 |
|  | Mean : 7.788 | Mean :170.8 | Mean :65.54 | Mean :21.23 |
|  | 3rd Qu.:11.250 | 3rd Qu.:249.0 | 3rd Qu.:77.75 | 3rd Qu.:26.18 |
|  | Max. :17.400 | Max. :337.0 | Max. :91.00 | Max. :46.00 |

#Check means and SDs to determine if scaling is necessary  
tibble(rate = c("Murder", "Assault", "UrbanPop", "Rape"),  
 means = colMeans(USArrests, na.rm = TRUE),   
 sd = apply(USArrests, 2, sd, na.rm = TRUE)) %>%  
 knitr::kable(digits = 2)

| rate | means | sd |
| --- | --- | --- |
| Murder | 7.79 | 4.36 |
| Assault | 170.76 | 83.34 |
| UrbanPop | 65.54 | 14.47 |
| Rape | 21.23 | 9.37 |

We can see that the means differ significantly across the features of murder, assault, and rape rates, as well as the urban population. Therefore, we should scale and center our features. The code chunk below centers and scales the features in the USArrests and outputs the transformed data into a transformed\_USArrests dataframe.

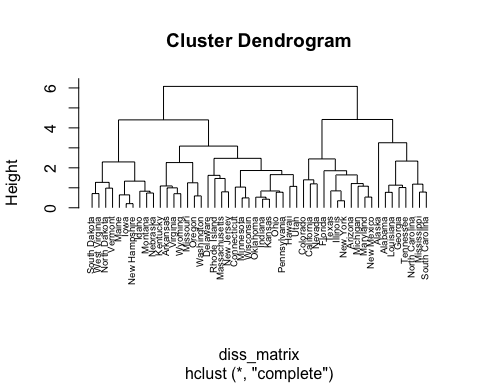
#Centering and Scaling  
USArrests\_numeric = USArrests %>% dplyr::select(where(is.numeric))  
USArrests\_preprocess = preProcess(USArrests\_numeric, method = c("center", "scale"))  
  
#Output pre-processed values  
transformed\_USArrests = predict(USArrests\_preprocess, USArrests\_numeric)  
transformed\_USArrests %>% head()

## Murder Assault UrbanPop Rape  
## Alabama 1.24256408 0.7828393 -0.5209066 -0.003416473  
## Alaska 0.50786248 1.1068225 -1.2117642 2.484202941  
## Arizona 0.07163341 1.4788032 0.9989801 1.042878388  
## Arkansas 0.23234938 0.2308680 -1.0735927 -0.184916602  
## California 0.27826823 1.2628144 1.7589234 2.067820292  
## Colorado 0.02571456 0.3988593 0.8608085 1.864967207

### Hierarchical clustering analysis

Now with the transformed USArrests data, we will conduct the hierarchical clustering analysis. The code chunk below creates a dissimilarity matrix based on **Euclidean** distances. We will then apply a **complete** linkage agglomeration method to measure the dissimilarity between clusters.

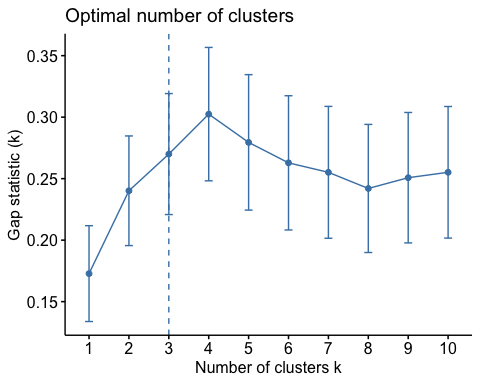
set.seed(123)  
  
# Create Dissimilarity matrix  
diss\_matrix = dist(transformed\_USArrests, method = "euclidean")  
  
# Hierarchical clustering using Complete Linkage  
clusters\_h = hclust(diss\_matrix, method = "complete")  
  
# Plot the obtained dendrogram  
plot(clusters\_h, cex = 0.6, hang = -1)



The above dendrogram shows the result of our hierarchical clustering analysis. We can see that the most granular clusters each contain 1-2 observations, and as we move up the tree, similar observations are combined into branches.

Finally, to determine the optimal number of clusters, we will use the gap statistic method.

#Calculate gap statistic  
gap\_stat = clusGap(transformed\_USArrests, FUN = hcut, nstart = 25, K.max = 10, B = 50)  
fviz\_gap\_stat(gap\_stat)

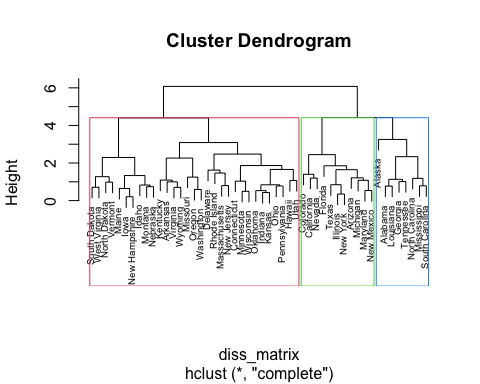


Based on the gap-statistic plot, we can see the optimal number of clusters is 3. Let’s use number of clusters from gap statistic to visualize each cluster in our dendrogram.

clusters\_h\_3 = cutree(clusters\_h, k = 3)  
table(clusters\_h\_3)

## clusters\_h\_3  
## 1 2 3   
## 8 11 31

plot(clusters\_h, cex = 0.6)  
rect.hclust(clusters\_h, k = 3, border = 2:5)



The resulting table and plot shows marked differences in the number observations per cluster, with 8 in the 1st cluster, and 31 in the third cluster. To inspect further, we can examine the feature characteristics from sample of 5 observations from each cluster in our data.

USArrests %>%  
 mutate(cluster = clusters\_h\_3) %>%  
 group\_by(cluster) %>%   
 slice\_sample(n = 5) %>%   
 knitr::kable()

| Murder | Assault | UrbanPop | Rape | cluster |
| --- | --- | --- | --- | --- |
| 14.4 | 279 | 48 | 22.5 | 1 |
| 13.2 | 188 | 59 | 26.9 | 1 |
| 17.4 | 211 | 60 | 25.8 | 1 |
| 13.0 | 337 | 45 | 16.1 | 1 |
| 10.0 | 263 | 48 | 44.5 | 1 |
| 9.0 | 276 | 91 | 40.6 | 2 |
| 11.3 | 300 | 67 | 27.8 | 2 |
| 10.4 | 249 | 83 | 24.0 | 2 |
| 15.4 | 335 | 80 | 31.9 | 2 |
| 11.1 | 254 | 86 | 26.1 | 2 |
| 6.3 | 106 | 72 | 14.9 | 3 |
| 3.2 | 120 | 80 | 22.9 | 3 |
| 2.2 | 48 | 32 | 11.2 | 3 |
| 8.5 | 156 | 63 | 20.7 | 3 |
| 2.6 | 120 | 54 | 14.2 | 3 |

We can see that cluster 1 is characterized by cities with a relatively low urban population (45-60%) and higher rates of murder (10-17.4 per 100,000). Cluster 2 is characterized by cities with a high urban population (67-91%), higher rates of rape and assault. Finally, cluster 3 is characterized by a medium urban population (32-80%), and relatively lower rates of murder (2.2-8.5 per 100,000), assault, and rape.

### Research Application of Clustering Analysis

Assuming these data are from 2020, we could potentially use these data to identify cities that vary in key crime characteristics, and design cluster-specific crime reduction programs that best suit the crime-related features of each cluster. Our research question would be whether designing crime prevention programs according to cities’ urban population, and rates of assault, rape, and murder are effective in reducing such crime rates. In this instance, clusters would serve as an exposure in identifying the appropriate crime prevention program to be applied to each city.

Before conducting this research, there are clearly some ethical considerations we should consider. For instance, we should consider the inherent biases present in the data, such as the factors that may lead to over- or under-reporting of crime statistics across cities. The implications of relying on invalid, biased data in a cluster analysis may justify excessive punitive crime prevention efforts in certain cities, reinforce existing crime-related stigmas or biases of different cities, and risk deeming high-crime clusters as ‘problematic’ for law enforcement and policymakers.