Assignment3 Clustering-CSE780

Paria, Fakhrzad
Stuent ID: 400353290

7-Nov-2021

Part a.Data Summary

In this project we will use a dataset related to an insurance company [1]. There are 10000 customer in this data. Also there is 18features that one of them is a label, a factor variable that shows whether customer had insurance claim last year or not. The correlation between variables shows in figure1.

There are 1939 NA values in dataset that all have removed. And since we need to perform distance and scale functions just below 15 numeric features select for this clustering. Also the last column that is label has been omitted.

$X1$ age - Min:20 Max:65 Mean:42

$X2$ gender - female: 4084(50%) male: 4065(50%)

$X3$ race - majority(0): 7323(90%) minority(1): 826(10%)

$X4$ driving experience - Min:5 Max:35 Mean:16

$X5$ education - none(1): 1528(19%) high school(2):3404(42%) university(3):3217(39%)

$X6$ income - upper(1):3588(44%) middle(2): 1727(21%) working(3):1375(17%) poverty(4):1459(18%)

$X7$ Credit score - Min:0.0533 Max:0.960 Mean:0.516

$X8$ vehicle ownership True(1):5698(70%) False(0): 2451(30%)

$X9$ married- True(1): 4083(50%) False(0): 4066(50%)

$X10$ children True(1): 5617(69%) False(0): 2532(31%)

$X11$ postal code- this is 5 different code and it has meaning based on company definition

$X12$ annual mileage- Min:2000 Max: 22000 Mean: 11693

$X13$ speeding violation- Min:0 Max:22 Mean: 1.48

$X14$ DUI - Min:0 Max:6 Mean: .25

$X15$ past accident - Min:0 Max:15 Mean: 1

We can see that customers with average age 20 and income level poverty and education level high school have more accident claim in last year.

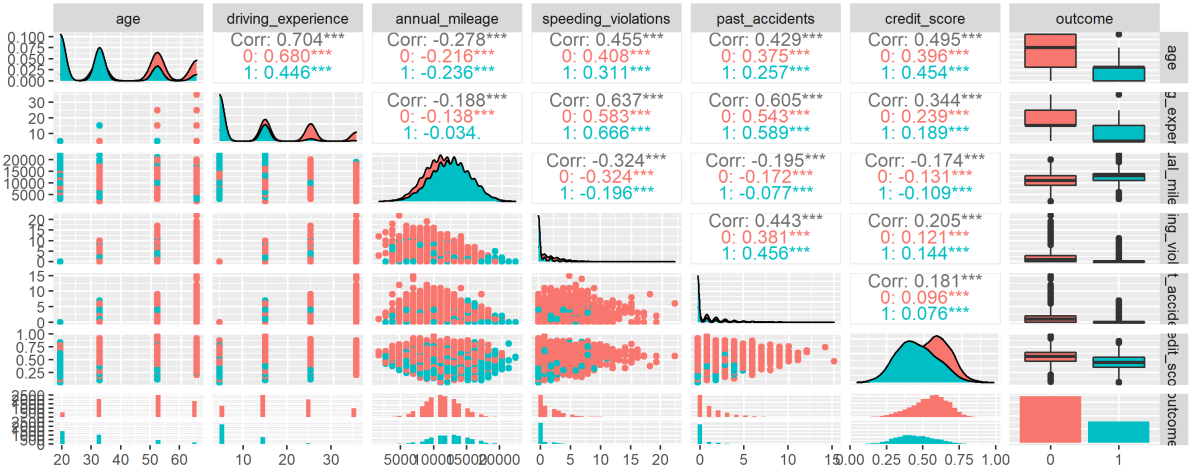

Figure 1: Features relationship

Part b.Clustering

In This part we use three clustering algorithm, Kmeans, Hierarchical and Gaussian Mixture model and the goal is finding the best clusters of customers that show same pattern based on this dataset features. For preparing the dataset, we normalize all columns also calculate the distance matrix by euclidean method.

b.1 K-means Clustering

First of all we need to know optimal number of clusters and there are two method for calculating this K. it shows in figure2 that is 2.

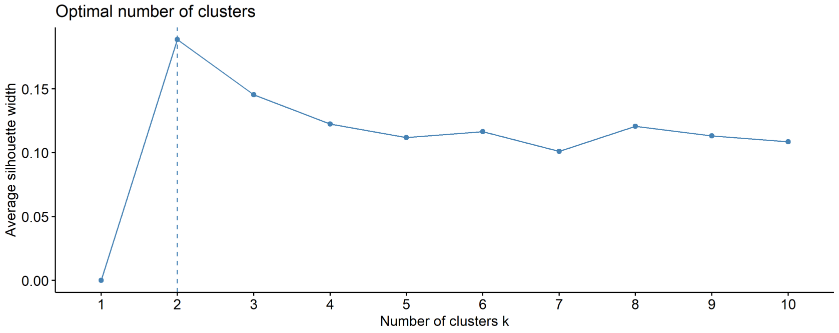

Figure 2: silhouette

After fitting K-mean we see that the $Withinclustersumofsquaresbycluster$ are 57355.83 and 39454.42 respectively and (between_SS / total_SS = 20.8%).Figure3 shows these clusters based on two features credit_score and annual_milleage.

b.2 Hierarchical Clustering

In this part we fit hierarchical model with four linkage *single*, *complete*,*average* and *centroid*. Since the number of customers are huge the dendogram plots are not clear and here we cut it by 2.

b.3 Gaussian Mixture Model clustering

As we can see in figure 4 and 5 the hierarchical and Gaussion models don't perform well in this dataset when the number of observations are huge and we need 2 clusters.

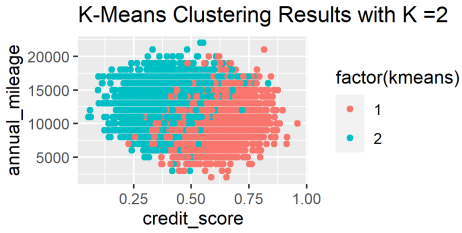

Figure 3: K-means

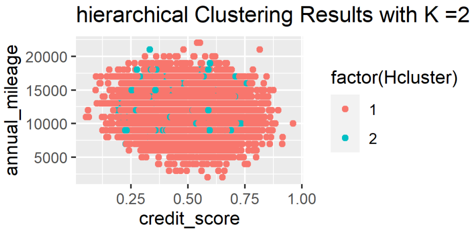

Figure 4: Hierarchical

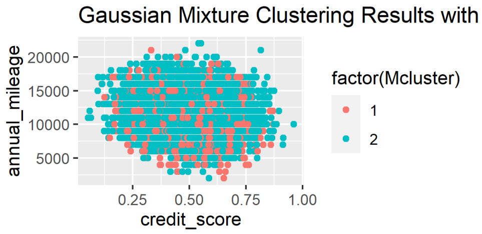

Figure 5: Gaussian Mixture Model

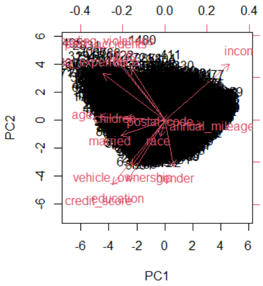

Figure 6: PCA Variance

c.Principal Component Analysis

In this Part we used PCA for dimension reduction to see the effect on the result of clustering models.

c.1 Perform PCA

When we perform PCA, see that 15 PCA will be generate that the details are in table1.

Table 1: PCA

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15
Standard deviation	2.1037	1.2785	1.1614	1.0429	1.0209	0.9884	0.8856	0.8644	0.8345	0.7565	0.7138	0.6776	0.5931	0.4458	0.4064
Proportion of Variance	0.2950	0.1090	0.0899	0.0725	0.0695	0.0651	0.0523	0.0498	0.0464	0.0382	0.0340	0.0306	0.0234	0.0132	0.0110
Cumulative Proportion	0.2950	0.4040	0.4939	0.5664	0.6359	0.7010	0.7533	0.8031	0.8496	0.8877	0.9217	0.9523	0.9757	0.9890	1.0000

Here we can see that PCA1 and PCA2 have 40% of variation. In rotation matrix we can see that score of 15 features for PCA1 is not equal and some of features such as ‘gender‘, ‘race‘ and ‘postal_code‘ have not explained by PC1, on the other hand PC4 for these features has the most score, see figure6.

c.2 Clustering with PCA

We fit K-means and hierarchical clustering models by PCA features and see the result in figure 7 and 8. Since there are less correlation between features and PCA1 and PCA2 just have 40% of variance, using the PCA reduction is not a good solution in this dataset.

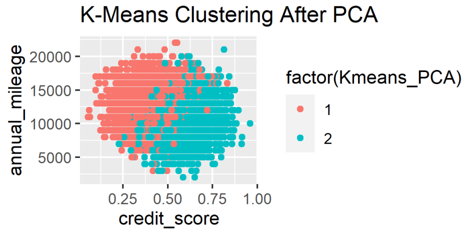

Figure 7: K-means after PCA

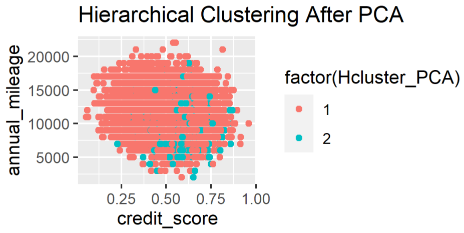

Figure 8: Hierarchical After PCA

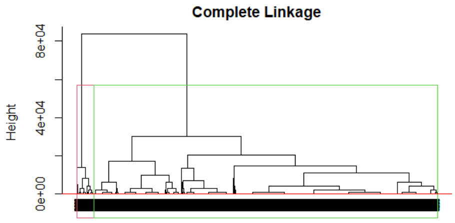

Figure 9:

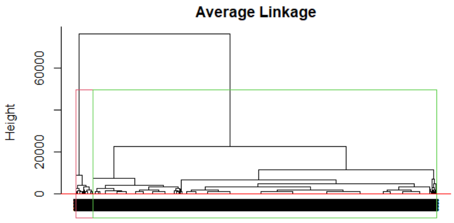

Figure 10:

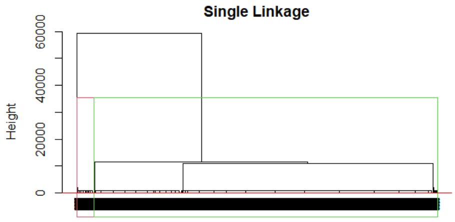

Figure 11:

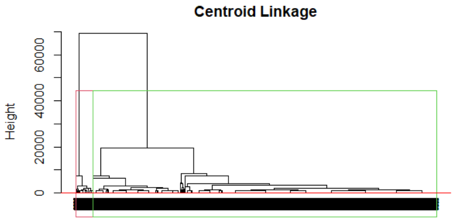

Figure 12:

References

[1] insurance. https://www.kaggle.com/racholsan/customer-data, 2021. URL.

Assignment3- Clustering // Supplementry

Paria Fakhrzad

11/7/2021

Contents

Part a- Data

1.1- loading libraries

```
library(dplyr) # using for dataset exploration
library(Hmisc) # for plots
library(magrittr)
library(readr) # for CSV reading
library(ggplot2) # for plots
library(ISLR2)
library(ggpubr)
library(GGally)
library(PreProcess)
library(caTools)
library(caret)
library(mclust) # for Gaussian Mixtures
```

1.2- loading dataset

In this assignment we are using a dataset related to insurance company[1]. There are 10000 samples in this dataset that are customers of this company and their 18 features in year 2020~[?]. There are 10000 customer as sample in this data frame.

```
cust_df_original <- readr:: read_csv("customer-data.csv")
cust_df <- cust_df_original
```

1.3- Dataset Exploration and transformation

1.3.1 - Summary of dataset

```
names(cust_df) #name of columns
dim(cust_df)   #dimention of dataset 10000 * 18
str(cust_df)
summary(cust_df)
# feature types
```

- `cust_df` is a data frame with 10000 observations on 18 variables.
- There are eighteen variables in the data set that are logical, number and character
- The data shows that 23percent of customers claimed the accident insurance last year.

1.3.2 - Outliers and missing values

```
#finding null observation
table(sapply(cust_df,function(x)all(is.na(x))))#Columns are totally empty
table(lapply(cust_df,function(x){length(which(is.na(x)))})) #Columns with NA
```

Here we can see that 957 samples have null value in annual_mileage and 982 samples have null value in credit_score.Therefore in below we omit these `NA` observations from our dataset.

```
cust_df<- dplyr::filter(cust_df,!is.na(annual_mileage))
cust_df<- dplyr::filter(cust_df,!is.na(credit_score))
```

After observing the summary table in these dataset we find there is no outlier in this dataset.

[1] https://www.kaggle.com/racholsan/customer-data

1.3.3 - Data Tyding

For using clustering method we need to kepp all number features or change the type to number so we will do below changes:

- The age and experience were interval that both have converted to Number(by mean of interval).
- Other char columns, now are number with this logic:
- Gender female=1 , male=0
- married True=1 , false=0
- children True=1 , false=0
- race majority=1, minority=0
- education high none=1, high school=2, university=3
- income Upper class=1 , middle class=2, working class=3, poverty=4
- vehicle ownership True=1 , false=0
- outcome True=1, False=0

```
# changing the age to numbers
cust_df$age <- ifelse(cust_df$age=="65+",65,cust_df$age)
cust_df$age <- ifelse(cust_df$age=="16-25",20,cust_df$age)
cust_df$age <- ifelse(cust_df$age=="26-39",33,cust_df$age)
cust_df$age <- ifelse(cust_df$age=="40-64",52,cust_df$age)
cust_df$age <- as.numeric(cust_df$age)
```

```
# changing the driving experience to number
cust_df$driving_experience <- ifelse(cust_df$driving_experience=="0-9y",5,
                                            cust_df$driving_experience)
cust_df$driving_experience <- ifelse(cust_df$driving_experience=="10-19y",15,
                                            cust_df$driving_experience)
cust_df$driving_experience <- ifelse(cust_df$driving_experience=="20-29y",25,
                                            cust_df$driving_experience)
cust_df$driving_experience <- ifelse(cust_df$driving_experience=="30y+",35,
                                            cust_df$driving_experience)
cust_df$driving_experience <- as.numeric(cust_df$driving_experience)
```

```
#change gender, children and married to number
cust_df$gender <- as.numeric(ifelse(cust_df$gender=="female",1,0))
cust_df$married <- as.numeric(ifelse(cust_df$married=="TRUE",1,0))
cust_df$children <- as.numeric(ifelse(cust_df$children=="TRUE",1,0))
cust_df$vehicle_ownership <- as.numeric(
  ifelse(cust_df$vehicle_ownership=="TRUE",1,0))
```

```
#income
cust_df$income <- ifelse(cust_df$income=="upper class",
                                           1,cust_df$income)
cust_df$income <- ifelse(cust_df$income=="middle class",
                                           2,cust_df$income)
cust_df$income <- ifelse(cust_df$income=="working class",
                                           3,cust_df$income)

cust_df$income <- ifelse(cust_df$income=="poverty",
                                           4,cust_df$income)
cust_df$income <-as.numeric(cust_df$income)

#education
cust_df$education <- ifelse(cust_df$education=="none",
                                           0,cust_df$education)
cust_df$education <- ifelse(cust_df$education=="high school",
                                           1,cust_df$education)
cust_df$education <- ifelse(cust_df$education=="university",
                                           2,cust_df$education)
cust_df$education <-as.numeric(cust_df$education)

#race
cust_df$race <- ifelse(cust_df$race=="minority",0,1)

cust_df$vehicle_year <- as.factor(cust_df$vehicle_year)
cust_df$vehicle_type<- as.factor(cust_df$vehicle_type)
```

```
#change the claim outcome to factor
cust_df <- mutate(cust_df, outcome =
                        factor(outcome, levels = c("FALSE", "TRUE"),
                        labels = c(0, 1)))
summary(cust_df)
```

1.4- Data analysis

In this section we are going to use visualization tools and statistical methods to find the relationship between features and find significant notice about this dataset.

```
xtabs(~age+outcome,cust_df)
```

```
xtabs(~income+outcome,cust_df)
```

```
xtabs(~education+outcome,cust_df)
```

In these tables it can bee seen that customers with average age 20 (between 15-25) has the most rate of accident claims, Also customers if highschool education have more tend to claim for car insurance compared to other education levels.

```r
#using ggpairs() for founding the correlation of numeric features
customer_corr_matrix<-dplyr::select(cust_df,c("age","driving_experience",
                                  "annual_mileage","speeding_violations",
                                  "past_accidents","credit_score","outcome"))

figure1 <-ggpairs(customer_corr_matrix, columns = 1:7,
                  ggplot2::aes(colour=outcome))
figure1
ggsave("figure1.png",figure1, width=10, height = 4)
```

In this ggpair plot, the significant point is that Age and driving experience are collinear. It appears that older customers who own the vehicle tended to less claim (outcome in figure legend) than those who did not. Also in figure5 it seems there are relationship between credit score and number of claims in last year.

```r
# speeding_violations and past accidents
figure2 <-ggscatter(cust_df,x ='speeding_violations',y ='past_accidents',
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson")+
          scale_color_manual(values = c("green", "red"))

# income and age
figure3 <- ggplot(cust_df) +
  geom_boxplot(aes(x = income, y = age, fill = outcome)) +
  theme(legend.position = "none") +
  theme_bw() +
  scale_fill_manual(values = c("blue2", "orange"))

# vehicle ownership and age
figure4 <- ggplot(cust_df) +
  geom_boxplot(aes(x = vehicle_ownership, y = age, fill = outcome)) +
  theme(legend.position = "none") +
  theme_bw() +
  scale_fill_manual(values = c("blue2", "orange"))

# credit score and outcome
figure5 <- ggplot(cust_df) +
  geom_boxplot(aes(x = outcome, y = credit_score)) +
  theme(legend.position = "none") +
  theme_bw()


figure2
figure3
figure4
figure5
```

```
ggsave("figure2.png",figure2,width = 3, height = 2)
ggsave("figure3.png",figure3,width = 3, height = 2)
ggsave("figure4.png",figure4,width = 3, height = 2)
ggsave("figure5.png",figure5,width = 3, height = 2)
```

Partb- Clustering

2.1- Data Preprocessing

In this part we are going to apply differenct clustering algorithm in customer dataset to see if the customers can be assigned to specific clusters. here first of all we keep just variables with numeric types and also remove the outcome label.

```
#remove categorical features
cust_df_num <- dplyr::select(cust_df,-vehicle_year)
cust_df_num <- dplyr::select(cust_df_num,-vehicle_type)
cust_df_num <- dplyr::select(cust_df_num,-outcome)
cust_df_num <- dplyr::select(cust_df_num,-id)

#check the data
str(cust_df_num)
```

now we can see that number of features are 15 here. We need to normalize the data by scale() function

```
#scale data
cust_df_num_scale <- scale(cust_df_num)
```

Also need to calculate distance matrix. here we use Euclidean method.

```
#distance
cust_df_dis<- dist(cust_df_num,method = 'euclidean')
cust_df_dis_scale<- dist(cust_df_num_scale,method = 'euclidean')
```

2.2- Kmeans clustering

In this part we use K-means clustering algorithm and for having the optimal number of K (clusters) here firstly we used elbow method to calculate Within sum of squares(WSS) and minimum one would be best response.There are two ways for finding the K:

```
#solution 1 silhouette
library(factoextra)
figure17<- factoextra:: fviz_nbclust(cust_df_num_scale, kmeans, method="silhouette")+
#  labs(subtitle = "Elbow method")
```

```
#solution 2 WSS elbow method
set.seed(345)
wss <- sapply(1:10, function(k){kmeans(cust_df_num_scale, k, nstart=20)$tot.withinss})

figure16<- plot(x= 1:10 ,y=wss, type="l", frame=FALSE, xlab="clusters-K", ylab="WSS per cluste
figure17
figure16
ggsave("figure17.png", figure17, width=10)
ggsave("figure16.png", figure18, width=10)
```

- fitting K-means with two cluster

```
set.seed(720)
KM_model <-kmeans(cust_df_num_scale, centers=2, nstart=20)
KM_model
KM_model$tot.withinss
```

with two clusters we can see that our customers are split to two clusters `4497` and `3652`. Lets see these customers in a plot,

```
KM_customer <- mutate(cust_df_num,kmeans=KM_model$cluster, outcome=cust_df$outcome)
xtabs(~kmeans+outcome,KM_customer )
```

We can see that probability in claim in customers cluster 1 is %12 and in cluster 2 is 55%.

```
# plotting K-means result
plot(x=KM_customer$credit_score, y=KM_customer$annual_mileage,
                col = KM_customer$kmeans,
                main = "K-Means Clustering Results with K =2",xlab="credit_score",
                ylab="annual_mileage", pch = 20, cex = 1)

figure6 <- ggplot(KM_customer,aes(x=credit_score,y=annual_mileage,
                        color = factor(kmeans))) +
  geom_point()+
  ggtitle("K-Means Clustering Results with K =2")

ggsave("figure6.png", figure6,width = 4, heigh=2)
```

2.3- Hierarchical Clustering

In this part we are using hierarchical clustering for our dataset,

```
hcl_com_model <-hclust(cust_df_dis, method = "complete")
hcl_ave_model <-hclust(cust_df_dis, method = "average")
hcl_sin_model <-hclust(cust_df_dis, method = "single")
hcl_cen_model <-hclust(cust_df_dis, method = "centroid")
```

We used 4 linkage to fit the cluster model and here we plot these 4 clusters:

```
figure7 <- plot(hcl_com_model, main = "Complete Linkage",
xlab = "", sub = "", cex = .5, hang=-10)
rect.hclust(hcl_com_model , k = 2, border = 2:6)
abline(h = 3, col = 'red')

figure8 <-plot(hcl_ave_model, main = "Average Linkage",
xlab = "", sub = "", cex = .5, hang=-10)
rect.hclust(hcl_ave_model , k = 2, border = 2:6)
abline(h = 3, col = 'red')

figure9 <-plot(hcl_sin_model, main = "Single Linkage",
xlab = "", sub = "", cex = .5, hang=-10)
rect.hclust(hcl_sin_model , k = 2, border = 2:6)
abline(h = 3, col = 'red')

figure10 <- plot(hcl_cen_model, main = "Centroid Linkage",
xlab = "", sub = "", cex = .5, hang=-10)
rect.hclust(hcl_cen_model , k = 2, border = 2:6)
abline(h = 3, col = 'red')

ggsave("figure7.png", figure7,width = 4, heigh=2)
ggsave("figure8.png", figure8,width = 4, heigh=2)
ggsave("figure9.png", figure9,width = 4, heigh=2)
ggsave("figure10.png", figure10,width = 4, heigh=2)
```

we can see that it is hard to recognize data point in the last clusters so we use cutree() function to cut the dendogram based on number of clusters.

```
cut_hclc_model<-cutree(hcl_com_model,k=2)
cut_hcla_model<-cutree(hcl_ave_model,k=2)
cut_hcls_model<-cutree(hcl_sin_model,k=2)
cut_hclcen_model<-cutree(hcl_cen_model,k=2)
```

```
hcl_customers<- mutate(cust_df_num,Hcluster=cut_hclc_model, outcome=cust_df$outcome, kmeans=KM_
xtabs(~Hcluster+outcome,hcl_customers)
xtabs(~Hcluster+kmeans,hcl_customers)
```

```
figure11 <- ggplot(hcl_customers, aes(x=credit_score, y =annual_mileage, color = factor(Hcluste
figure11
ggsave("figure11.png", figure11,width = 4, heigh=2)
```

- using scaled features

```
plot(
  hclust(cust_df_dis_scale, method = "complete"),
main = "Hierarchical Clustering with Scaled Features")
cut_customer_scale <- cutree(hclust(cust_df_dis_scale, method = "complete"),k=2)
```

2.4- Gaussian Mixture Model clustering

model based clustering with 2 clusters:

```
Mclust_model <- Mclust(cust_df_num_scale, 2)
summary(Mclust_model)
names(Mclust_model)
map(Mclust_model$z)
```

```
Mclust_customers <- mutate(cust_df_num,Mcluster=Mclust_model$classification, outcome=cust_df$ou
xtabs(~Mcluster+outcome,Mclust_customers)
figure12<- ggplot(Mclust_customers, aes(x=credit_score, y =annual_mileage,
                                        color = factor(Mcluster))) +
  geom_point()+
  ggtitle("Gaussian Mixture Clustering Results with K =2")
figure12
ggsave("figure12.png", figure12,width = 4, heigh=2)
```

```
#plot(Mclust_model,  what = c("BIC"))
plot(Mclust_model,  what = c("classification"), main = "Mclust clustering with five components
```

2.5- PCA

2.5.1 Perform PCA

```
#perform PCA
PCA_model <-prcomp(cust_df_num,center= TRUE, scale = TRUE)

#we can use scaled dataset directly
PCA_model2 <-prcomp(cust_df_num_scale,center= TRUE)

#summary PCR model
names(PCA_model)
xtable::xtable(summary(PCA_model))
```

- since there are 15 features, we have 15 PCAs as well

```
#loading scores
PCA_model$rotation
dim(PCA_model$x)
```

- The first PCA accounts for `30\%` variation of data and second accounts for ‘11%’.

```
prop_var <- round(((PCA_model$sdev^2)/sum(PCA_model$sdev^2))*100,0)
prop_var
figure14 <- (plot(prop_var, main="PCA Percent Variance",
                 xlab="Principal component",type="b", col="blue"))
figure15 <- (plot(cumsum(prop_var),main="PCA Cumulative Percent Variance",
     xlab="Principal component",type="b", col="blue"))
ggsave("figure15.png", figure15 ,width = 4, heigh=2)
```

In rotation matrix we can see that score of 15 features for PCA1 is not equal and some of features such as `gender`, `race` and `postal_code` have not explained by PC1, on the other hand PC4 for these features has the most score, so lets draw some plots:

```
figure20<- biplot(PCA_model, scale=0)
```

- Using the two PCA1 and PCA2 to draw a plot

```
Cols <- function(vec) {
cols <- rainbow(length(unique(vec)))
return(cols[as.numeric(as.factor(vec))])
}
plot(PCA_model$x[,1],PCA_model$x[,2],col = Cols(colnames(cust_df_num)))
```

- Using the two PCA1 and PCA4 to draw a plot

```
plot(PCA_model$x[,1],PCA_model$x[,4],col = Cols(colnames(cust_df_num)))
```

- eigenvalues of dataset and diagonal of the covariance matrix of PCA result

```
eigen(cor(cust_df_num))$value
diag(var(PCA_model$x[,]))
```

```
PCA_model$rotation=-PCA_model$rotation
PCA_model$x=-PCA_model$x
biplot(PCA_model, scale=0)
```

2.5.2 Clustering with PCA

```
KM_model_PCA <- kmeans(PCA_model$x[, 1:5],centers=2, nstart=5)
hcl_model_PCA <- hclust(dist(PCA_model$x[, 1:5]))
cut_hcl_PCA <-cutree(hcl_model_PCA, k=2)
plot(hcl_model_PCA ,main = "Hier. Clust. on First Five Score Vectors")

hcl_customers<- mutate(hcl_customers,Hcluster_PCA=cut_hcl_PCA, Kmeans_PCA=KM_model_PCA$cluster
xtabs(~Hcluster_PCA+outcome,hcl_customers)
xtabs(~Hcluster_PCA+Kmeans_PCA,hcl_customers)
```

```
figure21 <- ggplot(hcl_customers,aes(x=credit_score,y=annual_mileage,
                        color = factor(Kmeans_PCA))) +
  geom_point()+
  ggtitle("K-Means Clustering After PCA")

figure22 <- ggplot(hcl_customers,aes(x=credit_score,y=annual_mileage,
                        color = factor(Hcluster_PCA))) +
  geom_point()+
  ggtitle("Hierarchical Clustering After PCA")

ggsave("figure21.png", figure21 ,width = 4, heigh=2)
ggsave("figure22.png", figure22 ,width = 4, heigh=2)
```

2.6- DBSCAN clustering

```
library(fpc) #computing density-based clustering
set.seed(133)
DBSCAN_model <- fpc::dbscan(cust_df_num, eps = .9, MinPts = 5)

names(DBSCAN_model)
DBSCAN_model$cluster
```