CKME136 - CAPSTONE

Datasafe Prediction & Recommendation

Mohammed Amir

April 7, 2017

```
Dinesafe = read.csv("D:/CAPSTONE/CAPSTONE/DATASET/Final_DineSafe.csv",
na.strings='NULL')

## select a subset of dataset
Dinesafe1 <- unique(Dinesafe[c(2,5:7)])

## Select unique rows
Dinesafe2 <-  unique(Dinesafe1)

nrow(Dinesafe2)

## [1] 2723

## Index the cuisine Type label
CUISINE_IDX <- function(CUISINE)
{
  if(CUISINE == "African")
  {
    print ("1")
  }
  else
  {
    if(CUISINE == "Bakeries")
    {
      print ("2")
    }
    else
    {
      if(CUISINE == "Bar")
      {
        print ("3")
      }
      else
      {
        if(CUISINE == "Cafe")
        {
          print ("4")
```

```
}
else
{
  if(CUISINE == "Caribbean")
  {
    print ("5")
  }
  else
  {
    if(CUISINE == "Deli")
    {
      print ("6")
    }
    else
    {
      if(CUISINE == "Dessert")
      {
        print ("7")
      }
      else
      {
        if(CUISINE == "European")
        {
          print ("8")
        }
        else
        {
          if(CUISINE == "Far Eastern")
          {
            print ("9")
          }
          else
          {
            if(CUISINE == "Mediterranean")
            {
              print ("10")
            }
            else
            {
              if(CUISINE == "Middle Eastern")
              {
                print ("11")
              }
              else
              {
                if(CUISINE == "North American")
                {
                  print ("12")
                }
                else
```

```
                      {
                        if(CUISINE == "Juicery")
                        {
                          print ("13")
                        }
                        else
                        {
                          if(CUISINE == "Pastries")
                          {
                            print ("14")
                          }
                          else
                          {
                            if(CUISINE == "South Asian")
                            {
                              print ("15")
                            }
                            else
                            {
                              if(CUISINE == "South East Asian")
                              {
                                print ("16")
                              }
                              else
                              {
                                if(CUISINE == "Latin American")
                                {
                                  print ("17")
                                }
                                else
                                {
                                  print ("0")
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }

  }
}
```

```
}

## Apply the Index function to cuisine type column
Dinesafe2$CUISINE_IDX <- mapply(CUISINE_IDX,Dinesafe2$CUISINE_TYPE)



Dinesafe2$African <- ifelse(Dinesafe2$CUISINE_TYPE == "African",1,0)
Dinesafe2$Bakeries <- ifelse(Dinesafe2$CUISINE_TYPE == "Bakeries",1,0)
Dinesafe2$Bar <- ifelse(Dinesafe2$CUISINE_TYPE == "Bar",1,0)
Dinesafe2$Cafe <- ifelse(Dinesafe2$CUISINE_TYPE == "Cafe",1,0)
Dinesafe2$Caribbean <- ifelse(Dinesafe2$CUISINE_TYPE == "Caribbean",1,0)
Dinesafe2$Deli <- ifelse(Dinesafe2$CUISINE_TYPE == "Deli",1,0)
Dinesafe2$Dessert <- ifelse(Dinesafe2$CUISINE_TYPE == "Dessert",1,0)
Dinesafe2$European <- ifelse(Dinesafe2$CUISINE_TYPE == "European",1,0)
Dinesafe2$FarEastern <- ifelse(Dinesafe2$CUISINE_TYPE == "Far Eastern",1,0)
Dinesafe2$Mediterranean <- ifelse(Dinesafe2$CUISINE_TYPE ==
"Mediterranean",1,0)
Dinesafe2$MidEastern <- ifelse(Dinesafe2$CUISINE_TYPE == "Middle
Eastern",1,0)
Dinesafe2$NAmerican <- ifelse(Dinesafe2$CUISINE_TYPE == "North American",1,0)
Dinesafe2$Juicery <- ifelse(Dinesafe2$CUISINE_TYPE == "Juicery",1,0)
Dinesafe2$Pastries <- ifelse(Dinesafe2$CUISINE_TYPE == "Pastries",1,0)
Dinesafe2$SouthAsian <- ifelse(Dinesafe2$CUISINE_TYPE == "South Asian",1,0)
Dinesafe2$SEastAsian <- ifelse(Dinesafe2$CUISINE_TYPE == "South East
Asian",1,0)
Dinesafe2$LAmerican <- ifelse(Dinesafe2$CUISINE_TYPE == "Latin American",1,0)



str(Dinesafe2)

## 'data.frame':    2723 obs. of  22 variables:
##  $ ESTABLISHMENT_ID: int  1222579 1222807 1223056 9000004 9000026 9000029
9000031 9000046 9000109 9000116 ...
##  $ REVIEW          : num  5 3.5 3 4 2.5 2.5 2.5 2.5 3 2 ...
##  $ VALUE           : num  1 1 2 1 2 2 2 2 2 2 ...
##  $ CUISINE_TYPE    : Factor w/ 17 levels "African","Bakeries",..: 16 9 8 8
8 8 8 8 3 4 ...
##  $ CUISINE_IDX     : chr  "15" "9" "8" "8" ...
##  $ African         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bakeries        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bar             : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Cafe            : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Caribbean       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deli            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Dessert         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ European        : num  0 0 1 1 1 1 1 1 0 0 ...
##  $ FarEastern      : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Mediterranean   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MidEastern      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NAmerican       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Juicery         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pastries        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SouthAsian      : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ SEastAsian      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LAmerican       : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
head(Dinesafe2)
```

```
##    ESTABLISHMENT_ID REVIEW VALUE CUISINE_TYPE CUISINE_IDX African Bakeries
## 1           1222579    5.0     1  South Asian          15       0        0
## 2           1222807    3.5     1  Far Eastern           9       0        0
## 9           1223056    3.0     2     European           8       0        0
## 13          9000004    4.0     1     European           8       0        0
## 18          9000026    2.5     2     European           8       0        0
## 23          9000029    2.5     2     European           8       0        0
##    Bar Cafe Caribbean Deli Dessert European FarEastern Mediterranean
## 1    0    0         0    0       0        0          0             0
## 2    0    0         0    0       0        0          1             0
## 9    0    0         0    0       0        1          0             0
## 13   0    0         0    0       0        1          0             0
## 18   0    0         0    0       0        1          0             0
## 23   0    0         0    0       0        1          0             0
##    MidEastern NAmerican Juicery Pastries SouthAsian SEastAsian LAmerican
## 1           0         0       0        0          1          0         0
## 2           0         0       0        0          0          0         0
## 9           0         0       0        0          0          0         0
## 13          0         0       0        0          0          0         0
## 18          0         0       0        0          0          0         0
## 23          0         0       0        0          0          0         0
```

```
#Dinesafe3 <- subset( Dinesafe2, select = -c( 1 ))
#Dinesafe3
#str(Dinesafe3)

Dinesafe2$CUISINE_IDX <- as.numeric(Dinesafe2$CUISINE_IDX)
str(Dinesafe2)
```

```
## 'data.frame':    2723 obs. of  22 variables:
##  $ ESTABLISHMENT_ID: int  1222579 1222807 1223056 9000004 9000026 9000029
9000031 9000046 9000109 9000116 ...
##  $ REVIEW          : num  5 3.5 3 4 2.5 2.5 2.5 2.5 3 2 ...
##  $ VALUE           : num  1 1 2 1 2 2 2 2 2 2 ...
##  $ CUISINE_TYPE    : Factor w/ 17 levels "African","Bakeries",..: 16 9 8 8
8 8 8 8 3 4 ...
##  $ CUISINE_IDX     : num  15 9 8 8 8 8 8 8 3 4 ...
##  $ African         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bakeries        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bar             : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Cafe            : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Caribbean       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deli            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Dessert         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ European        : num  0 0 1 1 1 1 1 1 0 0 ...
##  $ FarEastern      : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Mediterranean   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MidEastern      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NAmerican       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Juicery         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pastries        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SouthAsian      : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ SEastAsian      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LAmerican       : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
#Normalize the dataset feature
normalize <- function(x)
{
num <- x - min(x)
denom <- max(x) - min(x)
return (num/denom)
}

#Apply normalizeto dataset feature
Norm_RATING <- as.data.frame(lapply(Dinesafe2[,c(2,3,5)], normalize))
str(Norm_RATING)
```

```
## 'data.frame':    2723 obs. of  3 variables:
##  $ REVIEW     : num  1 0.625 0.5 0.75 0.375 0.375 0.375 0.375 0.5 0.25 ...
##  $ VALUE      : num  0 0 0.333 0 0.333 ...
##  $ CUISINE_IDX: num  0.882 0.529 0.471 0.471 0.471 ...
```

```
#str(Norm_Dinesafe1)
Norm_Dinesafe <- subset( Dinesafe2, select = -c( 2,3,5 ))
#str(Norm_Dinesafe)


Norm_Dinesafe5 <- cbind.data.frame(Norm_Dinesafe, Norm_RATING)

Norm_Dinesafe6 <- subset( Norm_Dinesafe5, select = -c( 1,2 ))

str(Norm_Dinesafe6)
```

```
## 'data.frame':    2723 obs. of  20 variables:
##  $ African      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bakeries     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Bar          : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ Cafe         : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Caribbean    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Deli         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Dessert      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ European     : num  0 0 1 1 1 1 1 1 0 0 ...
##  $ FarEastern   : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ Mediterranean: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MidEastern   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ NAmerican    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Juicery      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pastries     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SouthAsian   : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ SEastAsian   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LAmerican    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ REVIEW       : num  1 0.625 0.5 0.75 0.375 0.375 0.375 0.375 0.5 0.25
...
##  $ VALUE        : num  0 0 0.333 0 0.333 ...
##  $ CUISINE_IDX  : num  0.882 0.529 0.471 0.471 0.471 ...
```

```
nrow(Norm_Dinesafe5)
```

```
## [1] 2723
```

```
nrow(Norm_Dinesafe6)
```

```
## [1] 2723
```

```
#set.seed(9850)
#gp <- runif(nrow(Norm_Dinesafe6))
#Dinesafe4 <- Norm_Dinesafe6[order(gp),]
#head(Dinesafe4)

## create a feature
Dine_train <- Norm_Dinesafe6[1:2000,]
Dine_test <- Norm_Dinesafe6[2001:2723,]
nrow(Dine_train)
```

```
## [1] 2000
```

```
nrow(Dine_test)
```

```
## [1] 723
```

```
Dine_trainLabel <- Dinesafe2[1:2000,4]
Dine_testLabel <- Dinesafe2[2001:2723,4]

NROW(Dine_trainLabel)
```

```
## [1] 2000
```

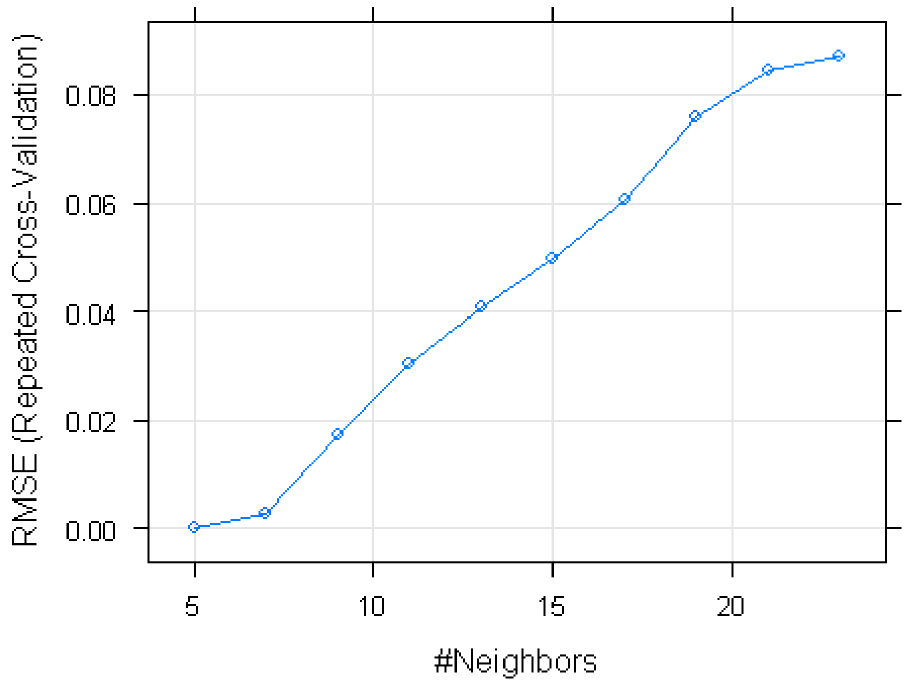
```
NROW(Dine_testLabel)
```

```
## [1] 723
```

```
# Determine best K value in KNN Crosss Validation
set.seed(3333)
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
```

```
knn_fit <- train(CUISINE_IDX~., data = Dine_train, method = "knn",
trControl=trctrl, preProcess = c("center", "scale"),tuneLength = 10)

knn_fit
plot(knn_fit)
```



```
model <- knn(train = Dine_train, test = Dine_test, cl = Dine_trainLabel, k =
5)
model

## 17 Levels: African Bakeries Bar Cafe Caribbean Deli Dessert ... South East
Asian

table (Dine_testLabel, model)

##                       model
## Dine_testLabel        African Bakeries Bar Cafe Caribbean Deli Dessert
##   African                   5        0   0    0         0    0       0
##   Bakeries                  0        2   0    0         0    0       0
##   Bar                       0        0   9    0         0    0       0
##   Cafe                      0        0   0  203         0    0       0
##   Caribbean                 0        0   0    0         6    0       0
##   Deli                      0        0   0    0         0  125       0
##   Dessert                   0        0   0    0         0    0      12
##   European                  0        0   0    0         0    0       0
##   Far Eastern               0        0   0    0         0    0       0
##   Juicery & Smoothies       0        0   0    0         0    0       0
##   Latin American            0        0   0    0         0    0       0
##   Mediterranean             0        0   0    0         0    0       0
##   Middle Eastern            0        0   0    0         0    0       0
##   North American            0        0   0    0         0    0       0
##   Pastries                  0        0   0    0         0    0       0
##   South Asian               0        0   0    0         0    0       0
##   South East Asian          0        0   0    0         0    0       0
##                       model
## Dine_testLabel        European Far Eastern Juicery & Smoothies
##   African                    0           0                   0
##   Bakeries                   0           0                   0
##   Bar                        0           0                   0
##   Cafe                       0           0                   0
##   Caribbean                  0           0                   0
##   Deli                       0           0                   0
##   Dessert                    0           0                   0
##   European                  96           0                   0
##   Far Eastern                0          60                   0
##   Juicery & Smoothies        0           0                  21
##   Latin American             0           0                   0
##   Mediterranean              0           0                   0
##   Middle Eastern             0           0                   0
##   North American             0           0                   0
##   Pastries                   0           0                   0
##   South Asian                0           0                   0
##   South East Asian           0           0                   0
```

```
##                       model
## Dine_testLabel        Latin American Mediterranean Middle Eastern
##   African                          0             0              0
##   Bakeries                         0             0              0
##   Bar                              0             0              0
##   Cafe                             0             0              0
##   Caribbean                        0             0              0
##   Deli                             0             0              0
##   Dessert                          0             0              0
##   European                         0             0              0
##   Far Eastern                      0             0              0
##   Juicery & Smoothies              0             0              0
##   Latin American                  18             0              0
##   Mediterranean                    0            31              0
##   Middle Eastern                   0             0              4
##   North American                   0             0              0
##   Pastries                         0             0              0
##   South Asian                      0             0              0
##   South East Asian                 0             0              0
##                       model
## Dine_testLabel        North American Pastries South Asian South East Asian
##   African                          0        0           0                0
##   Bakeries                         0        0           0                0
##   Bar                              0        0           0                0
##   Cafe                             0        0           0                0
##   Caribbean                        0        0           0                0
##   Deli                             0        0           0                0
##   Dessert                          0        0           0                0
##   European                         0        0           0                0
##   Far Eastern                      0        0           0                0
##   Juicery & Smoothies              0        0           0                0
##   Latin American                   0        0           0                0
##   Mediterranean                    0        0           0                0
##   Middle Eastern                   0        0           0                0
##   North American                  91        0           0                0
##   Pastries                         0       13           0                0
##   South Asian                      0        0           7                0
##   South East Asian                 0        0           0               20
```

```
## Accurry where predicted value is not equal to given label
sum(model != Dine_testLabel)
```

```
## [1] 0
```

```
confusion <- confusionMatrix(model, Dine_testLabel )
plot <- ggplot(as.data.frame(as.table(confusion)))



## RECOMMENDATION
library(class)
library(caret)


#load dataset
recommender  <- Dine_test

## Create a matrix using euclidean distance
distances <- as.matrix(dist(recommender , method="euclidean"))

k.nearest.neighbors <- function(i, recommender, k = 5)
{
  ordered.neighbors <- order(recommender[i, ])
  return(ordered.neighbors[2:(k + 1)])
}

seen.probability <- function(cuisine, restaurant, recommender, distances, k =
25)
{
  neighbors <- k.nearest.neighbors(which(row.names(recommender) ==
restaurant), distances, k)
  return(mean(recommender[neighbors, cuisine]))
}
```

```r
most.probable.recommend <- function(cuisine, recommender, distances, k = 25)
{
  probabilities <- rep(0, nrow(recommender))
  for (i in 1:nrow(recommender)) { # For each restaurant
    if (recommender[i,cuisine] == 1) {
      next
    }
    probabilities[i] <- seen.probability(cuisine, row.names(recommender)[i],
recommender, distances, k)
  }
  return(order(probabilities, decreasing=T))
}

cuisine <- "African"
listing <- most.probable.recommend(cuisine, recommender, distances)
rownames(recommender)[listing[1:3]]
```

```
## [1] "12970" "12996" "13057"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.