

# CKME136 - CAPSTONE

## Dinesafe 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 normalize to 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 ...
```

```
head(Norm_Dinesafe6)
```

```
##      African Bakeries Bar Cafe Caribbean Deli Dessert European FarEastern
## 1          0          0  0  0          0  0          0          0          0
## 2          0          0  0  0          0  0          0          0          1
## 9          0          0  0  0          0  0          0          1          0
## 13         0          0  0  0          0  0          0          1          0
## 18         0          0  0  0          0  0          0          1          0
## 23         0          0  0  0          0  0          0          1          0
##      Mediterranean MidEastern NAmerican Juicery Pastries SouthAsian
## 1              0              0          0          0          0          1
## 2              0              0          0          0          0          0
## 9              0              0          0          0          0          0
## 13             0              0          0          0          0          0
## 18             0              0          0          0          0          0
## 23             0              0          0          0          0          0
##      SEastAsian LAmerican REVIEW      VALUE CUISINE_IDX
## 1              0          0  1.000 0.0000000  0.8823529
## 2              0          0  0.625 0.0000000  0.5294118
## 9              0          0  0.500 0.3333333  0.4705882
## 13             0          0  0.750 0.0000000  0.4705882
## 18             0          0  0.375 0.3333333  0.4705882
## 23             0          0  0.375 0.3333333  0.4705882
```

```
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 <- Dinesafe4[1:2000,]
Dine_test <- Dinesafe4[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

```

k-Nearest Neighbors  
 2000 samples  
 19 predictor

Pre-processing: centered (19), scaled (19)  
 Resampling: Cross-Validated (10 fold, repeated 3 times)  
 Summary of sample sizes: 1799, 1800, 1800, 1802, 1801, 1799, ...  
 Resampling results across tuning parameters:

k	RMSE	Rsquared
5	0.0004542904	0.9999357
7	0.0021647030	0.9997183
9	0.0238553329	0.9811620
11	0.0265061535	0.9791231
13	0.0267406724	0.9789902
15	0.0267336376	0.9790015
17	0.0297128012	0.9762035
19	0.0385952570	0.9585403
21	0.0484653724	0.9406148
23	0.0708054457	0.8850138

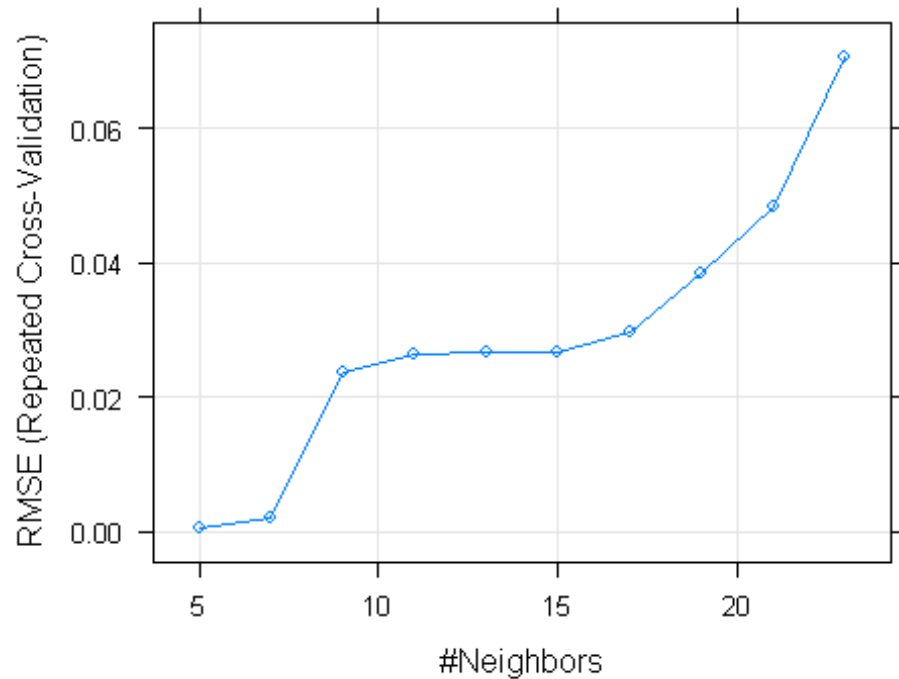
RMSE was used to select the optimal model using the smallest value.  
 The final value used for the model was k = 5.

```

# Plot Crosss Validation graph
plot(knn_fit)

```





```
sqrt(2723)
```

```
## [1] 52.18237
```

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

```
## [1] Cafe           Cafe           North American North American
## [5] Cafe           North American Cafe           Cafe
## [9] Cafe           North American Cafe           Cafe
## [13] Cafe           North American Cafe           Cafe
## [17] Cafe           Cafe           Cafe           European
## [21] Cafe           Cafe           Cafe           Cafe
## [25] Cafe           Cafe           Cafe           Deli
## [29] Cafe           Cafe           Cafe           Cafe
## [33] Cafe           European       Cafe           Cafe
## [37] Cafe           Cafe           North American European
## [41] Cafe           European       Cafe           Cafe
## [45] Cafe           Cafe           North American Cafe
## [49] Cafe           Deli           Cafe           Cafe
## [53] Cafe           Cafe           North American Cafe
## [57] Cafe           Cafe           Cafe           Cafe
## [61] European       Cafe           Cafe           Cafe
## [65] Cafe           Cafe           Cafe           Cafe
## [69] North American Cafe           Far Eastern    Cafe
## [73] Cafe           Far Eastern    Cafe           Cafe
```

## [77]	Cafe	Cafe	Cafe	Cafe
## [81]	Cafe	Cafe	Cafe	Cafe
## [85]	Cafe	Cafe	Cafe	Deli
## [89]	Cafe	Cafe	European	Cafe
## [93]	Cafe	Cafe	Deli	Cafe
## [97]	Cafe	Cafe	Cafe	Cafe
## [101]	Cafe	Cafe	Cafe	Cafe
## [105]	Cafe	Cafe	Cafe	Cafe
## [109]	Cafe	Cafe	Cafe	Cafe
## [113]	Cafe	Cafe	Cafe	Cafe
## [117]	Cafe	Cafe	European	Cafe
## [121]	Cafe	North American	Cafe	Cafe
## [125]	Cafe	Cafe	Cafe	Cafe
## [129]	Cafe	Cafe	Cafe	Cafe
## [133]	Cafe	Cafe	Cafe	Cafe
## [137]	Cafe	Cafe	Cafe	European
## [141]	Deli	Cafe	Cafe	Cafe
## [145]	Deli	Cafe	Cafe	Cafe
## [149]	North American	Cafe	Cafe	Cafe
## [153]	Cafe	Cafe	Cafe	Cafe
## [157]	Cafe	Cafe	Cafe	Cafe
## [161]	Cafe	Cafe	European	Cafe
## [165]	North American	Cafe	Cafe	Cafe
## [169]	Cafe	Cafe	European	Cafe
## [173]	European	Cafe	Cafe	Cafe
## [177]	Cafe	North American	Cafe	Cafe
## [181]	Cafe	Cafe	Cafe	Cafe
## [185]	European	Cafe	Cafe	Cafe
## [189]	Cafe	Cafe	Cafe	Cafe
## [193]	Cafe	Cafe	European	European
## [197]	Cafe	Cafe	North American	Cafe
## [201]	Cafe	North American	Cafe	Cafe
## [205]	Cafe	Cafe	Cafe	Deli
## [209]	North American	Cafe	Cafe	Cafe
## [213]	Cafe	Cafe	North American	Cafe
## [217]	Cafe	Cafe	Cafe	Cafe
## [221]	Cafe	Cafe	Cafe	Cafe
## [225]	Cafe	Cafe	Cafe	Cafe
## [229]	Cafe	Cafe	European	North American
## [233]	Cafe	Cafe	Deli	Cafe
## [237]	Cafe	Cafe	Cafe	Cafe
## [241]	Cafe	Cafe	Deli	Cafe
## [245]	Cafe	Cafe	North American	Cafe
## [249]	Cafe	Cafe	Cafe	Cafe
## [253]	Cafe	Cafe	Far Eastern	North American
## [257]	Cafe	Cafe	Cafe	Deli
## [261]	Deli	Deli	Cafe	Cafe
## [265]	North American	Cafe	Cafe	Cafe
## [269]	North American	North American	Cafe	Cafe
## [273]	Cafe	Cafe	Cafe	Cafe

## [277]	Cafe	Cafe	Cafe	Cafe
## [281]	Deli	Cafe	Cafe	Cafe
## [285]	Cafe	Cafe	Cafe	Cafe
## [289]	Cafe	Cafe	Cafe	Deli
## [293]	Cafe	Cafe	Cafe	Cafe
## [297]	Cafe	Cafe	Cafe	Cafe
## [301]	Cafe	Cafe	Cafe	North American
## [305]	Cafe	Cafe	Cafe	Cafe
## [309]	Cafe	Cafe	Cafe	Cafe
## [313]	Cafe	North American	Cafe	Cafe
## [317]	North American	Cafe	Cafe	Cafe
## [321]	Cafe	Cafe	Cafe	North American
## [325]	Far Eastern	Cafe	Cafe	Cafe
## [329]	Cafe	Cafe	Cafe	Cafe
## [333]	Far Eastern	Cafe	Cafe	Cafe
## [337]	Cafe	Cafe	Cafe	North American
## [341]	Cafe	Cafe	Cafe	Cafe
## [345]	Cafe	Cafe	Cafe	Cafe
## [349]	Cafe	Cafe	Cafe	Cafe
## [353]	Cafe	Cafe	Cafe	Cafe
## [357]	Cafe	Cafe	Cafe	Cafe
## [361]	Cafe	Cafe	Cafe	European
## [365]	European	Cafe	Deli	Cafe
## [369]	Cafe	Far Eastern	Cafe	European
## [373]	Cafe	Cafe	Cafe	Cafe
## [377]	Cafe	Cafe	European	Cafe
## [381]	Cafe	Cafe	Cafe	Cafe
## [385]	South East Asian	Cafe	Cafe	Cafe
## [389]	Far Eastern	Cafe	Cafe	Cafe
## [393]	Far Eastern	Cafe	Cafe	Cafe
## [397]	Cafe	Cafe	North American	Cafe
## [401]	North American	Deli	Cafe	Cafe
## [405]	Cafe	Cafe	Cafe	Cafe
## [409]	Cafe	Cafe	Cafe	Cafe
## [413]	Cafe	Cafe	Cafe	Cafe
## [417]	North American	Cafe	Cafe	Cafe
## [421]	Cafe	North American	Cafe	Cafe
## [425]	Cafe	Cafe	Cafe	Cafe
## [429]	Deli	Cafe	Cafe	Cafe
## [433]	Cafe	North American	Cafe	Deli
## [437]	Cafe	North American	Cafe	Cafe
## [441]	Cafe	Cafe	Cafe	Cafe
## [445]	European	Cafe	Deli	Cafe
## [449]	Cafe	Cafe	Cafe	Cafe
## [453]	Deli	North American	Cafe	Cafe
## [457]	Cafe	Cafe	Cafe	Cafe
## [461]	Cafe	Cafe	Deli	Cafe
## [465]	Cafe	Cafe	Bar	Cafe
## [469]	Cafe	Cafe	Cafe	Cafe
## [473]	European	Cafe	Cafe	Cafe

## [477]	Cafe	Cafe	Cafe	Cafe
## [481]	Cafe	North American	Cafe	Deli
## [485]	Cafe	Cafe	Cafe	Cafe
## [489]	Cafe	Cafe	North American	Cafe
## [493]	Cafe	Cafe	Cafe	Cafe
## [497]	Cafe	Cafe	Cafe	Cafe
## [501]	Cafe	North American	Cafe	Cafe
## [505]	Cafe	Cafe	Cafe	Cafe
## [509]	Cafe	Cafe	Cafe	Deli
## [513]	Cafe	Cafe	Cafe	North American
## [517]	Cafe	Cafe	Cafe	Cafe
## [521]	Deli	Cafe	Cafe	Deli
## [525]	Cafe	Cafe	Cafe	Cafe
## [529]	North American	Cafe	Deli	Cafe
## [533]	Cafe	Cafe	Cafe	Cafe
## [537]	Deli	Cafe	Cafe	Cafe
## [541]	North American	European	Cafe	Cafe
## [545]	Cafe	Cafe	Cafe	Far Eastern
## [549]	Cafe	Cafe	North American	Cafe
## [553]	Cafe	Cafe	Cafe	Cafe
## [557]	Cafe	Cafe	Cafe	Cafe
## [561]	Cafe	Cafe	Cafe	Cafe
## [565]	Cafe	North American	North American	Bar
## [569]	Cafe	Cafe	Cafe	Deli
## [573]	Cafe	Cafe	Cafe	Cafe
## [577]	Cafe	Cafe	Cafe	Cafe
## [581]	Cafe	Cafe	Cafe	Cafe
## [585]	Cafe	Cafe	Cafe	Cafe
## [589]	Cafe	Cafe	Cafe	Cafe
## [593]	North American	Cafe	Cafe	Cafe
## [597]	Cafe	Cafe	Cafe	European
## [601]	Cafe	Cafe	Cafe	Cafe
## [605]	Cafe	European	Cafe	Cafe
## [609]	Cafe	Cafe	Cafe	Deli
## [613]	Cafe	Cafe	Cafe	Cafe
## [617]	European	Cafe	Cafe	Cafe
## [621]	Cafe	Cafe	Cafe	Cafe
## [625]	Cafe	Cafe	Cafe	North American
## [629]	Cafe	Cafe	Cafe	Cafe
## [633]	Cafe	Cafe	Cafe	Cafe
## [637]	Cafe	Cafe	Deli	Cafe
## [641]	Deli	Cafe	Cafe	Cafe
## [645]	Cafe	Cafe	Cafe	Cafe
## [649]	North American	Cafe	Deli	Cafe
## [653]	Cafe	Cafe	Cafe	Cafe
## [657]	Cafe	Cafe	Cafe	Cafe
## [661]	Cafe	Cafe	Cafe	Cafe
## [665]	Cafe	Cafe	Cafe	Cafe
## [669]	Cafe	Cafe	European	Cafe
## [673]	Cafe	North American	European	Cafe

```
## [677] Cafe          Cafe          Cafe          Cafe
## [681] Cafe          European        Cafe          Cafe
## [685] Cafe          Cafe          Cafe          Cafe
## [689] European        Cafe          Cafe          Cafe
## [693] Deli           Cafe          North American Cafe
## [697] Cafe          Cafe          Cafe          Cafe
## [701] European        Cafe          North American Cafe
## [705] Deli           Cafe          North American Cafe
## [709] Cafe          Deli          Cafe          Cafe
## [713] Cafe          Cafe          Cafe          Cafe
## [717] Cafe          Cafe          Deli          Cafe
## [721] Cafe          Cafe          Cafe
## 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          0          0  0   5          0   0          0
## Bakeries          0          0  0   2          0   0          0
## Bar              0          0  0   9          0   0          0
## Cafe              0          0  1 163          0  13          0
## Caribbean         0          0  0   5          0   0          0
## Deli              0          0  1 102          0   4          0
## Dessert           0          0  0   9          0   1          0
## European          0          0  0  80          0   3          0
## Far Eastern       0          0  0  49          0   1          0
## Juicery & Smoothies 0          0  0  16          0   3          0
## Latin American    0          0  0  17          0   0          0
## Mediterranean     0          0  0  27          0   2          0
## Middle Eastern    0          0  0   3          0   0          0
## North American    0          0  0  76          0   3          0
## Pastries          0          0  0  12          0   1          0
## South Asian        0          0  0   6          0   1          0
## South East Asian  0          0  0  13          0   4          0
##          model
## Dine_testLabel European Far Eastern Juicery & Smoothies
## African          0          0          0
## Bakeries          0          0          0
## Bar              0          0          0
## Cafe              9          2          0
## Caribbean         1          0          0
## Deli              6          1          0
## Dessert           1          0          0
## European          2          2          0
## Far Eastern       3          2          0
## Juicery & Smoothies 1          0          0
## Latin American    0          0          0
## Mediterranean     1          0          0
```

```

## Middle Eastern          0          0          0
## North American         5          2          0
## Pastries               0          0          0
## South Asian            0          0          0
## South East Asian       1          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   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 North American Pastries South Asian South East Asian
## African          0          0          0          0
## Bakeries         0          0          0          0
## Bar              0          0          0          0
## Cafe            15          0          0          0
## Caribbean        0          0          0          0
## Deli            11          0          0          0
## Dessert          1          0          0          0
## European         9          0          0          0
## Far Eastern      5          0          0          0
## Juicery & Smoothies 1          0          0          0
## Latin American   1          0          0          0
## Mediterranean    1          0          0          0
## Middle Eastern   1          0          0          0
## North American   4          0          0          1
## Pastries         0          0          0          0
## South Asian      0          0          0          0
## South East Asian 2          0          0          0
##
## Accuray where predicted value is not equal to given label
sum(model != Dine_testLabel)
## [1] 548

```

```

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

# Put `iris.testLabels` in a data frame
DineTestLabels <- data.frame(Dine_testLabel)

# Merge `iris_pred` and `iris.testLabels`
merge <- data.frame(model, Dine_testLabel)

# Specify column names for `merge`
names(merge) <- c("Predicted Cuisine", "Observed Cuisine")

# Inspect `merge`
head(merge,10)

##      Predicted Cuisine Observed Cuisine
## 1              Cafe    North American
## 2              Cafe    North American
## 3    North American              Dessert
## 4    North American              Cafe
## 5              Cafe              Dessert
## 6    North American              European
## 7              Cafe              Deli
## 8              Cafe              Cafe
## 9              Cafe              European
## 10    North American    Latin American

## RECOMMENDATION

library(class)
library(caret)
#library(lattice)
#library(ggplot2)

#Load dataset
recommender <- Dine_test

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

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

## calculate probability of the closest restaurant
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]))
}

## Predict a recommendation based on cuisine input, recommender matrix,
distance between restaurants and K value
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] "10465" "7501" "2413"

```

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