Dinesafe Restaurant and Food Recommender System

The first section of the capstone project is an exploratory & relevance analysis of the dataset to determine the quality of the data and understand the relevance of it to the objective of the project.

So far the below analysis was performed on the dataset.

Step 1. Download Dinesafe Food Hygiene Inspection dataset from city of Toronto public data

Step 2. Convert the dataset from xml to csv file format

Step 3. Remove non-public food serving premises such as shops, institutions, schools, food processes plants, hospitals etc from the dataset

Step 4. Extract the address column from the dataset and feed it to google ggmap in order to download full address, district, postal code for each premises. The primary constraint of this activity is that Google has a limit of 2500 address queries and not all queries work successfully the first time.

Dinesafe dataset has over 10,000 unique addresses with close to 20,000 unique establishments. I was able to get the address data after multiple attempts and massage the data in excel to fit the need.

```
library(ggmap)
Address <- read.csv(file="F:/Dinesafe.csv", header=TRUE, sep=",")
Address <- as.character(Address$ESTABLISHMENT_ADDRESS)
Address <- geocode(paste0(x, ", Toronto"), output = "more")
write.csv(Address, file = "F:/Address.csv")
```

Step 5. Dinesafe data set has no customer rating or review information and as this information was vital in order to build a recommender system I have added three columns to the dataset “Review”, “Value” and “Cuisine Type” and start populating this data for each establishment manually from yelp.ca and traveladvise.ca website.

- Review : Service rating from 1 to 5, five being the highest
- Value : Dollar value as a measure of affordability from 1 to 5, five being the highest
- Cuisine Type : Food speciality, this is divided in to sub-regions of the world as below
  - ❖ African
  - ❖ Bakeries
  - ❖ Bar/Pub
  - ❖ Café
  - ❖ Caribbean
  - ❖ Deli
  - ❖ Dessert
  - ❖ European
  - ❖ Far Eastern
  - ❖ Juicery & Smoothies
  - ❖ Latin American
  - ❖ Mediterranean
  - ❖ Middle Eastern
  - ❖ North American
  - ❖ Pastries
  - ❖ South Asian
  - ❖ South East Asian

This exercise is challenging and time consuming because it is a manual process and there are still outstanding establishments that need to be updated. Yelp.ca only has a dataset for Montreal and Kitchener and it was difficult to find similar dataset from different organizations.

Step 6. Load Dinesafe and Address datasets in R Studio

```
Dinesafe = read.csv("D:/CAPSTONE/data/DineSafe_02162017.csv")
Address  = read.csv("D:/CAPSTONE/data/ADDRESS_02262017.csv")
```

Step 7. Review the data structure (str), dimension (dim) and dataset summary (summary)

Step 8. Merge Dinesafe and Address datasets based on establishment id column

```
Dinesafe <- merge(Dinesafe,Address,by="ESTABLISHMENT_ID")
```

Step 9. Remove redundant columns from the dataset

```
# Remove COURT_OUTCOME and AMOUNT_FINED Columns from Dinesafe dataset
Dinesafe <- subset(Dinesafe, select = -c(ROW_ID, COURT_OUTCOME,AMOUNT_FINED) )

# Remove Establishement Name from Address dataset to avoid duplicate rows
Dinesafe <- subset(Dinesafe, select = -c(ESTABLISHMENT_NAME.y, ESTABLISHMENT_ADDRESS) )

# Rename ESTABLISHMENT_NAME.x column name to ESTABLISHMENT_NAME
colnames(Dinesafe)[colnames(Dinesafe) == 'ESTABLISHMENT_NAME.x'] <- 'ESTABLISHMENT_NAME'
```

Step 10. Convert "VALUE" column data type from factor to numeric

```
as.numeric(as.character(Dinesafe$VALUE))
```

Step 11. Convert "INSPECTION_DATE" column from factor to numeric using "**lubridate**" package to prevent the data from being changed to NA

```
dmy(as.character(Dinesafe$INSPECTION_DATE))
```

Step 12. Show completenss of the dataset

```
Dinesafe_CS  <- Dinesafe[complete.cases(Dinesafe),]
nrow(Dinesafe_CS)
```

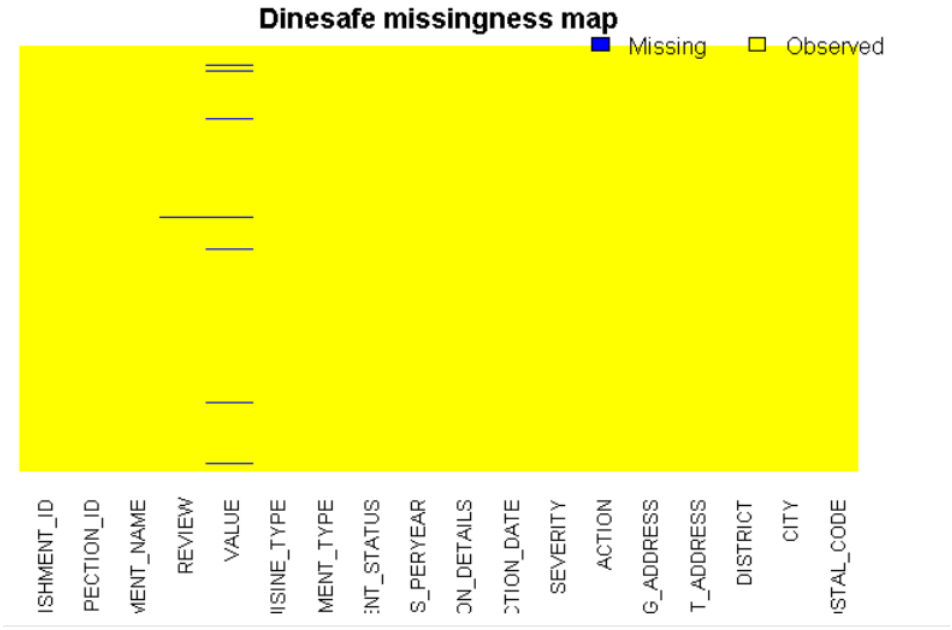
Step 13. Show missingness of the data with NA values

```
Dinesafe_NA  <- Dinesafe[!complete.cases(Dinesafe),]
nrow(Dinesafe_NA)
```

Step 14. Quantify missing values and plot missingness map

```
#Quantify missing values
apply(Dinesafe, 2, function(x) sum(is.na(x)))

# Plot missingness map
missmap(Dinesafe, col = c("Blue","Yellow"), y.cex = 0.8, x.cex = 0.8, legend = TRUE, rank.order = "False" ,main
= "Dinesafe missingness map", y.labels = NULL,y.at = NULL)
```



Step 15. Impute REVIEW with Mean Review Value for each missing review value based cuisine type

```
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="African"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="African"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Bakeries"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Bakeries"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Bar"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Bar"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Cafe"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Cafe"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Caribbean"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Caribbean"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Deli"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Deli"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Dessert"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Dessert"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="European"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="European"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Far Eastern"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Far Eastern"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Pastries"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Pastries"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="South Asian"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="South Asian"], na.rm=TRUE)
```

```
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="South East Asian"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="South East Asian"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Latin American"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Latin American"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Mediterranean"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Mediterranean"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Middle Eastern"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Middle Eastern"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="North American"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="North American"], na.rm=TRUE)

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE_TYPE=="Juicery & Smoothies"] =
mean(Dinesafe$REVIEW[Dinesafe$CUISINE_TYPE=="Juicery & Smoothies"], na.rm=TRUE)
```

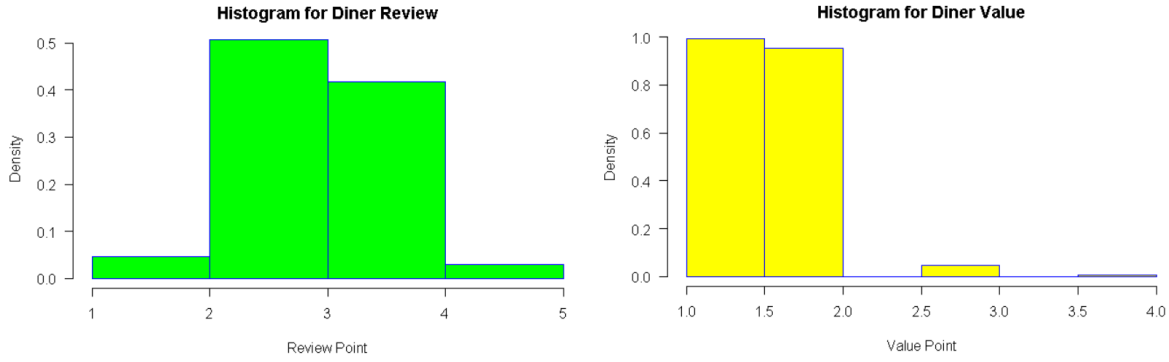
Step 16. Impute Dinesafe$VALUE with Mean Value for each missing value based cuisine type

```
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="African"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="African"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Bakeries"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Bakeries"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Bar"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Bar"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Cafe"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Cafe"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Caribbean"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Caribbean"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Deli"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Deli"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Dessert"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Dessert"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="European"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="European"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Far Eastern"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Far Eastern"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Pastries"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Pastries"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="South Asian"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="South Asian"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="South East Asian"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="South East Asian"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Latin American"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Latin American"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Mediterranean"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Mediterranean"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Middle Eastern"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Middle Eastern"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="North American"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="North American"], na.rm=TRUE)
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE_TYPE=="Juicery & Smoothies"] =
mean(Dinesafe$VALUE[Dinesafe$CUISINE_TYPE=="Juicery & Smoothies"], na.rm=TRUE)
```

STEP 17. Plot histogram graph

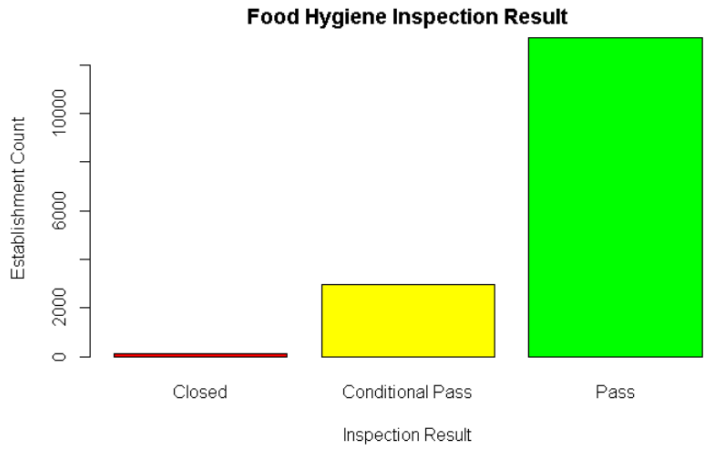
```
hist(Dinesafe$REVIEW, xlab = "Review Point", main="Histogram for Diner Review",
  border="blue", col="GREEN", las=1, breaks=5,prob = TRUE)

hist(Dinesafe$VALUE, xlab = "Value Point", main="Histogram for Diner Value",
    border="blue", col="YELLOW", las=1, breaks=5,prob = TRUE
```



STEP 18. Plot barplot graph

```
status <- table(Dinesafe$ESTABLISHMENT_STATUS)
barplot(status, main="Food Hygiene Inspection Result",xlab="Inspection Result", ylab="Establishment
Count", col=c("red","yellow","green"), beside=TRUE)
```



What is NEXT?

The next step of the analysis is to understand the relationship between multiple attributes within the dataset for a multivariate analysis as well as understanding NULL values and its imputation in the severity and action columns.

Also I will continue to update the REVIEW, VALUE and CUISINE TYPE columns in the dataset for the remaining restaurants in the dataset.