﻿#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""

Created on Thu Nov 29 13:15:39 2018

@author: junkangzhang

"""

import pandas as pd

business = pd.read\_csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/new\_business.csv')

business\_hours = pd.read\_csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/yelp\_business\_hours.csv')

review\_cleaned = pd.read\_csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/review\_cleaned.csv')

########## Variation manully

### high review high stars

merge1 = pd.merge(business,review\_cleaned,on='business\_id')

hshr = merge1[merge1.review\_count\_x >= 30][merge1.stars > 4]

volatility1\_1 = hshr.groupby('business\_id')['mean\_star'].var().reset\_index(name ='volatility')

hshr = pd.merge(hshr,volatility1\_1,on='business\_id')

print(hshr['volatility'].mean())

### high review low stars

merge1 = pd.merge(business,review\_cleaned,on='business\_id')

lshr = merge1[merge1.review\_count\_x >= 30][merge1.stars <= 2.5]

volatility1\_2 = lshr.groupby('business\_id')['mean\_star'].var().reset\_index(name ='volatility')

lshr = pd.merge(lshr,volatility1\_2,on='business\_id')

print(lshr['volatility'].mean())

### high review mid stars

merge1 = pd.merge(business,review\_cleaned,on='business\_id')

mshr = merge1[merge1.review\_count\_x >= 30][merge1.stars > 2.5][merge1.stars <= 4]

volatility1\_3 = mshr.groupby('business\_id')['mean\_star'].var().reset\_index(name ='volatility')

mshr = pd.merge(mshr,volatility1\_3,on='business\_id')

print(mshr['volatility'].mean())

########## Variation k-mean

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_std = scaler.fit\_transform(business[['stars','review\_count']])

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=3)

model = kmeans.fit(X\_std)

labels = model.predict(X\_std)

# Plot cluster membership

from matplotlib import pyplot

pyplot.scatter(business['stars'], business['review\_count'], c=labels, cmap='rainbow')

cluster\_list = labels.tolist()

cluster\_df = pd.DataFrame({'cluster':cluster\_list})

business['cluster'] = cluster\_df['cluster']

#

volatility2 = review\_cleaned.groupby('business\_id')['mean\_star'].var().reset\_index(name ='volatility')

business\_kmean = pd.merge(business,volatility2,on='business\_id')

kmean\_volatility = business\_kmean.groupby('cluster',as\_index=False)['volatility'].mean()

print(kmean\_volatility)

tosee= business\_kmean[['business\_id','review\_count','stars','cluster']]

tosee.to\_csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/business\_cluster.csv')

####################

business\_large\_review = business[business.review\_count > 20]

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_std = scaler.fit\_transform(business\_large\_review[['stars','review\_count']])

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=3)

model = kmeans.fit(X\_std)

labels = model.predict(X\_std)

# Plot cluster membership

from matplotlib import pyplot

pyplot.scatter(business\_large\_review['stars'], business\_large\_review['review\_count'], c=labels, cmap='rainbow')

cluster\_list = labels.tolist()

cluster\_df = pd.DataFrame({'cluster':cluster\_list})

business\_large\_review['cluster'] = cluster\_df['cluster']

#

volatility3 = review\_cleaned.groupby('business\_id')['mean\_star'].var().reset\_index(name ='volatility')

business\_kmean = pd.merge(business\_large\_review,volatility3,on='business\_id')

kmean\_volatility = business\_kmean.groupby('cluster',as\_index=False)['volatility'].mean()

print(kmean\_volatility)

########## Clean review in R

########## Pick attributes from review

merged = pd.merge(business,business\_hours,on='business\_id')

merged = pd.merge(merged,review\_cleaned,on='business\_id')

########## drop neighborhood

merged = merged.drop(['neighborhood'],axis=1)

merged = merged.drop(['Unnamed: 0'],axis=1)

merged = merged.dropna()

########## modify business hours

merged = merged.assign(is\_business\_hour

########## filter isopen = 1

new\_houseprices[new\_houseprices.PoolArea > 0]

############################ section 2

review = pd.read\_csv('/Users/junkangzhang/Downloads/yelp\_review.csv')

bus\_review = pd.merge(tosee,review,on='business\_id')

bus\_review1 = bus\_review.sample(n=100000,random\_state = 1)

bus\_review1.to\_csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/business\_review\_100000.csv')

R

library(tidyr)

library(dplyr)

business = read.csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/yelp\_business.csv')

#business = na.omit(business)

us\_states =c('AL','AK','AZ','AR','CA','CO','CT','DE','FL','GA','HI','ID','IL',

'IN','IA','KS','KY','LA','ME','MD','MA','MI','MN','MS','MO','MT','NE',

'NV','NH','NJ','NM','NY','NC','ND','OH','OK','OR','PA','RI','SC','TN',

'TX','UT','VT','VA','WA','WV','WI','WY')

ca\_states =c("AB", "BC","LB", "MB", "NB", "NF", "NS", "NU", "NW", "ON", "PE", "QC", "SK", "YU")

library(dplyr) r

business=business %>% mutate(is\_us = (state %in% us\_states))

business=business %>% mutate(is\_ca = (state %in% ca\_states))

write.csv(business,'/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/new\_business.csv',row.names = FALSE)

new\_business = read.csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/new\_business.csv')

review\_withoutext = read.csv('/Users/junkangzhang/Downloads/yelp\_review.csv',colClasses = c(NA,NA,NA,NA,NA,'NULL',NA,NA,NA))

head(review\_withoutext,10)

write.csv(review\_withoutext,'/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/review\_withouttext.csv')

################## remove text

review = read.csv('/Users/junkangzhang/Downloads/yelp\_review.csv',nrows=100)

user = read.csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp\_sample/sample\_yelp\_user.csv')

################## clean review

review\_withouttext = read.csv('/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/review\_withouttext.csv')

review\_withouttext$date = format(as.Date(review\_withouttext$date, format="%Y-%m-%d"),"%Y")

sample = head(review\_withouttext,100)

review\_withouttext = review\_withouttext %>% group\_by(business\_id,date) %>% summarise(review\_count = n(), mean\_star = mean(stars))

review\_cleaned = review\_withouttext

write.csv(review\_cleaned,'/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/review\_cleaned.csv')

#################### clean checkins

#################### clean business\_attr

business\_attr\_picked = subset(business\_attr, select=c(business\_id,BusinessParking\_garage,BusinessParking\_street,BusinessParking\_validated,BusinessParking\_lot,BusinessParking\_valet))

business\_attr\_picked = business\_attr\_picked %>% mutate(all\_na = ((BusinessParking\_validated=='Na')&(BusinessParking\_validated=='Na')&(BusinessParking\_garage=='Na')&(BusinessParking\_lot=='Na')&(BusinessParking\_street=='Na')&(BusinessParking\_valet=='Na')))

business\_attr\_picked1 = business\_attr\_picked %>% filter(all\_na == FALSE)

attach(business\_attr\_picked1)

#business\_attr\_picked[business\_attr\_picked == 'Na'] <- NA

business\_attr\_picked2 = business\_attr\_picked1 %>% mutate(is\_parking = ((BusinessParking\_validated=='True')|(BusinessParking\_garage=='True')|(BusinessParking\_lot=='True')|(BusinessParking\_street=='True')|(BusinessParking\_valet=='True')))

write.csv(business\_attr\_picked2,'/Users/junkangzhang/Downloads/MMA 2019/Courses/INSY662/group project/yelp/business\_attr\_picked2.csv')

################### join business\_