**Featuretools Exercise 1**

**Source Code**

import pandas as pd

import numpy as np

import featuretools as ft

#read data from csv

clients=pd.read\_csv('clients.csv')

loans=pd.read\_csv('loans.csv')

payments=pd.read\_csv('payments.csv')

#make entityset

es=ft.EntitySet(id='clients')

#add data to entityset

es=es.add\_dataframe(dataframe\_name='clients',index='client\_id',time\_index='joined',dataframe=clients)

es=es.add\_dataframe(dataframe\_name='loans', dataframe=loans, index='loan\_id')

es=es.add\_dataframe(dataframe\_name='payments', dataframe=payments, make\_index=True, index='payment\_id', time\_index='payment\_date')

print(es['payments'])

#aggregation

stats=loans.groupby('client\_id')['loan\_amount'].agg(['sum'])

stats.columns=['total\_loan\_amount']

stats=clients.merge(stats,left\_on='client\_id', right\_index=True,how='left')

print(stats.head(10))

#relationship

r\_client\_previous = ft.Relationship(entityset=es, parent\_dataframe\_name="clients",parent\_column\_name="client\_id",child\_dataframe\_name="loans",child\_column\_name="client\_id")

es=es.add\_relationship(relationship=r\_client\_previous)

r\_payments=ft.Relationship(entityset=es, parent\_dataframe\_name="loans",parent\_column\_name="loan\_id",child\_dataframe\_name="payments",child\_column\_name="loan\_id")

es=es.add\_relationship(relationship=r\_payments)

print(es)

#dfs

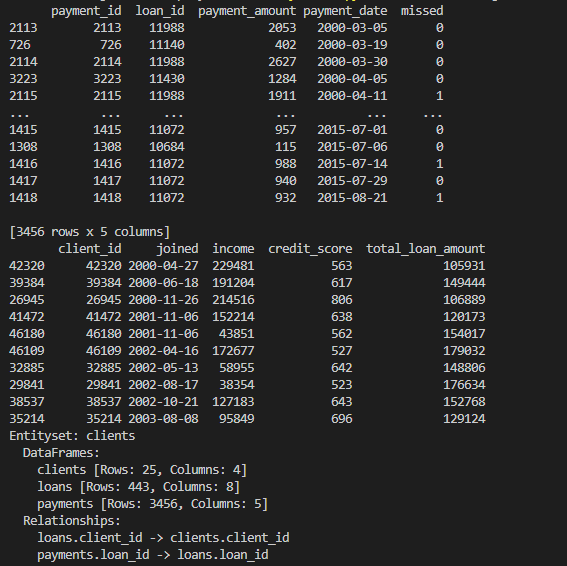
featuref, feature\_names = ft.dfs(entityset=es, target\_dataframe\_name="clients",agg\_primitives=["mean"], trans\_primitives=["month"])

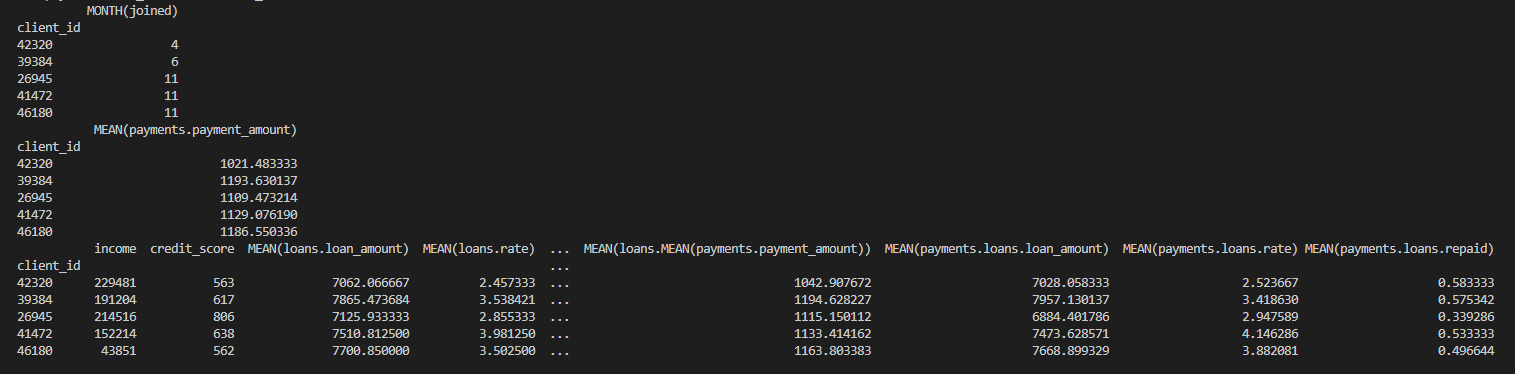
print(pd.DataFrame(featuref['MONTH(joined)'].head()))

print(pd.DataFrame(featuref['MEAN(payments.payment\_amount)'].head()))

print(featuref.head())

**Output Screen Capture**





**Feature Selection Exercise**

**Source Code**

import pandas as pd

import numpy as np

from sklearn.feature\_selection import SelectKBest

from sklearn.feature\_selection import chi2

#read data

dataf = pd.read\_csv("housing.csv")

dataf = dataf[['housing\_median\_age','total\_rooms','total\_bedrooms','population','households','median\_income','median\_house\_value','ocean\_proximity']]

#one-hot encoded

dataf = pd.get\_dummies(data=dataf, columns=["ocean\_proximity"], prefix="ocean")

#fill missing data

dataf.fillna(method='ffill', inplace=True)

#set independent, target

y=dataf['median\_house\_value']

X=dataf[dataf.columns.difference(['median\_house\_value'])]

#selectkbest

bestfeatures = SelectKBest(score\_func=chi2, k=5)

fit=bestfeatures.fit(X,y)

dfcolumns=pd.DataFrame(X.columns)

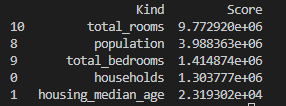
dfscores=pd.DataFrame(fit.scores\_)

featureScores=pd.concat([dfcolumns, dfscores],axis=1)

featureScores.columns=['Kind','Score']

print(featureScores.nlargest(5,'Score'))

**Output Screen Capture**



**Source Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.ensemble import ExtraTreesClassifier

from sklearn.model\_selection import train\_test\_split

#read data

dataf = pd.read\_csv("housing.csv")

dataf = dataf[['housing\_median\_age','total\_rooms','total\_bedrooms','population','households','median\_income','median\_house\_value','ocean\_proximity']]

#one-hot encoded

dataf = pd.get\_dummies(data=dataf, columns=["ocean\_proximity"], prefix="ocean")

#fill missing data

dataf.fillna(method='ffill', inplace=True)

#set independent, target

y=dataf['median\_house\_value']

X=dataf[dataf.columns.difference(['median\_house\_value'])]

#Divide trainset and testset and use trainset because of momory error

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,train\_size=0.2,test\_size=0.8)

#extratreesclassifier

model=ExtraTreesClassifier()

model.fit(X\_train,y\_train)

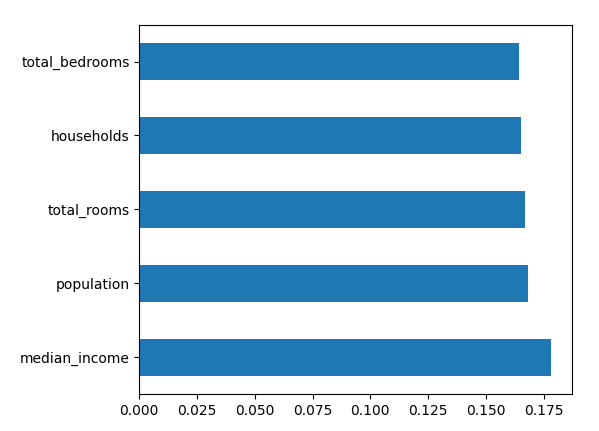
print(model.feature\_importances\_)

feat\_importances = pd.Series(model.feature\_importances\_, index=X.columns)

feat\_importances.nlargest(5).plot(kind='barh')

plt.show()

**Output Screen Capture**



**Source Code**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

#read data

dataf = pd.read\_csv("housing.csv")

dataf = dataf[['housing\_median\_age','total\_rooms','total\_bedrooms','population','households','median\_income','median\_house\_value','ocean\_proximity']]

#one-hot encoded

dataf = pd.get\_dummies(data=dataf, columns=["ocean\_proximity"], prefix="ocean")

#fill missing data

dataf.fillna(method='ffill', inplace=True)

#set independent, target

y=dataf['median\_house\_value']

X=dataf[dataf.columns.difference(['median\_house\_value'])]

#make heatmap

corrmat = dataf.corr()

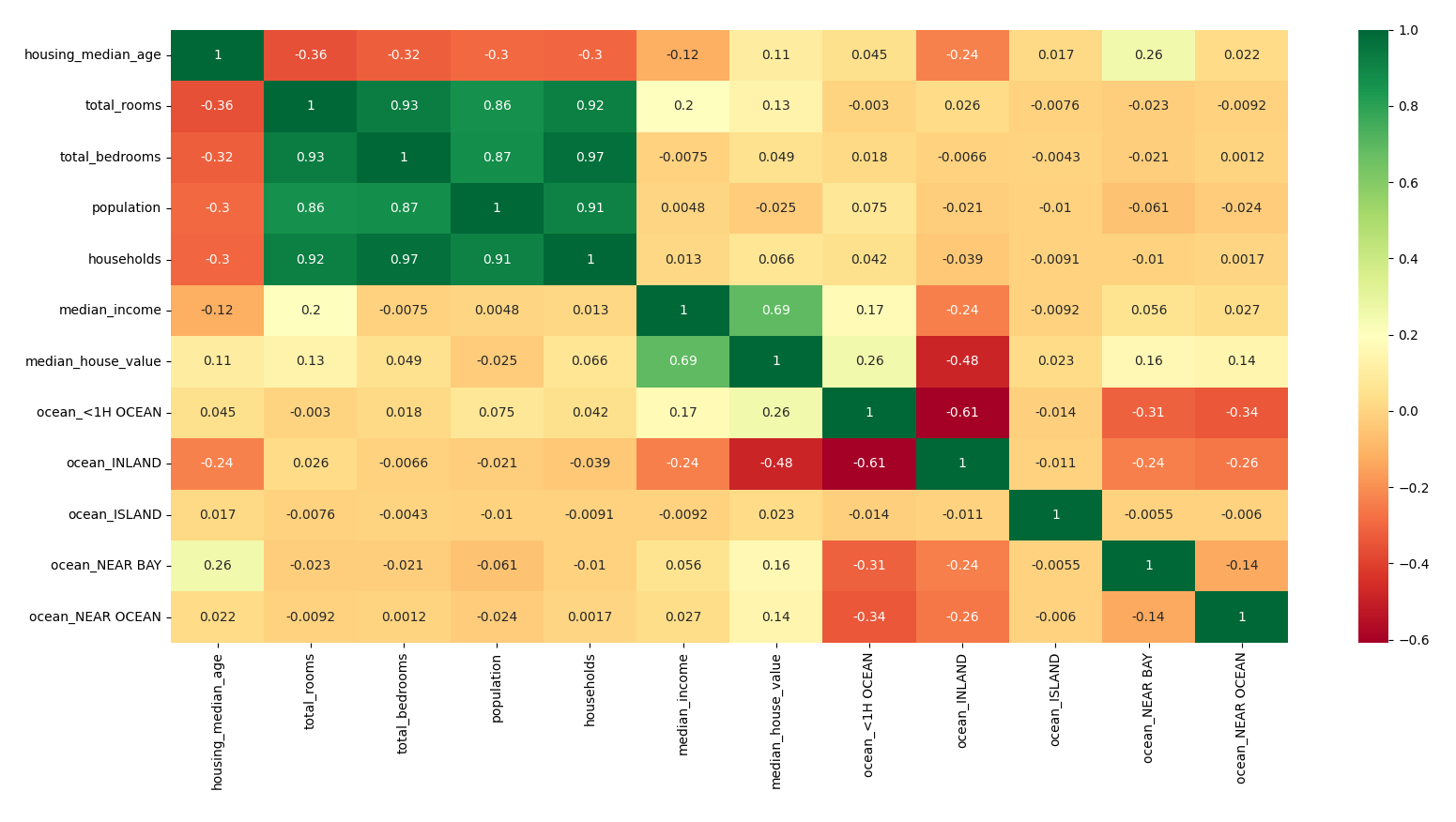
top\_corr\_features=corrmat.index

plt.figure(figsize=(20,20))

g=sns.heatmap(dataf[top\_corr\_features].corr(),annot=True,cmap="RdYlGn")

plt.show()

**Output Screen Capture**



**PCA Exercise**

**Source Code**

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

#read data

dataf=pd.read\_csv("housing.csv")

dataf=dataf[['housing\_median\_age','total\_rooms','total\_bedrooms','population','households','median\_income','median\_house\_value']]

#fill missing data

dataf.fillna(method='ffill', inplace=True)

scaler=StandardScaler()

#set independent, target

y=dataf['median\_house\_value']

X=dataf[dataf.columns.difference(['median\_house\_value'])]

X=scaler.fit\_transform(X)

features=['housing\_median\_age','total\_rooms','total\_bedrooms','population','households','median\_income']

#use 3 components [0.64603148 0.17879772 0.13526617], 2 components [0.64603148 0.17879772] so choose 3 components

pca=PCA(n\_components=3)

printcipalcomponents=pca.fit\_transform(X)

principaldf=pd.DataFrame(data=printcipalcomponents,columns=['principal component 1', 'principal component 2', 'principal component 3'])

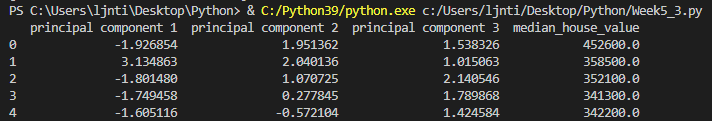
#match principal data to target data

finaldf=pd.concat([principaldf,dataf[['median\_house\_value']]],axis=1)

#print

print(finaldf.head())

**Output Screen Capture**



**Written Homework 1 – Internet search about predict the result of an election**

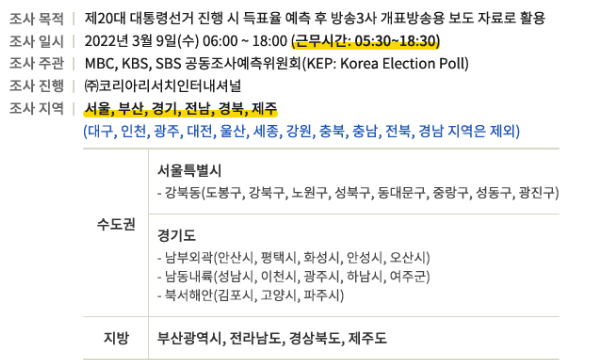
20th president election sampling is done by KBS, SBS, MBC with Korea Research

About 1670 people of researcher are hired to election sampling

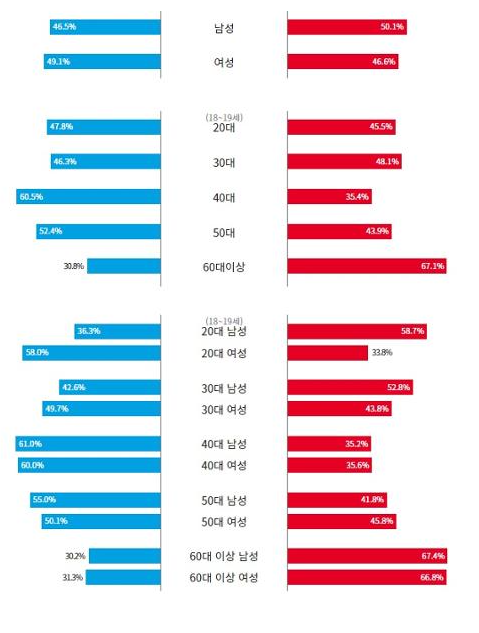
These people stand-by 50m far from polling place and interview people

It is planned to select 85000 people of 44197692 and real reply people was 73297

It is done at several regions(서울, 경기, 부산, 전남, 경북, 제주)



It was analyzed by age, gender and both, etc.



It ask several question

* Which candidate do you select?
* Why do you select that candidate?
* What is most important issue of government?
* Do you satisfied with candidates?
* Etc

It predicted almost same with real ratio of approval votes

**Reference for Written Homework 1**

중앙일보 – 출구조사 응답자 49% ~ - [https://www.joongang.co.kr/article/25054257#home](https://www.joongang.co.kr/article/25054257)

코리아리서치 - 20대 대통령 출구조사원 모집 - <https://poll.kric.com/main/Info.asp>

한겨레21 – 대선 출구조사 결과는 왜 정확했나

<https://h21.hani.co.kr/arti/politics/politics_general/51706.html>

한국일보 - <https://www.hankookilbo.com/News/Read/A2022030919580005949>

**Written Homework 2 – Stratified random sampling in market research for food-delivery**

I want to do ‘hand-made hamburger delivery service’ in Pangyo

Divide group by ‘apartment complex’ (within half and hour to go with motorcycle)

Employ researcher for each ‘apartment complex’

Select samples randomly from each stratum and do survey or interview. (phone call or home visit)

Survey or interview question

* How many time do you use food-delivery service at month?
* When(time) do you use food-delivery service?
* What kind of food do you prefer to use delivery service?
* Is there any new food you want to use delivery service?
* What kind of hamburger do you like? Hand-made or fast-food?
* If you eat hamburger combo, what kind of side dishes do you want?

Survey or interview is doing in 5pm ~ 8pm (After office hours)

If not reply for survey or interview, then select other sample

**Reference for Written Homework 2**

Investopedia – How Stratified Random Sampling Works

<https://www.investopedia.com/ask/answers/032615/what-are-some-examples-stratified-random-sampling.asp>

Naver Post – 시장 조사를 잘 하는 습관

<https://post.naver.com/viewer/postView.nhn?volumeNo=26697641&memberNo=2924457>