Lab Assignment 1
CIS 660 Data Mining
Adventure Works Cycles

Data Processing Language : Python 3.5 IDE

Part 1: Feature Selection, Cleaning, and Preprocessing to Construct an Input from Data Source

-Purchase a bike is not effected by name of customer. So, droping that columns.

-Also many states and cities to work on. So, droping all address related data and only using country for getting results

CustomerKey	Discrete
GeographyKey	Discrete
CustomerAlternateKey	Nominal
Gender	Nominal
MaritalStatus	Nominal
EnglishEducation	Nominal
SpanishEducation	Nominal
FrenchEducation	Nominal
EnglishOccupation	Nominal
SpanishOccupation	Nominal
FrenchOccupation	Nominal
HouseOwnerFlag	Discrete
DateFirstPurchase	Nominal
CommuteDistance	Nominal
Region	Nominal
Age	Discrete
BikeBuyer	Discrete
NumberCarsOwned	Discrete
NumberChildrenAtHome	Discrete
TotalChildren	Discrete
YearlyIncome	Discrete

Part 2: Data Preprocessing and Transformation

One Hot Encoding for nominal data
1)All Education
2) All occupation

Part 3: Calculating Proximity of Two Binary Object Vectors With Simple Matching , Jaccard Similarity, Cosine Similarity

EnglishEducation	Nominal
EnglishOccupation	Nominal
HouseOwnerFlag	Discrete
DateFirstPurchase	Nominal
CommuteDistance	Nominal
Region	Nominal
Age	Discrete
BikeBuyer	Discrete
NumberCarsOwned	Discrete
NumberChildrenAtHome	Discrete
TotalChildren	Discrete
YearlyIncome	Discrete

transforming the values in Normalizer form

applied
Jaccard Similarity . Cosine Similarity and pearsonr

for occupation and YearlyIncome ----------(1)
and also for Education and YearlyIncome --------(2)

which gave the values
Cosine Similarity
(1):0.42748354460006355
(2):0.5826049619920608

Jaccard Similarity
(1):1.0
(2):1.0

pearsonr
(1):0.4845560071281075
(2):0.12432250951401452

```
(18484, 32)
CustomerKey              int64
GeographyKey             int64
Gender                  object
MaritalStatus           object
EnglishEducation        object
EnglishOccupation       object
HouseOwnerFlag           int64
Age                      int64
BikeBuyer                int64
NumberCarsOwned          int64
NumberChildrenAtHome     int64
TotalChildren            int64
YearlyIncome             int64
dtype: object
Cosine Similarity btw Management and YearlyIncome:  0.42748354460006355
Cosine Similarity btw Graduate Degree and YearlyIncome:  0.5826049619920608
jaccard Similarity btw Management and YearlyIncome:  1.0
jaccard Similarity btw Graduate Degree and YearlyIncome:  1.0
pearsonr stats btw Management and YearlyIncome:  0.4845560071281075
pearsonr stats btw Graduate Degree and YearlyIncome:  0.12432250951401452
Cosine Similarity btw 11000 and 11001:  5.236264200014773e-09
Cosine Similarity btw 11000 and 11002:  0.0003233364996653165
>>>
```

```
EnglishEducation_df=pd.get_dummies(df['EnglishEducation'],drop_first=True)
df.drop(columns=['EnglishEducation'],axis=1,inplace=True)
df=pd.concat([EnglishEducation_df,df],axis=1)


EnglishOccupation_df=pd.get_dummies(df['EnglishOccupation'],drop_first=True)
df.drop(columns=['EnglishOccupation'],axis=1,inplace=True)
df=pd.concat([EnglishOccupation_df,df],axis=1)
#df= df.join(EnglishEducation_df)
#df= df.join(EnglishOccupation_df)


from sklearn.preprocessing import Normalizer
norm=Normalizer().fit(df)
normalized_df=norm.transform(df)

#print(EnglishEducation_df)

from scipy.spatial import distance

print("Cosine Similarity btw Management and YearlyIncome: ",distance.cosine(df['Management'].values,df['Ye

print("Cosine Similarity btw Graduate Degree and YearlyIncome: ",distance.cosine(df['Graduate Degree'].va


print ("jaccard Similarity btw Management and YearlyIncome: ",distance.jaccard(df['Management'].values,df
print ("jaccard Similarity btw Graduate Degree and YearlyIncome: ",distance.jaccard(df['Graduate Degree']

from scipy.stats import pearsonr
print ("pearsonr stats btw Management and YearlyIncome: ",pearsonr(df['Management'].values,df['YearlyIncor
print ("pearsonr stats btw Graduate Degree and YearlyIncome: ",pearsonr(df['Graduate Degree'].values,df['



print("Cosine Similarity btw 11000 and 11001: ",distance.cosine(normalized_df[1],normalized_df[2]))

print("Cosine Similarity btw 11000 and 11002: ",distance.cosine(normalized_df[1],normalized_df[3]))
```