Assignment-2

# Parth Vohra 102016044 3CS10

**UML501**

**Part I:** Based on Feature Selection, Cleaning, and Preprocessing to Construct an Input from Data

Source

(a) Examine the values of each attribute and select a set of attributes only that would affect to predict

future bike buyers to create your input for data mining algorithms. Remove all the unnecessary

attributes. (Select features just by analysis).

(b) Create a new Data Frame with the selected attributes only.

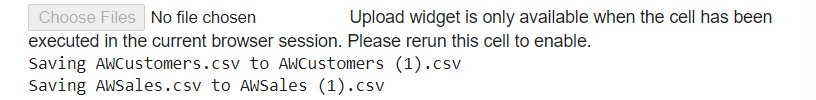
(c) Determine a Data value type (Discrete, or Continuous, then Nominal, Ordinal, Interval, Ratio) of

each attribute in your selection to identify preprocessing tasks to create input for your data mining.

**Code:**

**from google.colab import files**

**uploaded = files.upload()**



**Code:**

**from sklearn import preprocessing**

**import numpy as np**

**import pandas as pd**

**from sklearn.impute import SimpleImputer**

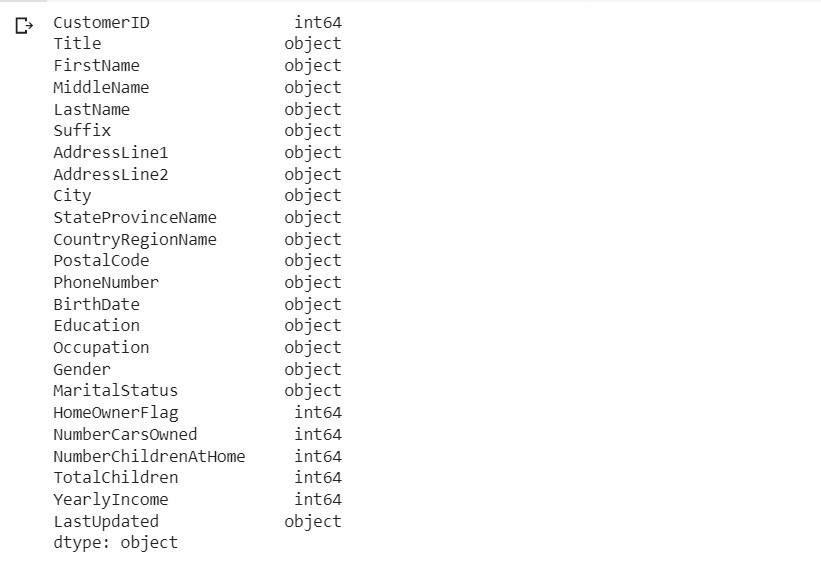
**from sklearn.model\_selection import train\_test\_split**

**from sklearn.feature\_selection import VarianceThreshold**

**data=pd.read\_csv('AWCustomers.csv')**

**print(data)**

**print(data.dtypes)**



**Code:**

**dat=data.copy()**

**dat.head()**



**Part II:** Data Preprocessing and Transformation

Depending on the data type of each attribute, transform each object from your pre-processed data.

Use all the data rows (~= 18000 rows) with the selected features as input to apply all the tasks below, do

not perform each task on the smaller data set that you got from your random sampling result.

(a) Handling Null values

(b) Normalization

(c) Discretization (Binning) on Continuous attributes or Categorical Attributes with too many different

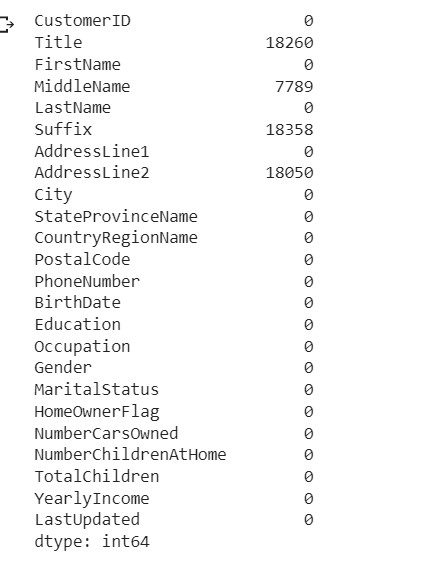
values

(d) Standardization/Normalization

(e) Binarization (One Hot Encoding)

**Code:**

**dat.isnull().sum()**



**Code:**

**dat.shape**



**dat.Title.fillna("K",inplace=True)**

**dat.head()**



**Code:**

**dat.drop("Suffix",axis=1,inplace=True)**

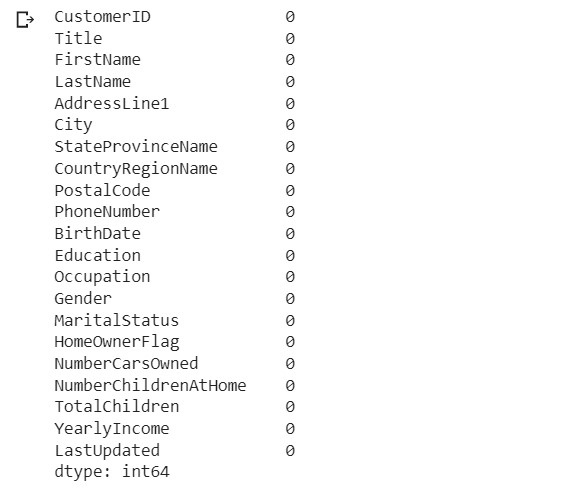
**dat.isna().sum()**

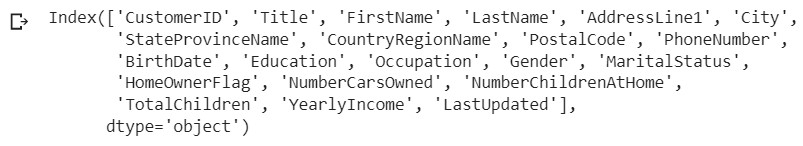
**#dat.drop("MiddleName",axis=1,inplace=True)**

**dat.drop("AddressLine2",axis=1,inplace=True)**

**dat.isna().sum()**

**dat.columns**





**Code:**

**da=dat.copy()**



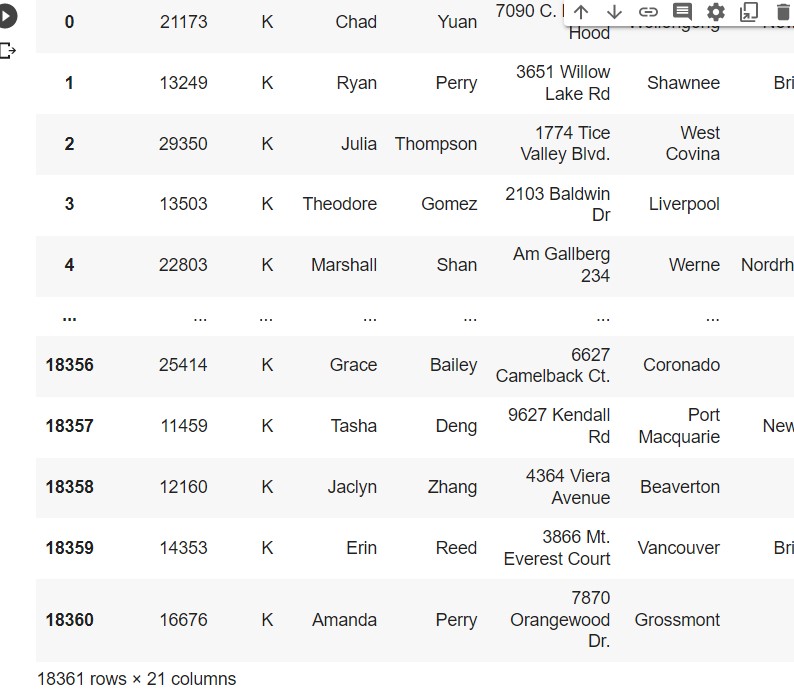
**Code:**

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

dat['YearlyIncome']=scaler.fit\_transform(dat['YearlyIncome'].values.reshape(-1, 1)).flatten()

dat

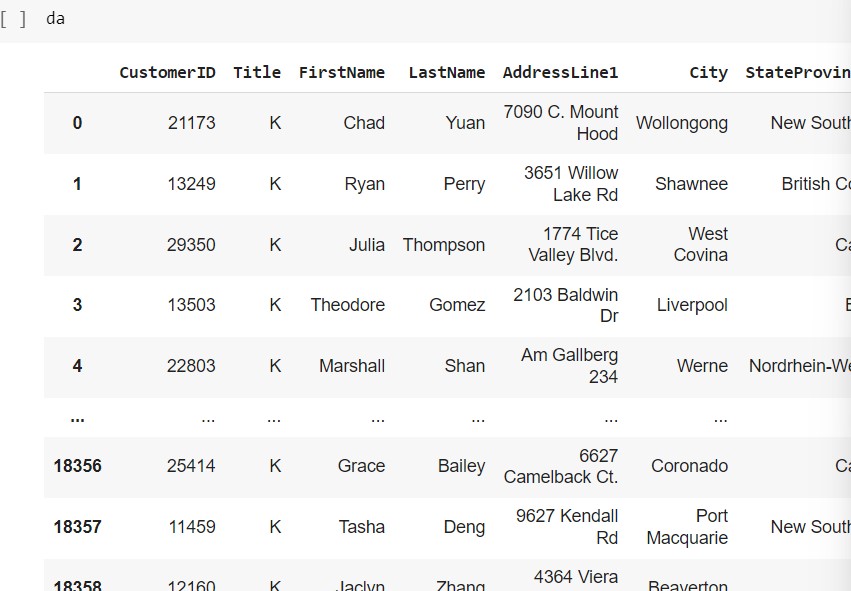


**Code:**

from sklearn.preprocessing import OneHotEncoder

onehotencoder = OneHotEncoder()

onehotencoder.fit\_transform(da)



**Part III:** Calculating Proximity /Correlation Analysis of two features

Make sure each attribute is transformed in a same scale for numeric attributes and Binarization for each

nominal attribute, and each discretized numeric attribute to standardization. Make sure to apply a correct

similarity measure for nominal (one hot encoding)/binary attributes and numeric attributes respectively.

(a) Calculate Similarity in Simple Matching, Jaccard Similarity, and Cosine Similarity between two

following objects of your transformed input data.

(b) Calculate Correlation between two features Commute Distance and Yearly Income

**Code:**

def cosine\_similarity(a, b):

if len(a) != len(b):

return None

a=da['NumberCarsOwned']

b=da['NumberChildrenAtHome']

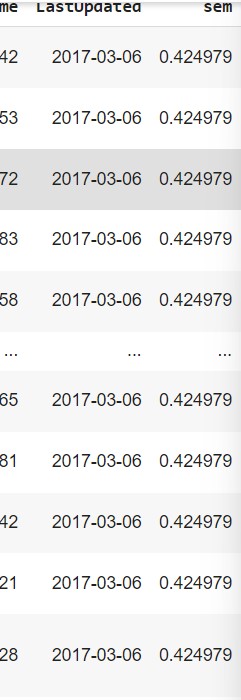
dot\_product = np.dot(a, b)

l2\_norm\_a = np.sqrt(np.sum(a\*\*2))

l2\_norm\_b = np.sqrt(np.sum(b\*\*2))

cosine\_similarity = dot\_product / (l2\_norm\_a \* l2\_norm\_b)

da["sem"]=cosine\_similarity



**Code:**

print(da.corr(method="pearson")

