**Part 1**

Follow the instructions from the video for this part.

1. Download the US Census Income Dataset:
   * <https://archive.ics.uci.edu/ml/datasets/census+income>

Graphical user interface, text, application, email

Description automatically generated

1. Write a python script that:
   1. Removes all rows that contain missing values. These are represented as **' ?'** (Notice the white space preceding the question mark).
   2. Transforms the categorical & ordinal attributes based on the discussed materials.

import pandas as pd  
import numpy as np   
  
  
df\_census = pd.read\_csv("adult.data" ,header =None);  
df\_census

A screenshot of a computer

Description automatically generated  
  
  
Removes all rows that contain missing values. These are represented as **' ?'** (Notice the white space preceding the question mark)

df\_census = df\_census[~ df\_census.isin([' ?']).any(axis=1)]  
df\_census

Graphical user interface, text, application, email

Description automatically generated

Transforms the categorical & ordinal attributes based on the discussed materials.

#Ratio 0 ,2,4,10,11,12  
#Categorical : 1,5,6,7,8,9,13 : get\_dumies  
#Ordinal : 3 ,14: int which maintain order

#Ratio 0,2,4,10,11,12  
  
new\_df = df\_census [[0,2,4,10,11,12]]  
new\_df

A screenshot of a computer

Description automatically generated

#Categorical   
#lets work on df\_census[1]  
  
  
df\_census[1].value\_counts()

Private 22286  
 Self-emp-not-inc 2499  
 Local-gov 2067  
 State-gov 1279  
 Self-emp-inc 1074  
 Federal-gov 943  
 Without-pay 14  
Name: 1, dtype: int64

# want to Do same for multiple Colums so   
  
col =[1,5,6,7,8,9,13];  
for c in col :  
 new\_df = pd.concat( [new\_df , pd.get\_dummies(df\_census[c])] ,axis = 1)  
new\_df

A picture containing graphical user interface

Description automatically generated

#Ratio 3 , 14  
  
df\_census[3].unique()

edu = {  
   
   
 ' Doctorate':14,  
 ' Prof-school':13,  
 ' Masters':12,  
 ' Bachelors':11,  
 ' Assoc-acdm':10,  
 ' Assoc-voc':10,  
 ' Some-college':9,  
 ' HS-grad':8,  
 ' 12th':7,  
 ' 11th':6,  
 ' 10th':5,  
 ' 9th':4,  
 ' 7th-8th':3,   
   
 ' 5th-6th':2,  
 ' 1st-4th':1,  
 ' Preschool':0  
   
   
   
}

df\_census[3].map(edu)

0 11  
1 11  
2 8  
3 6  
4 11  
 ..  
32556 10  
32557 8  
32558 8  
32559 8  
32560 8  
Name: 3, Length: 30162, dtype: int64

Graphical user interface, text, application, email

Description automatically generated

new\_df[3]=df\_census[3].map(edu)  
new\_df

A screenshot of a computer

Description automatically generated

df\_census[14].unique()

array([' <=50K', ' >50K'], dtype=object)

dict = {' <=50K':0, ' >50K':1}

new\_df[14] = df\_census[14].map(dict)

new\_df

A screenshot of a computer

Description automatically generated

**Part 2**

Write a python script that cleans and transforms the Google Play Store Dataset.

Dataset file can be downloaded here:

* [https://www.kaggle.com/lava18/google-play-store-apps (Links to an external site.)](https://www.kaggle.com/lava18/google-play-store-apps)
* Use the file googleplaystore.csv

**Graphical user interface, text, application, email

Description automatically generated**

import pandas as pd  
import numpy as np   
  
  
df\_gps = pd.read\_csv("googleplaystore.csv")  
df\_gps

Graphical user interface, text, application

Description automatically generated

df\_gps =df\_gps.dropna()  
df\_gps

Graphical user interface, text, application

Description automatically generated

new\_df = df\_gps[['Rating' ,'Reviews','Price' ]]  
new\_df

Rating Reviews Price  
0 4.1 159 0  
1 3.9 967 0  
2 4.7 87510 0  
3 4.5 215644 0  
4 4.3 967 0  
... ... ... ...  
10834 4.0 7 0  
10836 4.5 38 0  
10837 5.0 4 0  
10839 4.5 114 0  
10840 4.5 398307 0  
  
[9360 rows x 3 columns]

A screenshot of a computer

Description automatically generated

df\_gps.replace('Varies with device' ,'0',inplace=True)  
df\_gps

Graphical user interface, text, application

Description automatically generated

df\_gps.isin(['Varies with device']).sum()

App 0  
Category 0  
Rating 0  
Reviews 0  
Size 0  
Installs 0  
Type 0  
Price 0  
Content Rating 0  
Genres 0  
Last Updated 0  
Current Ver 0  
Android Ver 0  
dtype: int64

Graphical user interface, text, application

Description automatically generated

s = df\_gps['Size'].str.replace('M' ,'')  
df\_gps = df\_gps.drop(['Size'],axis=1)  
df\_gps['Size'] =s  
  
  
  
i = df\_gps['Installs'].str.replace('+','')  
df\_gps =df\_gps.drop(['Installs'],axis=1)  
df\_gps['Installs'] =i  
  
df\_gps

A screenshot of a computer

Description automatically generated

new\_df = pd.concat([new\_df , df\_gps['Installs']] ,axis=1)  
new\_df = pd.concat([new\_df , df\_gps['Size']] ,axis=1)  
  
new\_df

Graphical user interface

Description automatically generated

list = ['Category', 'Type' ,'Content Rating' ,'Genres' ,'Current Ver']  
for l in list:  
 new\_df = pd.concat( [new\_df , pd.get\_dummies(df\_gps[l])] ,axis = 1)  
   
new\_df

Graphical user interface, application

Description automatically generated

import datetime as dt   
  
  
pd.to\_datetime( df\_gps['Last Updated']).min()  
df\_gps['Last Updated'] = pd.to\_datetime(df\_gps['Last Updated']) - dt.datetime(2010,5,21)  
new\_df = pd.concat([new\_df , df\_gps['Last Updated']] ,axis=1)  
new\_df

A screenshot of a computer

Description automatically generated

version ={  
   
 '8.0 and up':29,  
 '7.1 and up':28,  
 '7.0 - 7.1.1':27,  
 '7.0 and up':26,  
 '6.0 and up':25,  
 '5.1 and up':24,  
 '5.0 - 6.0':23,  
 '5.0 and up':22,  
 '5.0 - 8.0':21,   
 '4.4W and up':20,  
 '4.4 and up':19,  
 '4.3 and up':18,  
 '4.2 and up':17,  
 '4.1 and up':16,   
 '4.1 - 7.1.1':15,   
 '4.0.3 - 7.1.1':15,  
 '4.0.3 and up':14,  
 '4.0 and up':13,  
 '3.2 and up':12,  
 '3.1 and up':11,  
 '3.0 and up':10,   
 '2.3.3 and up':9,  
 '2.3 and up':8,  
 '2.2 and up':7,  
 '2.1 and up':6,  
 '2.0 and up':5,  
 '2.0.1 and up':4,  
 '1.6 and up':3,   
 '1.5 and up':2,  
 '1.0 and up':1,  
 #'Varies with device':0  
   
}

Graphical user interface, text, application

Description automatically generated

df\_gps['Android Ver'].map(version)

0 14.0  
1 14.0  
2 14.0  
3 17.0  
4 19.0  
 ...   
10834 16.0  
10836 16.0  
10837 16.0  
10839 NaN  
10840 NaN  
Name: Android Ver, Length: 9360, dtype: float64

new\_df['Android Ver'] = df\_gps['Android Ver'].map(version)  
new\_df

A screenshot of a computer

Description automatically generated