موسسه آموزش عالی آزاد توسعه

برگزار کننده دوره‌های تخصصی علم داده

Homework 5: Winter 2020

Due date: ۲۳ بهمن

Please email your HWs to y.zerehsaz@gmail.com
Please append all your codes to your response
*****Please hand in your HWs as a word file with your email as the document's name. For instance, I would name my word file as y.zerehsaz@gmail.com.docx.**
*****Make sure to copy and paste the codes that you used for each question. I need to see your plots, results and conclusions but not the long output of your codes.**
*****When asked, please explain your results.**

Consider the "digi.csv" dataset to answer the questions.

Since the dataset contains some Farsi words, we need to read the file by specifying the encoding argument.

df=pd.read_csv('…/digi2.csv',encoding='UTF-8')

To be able to do this HW, you need to take a look at your googleplayapps.py script.

1) Get the columns of the data frame df.

2) Count the number of missing values in the columns of this dataset.
Hint: use df.isna()

3) Remove the column 'Amount_Gross_Order' since it is not well defined.
Hint: use df.drop()

4) Drop the duplicates in the dataset.
Hint: df.drop_duplicates(inplace=True)

5) Change the structure of the variable 'DateTime_CartFinalize' to timestamp. You can use pd.Timestamp function.

This part of the homework is optional:
Note that you can change the "Miladi" date to "Shamsi" using a package called persiantools. So, you can install this package in your Python console or Anaconda Prompt by typing and entering

pip install persiantools

After installing the package, call the appropriate function for this conversion as
from persiantools.jdatetime import JalaliDate

You can use it as JalaliDate(pd.Timestamp('1/17/2020')), and it will return JalaliDate(1398, 10, 27, Jomeh)
You can even get the year, day and month.
JalaliDate(pd.Timestamp('1/6/2020')).year
1398
JalaliDate(pd.Timestamp('1/6/2020')).month
10
JalaliDate(pd.Timestamp('1/6/2020')).day
27
You can change the 'DateTime_CartFinalize' to 'Shamsi' dates using JalaliDate function.

6) When was the first purchase in the data? When was the last one?
Hint: use the 'DateTime_CartFinalize' column

7) The columns 'ID_Item' and 'ID_Customer' show the IDs of the sold items and customers, respectively.
How many unique items have been sold?
How many unique customers have bought items?

8) The column 'city_name_fa' shows the name of the cities. How many unique cities have been used in the dataset?

9) Count the number of transactions in the cities. Sort the results in a descending manner and show the first 20 cities. To do this, you need to, first, groupby your data frame based on the 'city_name_fa' variable. Apply the count() function and choose the 'ID_Order' column. Then, you can use the sort_values(ascending=False) function to sort the data and choose the first twenty cities. (we have similar codes in googleplayapps.py script)

10) Use the same logic to count the number of transactions in each year. To do this, you first need to extract the year from the elements of 'DateTime_CartFinalize' column. You can use

df['DateTime_CartFinalize'].agg(…) to construct a new column called 'Year'. However, to be able to extract the year element from a time stamp, we need to write a function and then aggregate this function on the elements of 'DateTime_CartFinalize' column. So, write a function extracting the year first and then aggregate it. After generating a new column called 'Year', you can groupby the whole dataset based on Year and use the logic in the previous question.

11) We need to count the number of transactions for some cities and in some specific years.
We can groupby our dataset based on both cities and year of purchase.
df.groupby(['city_name_fa','Year_of_purchase']).count()['ID_Order']
This gives you a multi_index dataset, and you can use get function to choose a specific city to acquire some additional information about. Please use this function to get some information about 'تهران' and 'اصفهان'.

12) The column 'Quantity_item' gives you the number of items purchased in each transaction. What are the quantity levels used in this dataset? (by quantity levels I mean numbers and not categorical levels)
Use the logic in Questions 9 and 10 to see what the twenty mostly-used quantity levels are in the dataset.

13) Use the same method in Question 11 to obtain Quantity levels used in different cities separately. Get the information about 'رشت' and 'مشهد'.

14) We can use the same approaches in Questions 11 and 13 to obtain Quantity levels used in different years. (You do not need to do that, but it would be helpful to try)

15) We would like to get some information about our customers. So, we need to, first, group by our dataset based on the column 'ID_Customer'. The first question is that what is the number of times each customer has made a purchase? Use the same method you used in Questions 9 and 10.

16) The 'ID_Item' column gives you the IDs for the items sold in Digikala. We can use this column to see what are the most frequently sold items in the dataset. Please find twenty of the best sellers. Hint: use the 'ID_Item' column and apply the value_counts function to it.

17) This code will give you the top five items sold in each city. (This question is optional by highly recommended)
cities=dff['city_name_fa'].unique()# find the cities
b=dict()#define an empty dictionary. A dictionary is similar to a list only that each element can have a name in a dictionary
Write a for loop to:
1) Get each city as x

2) Call the ID_Item column
3) Count the number of items sold in each of these cities
4) Get only the first five IDs

```
for x in cities:
 b[x]=df.groupby('city_name_fa').get_group(x)['ID_Item'].value_counts().index[:5]
```

‘b’ will be a dictionary with cities as the keys and the top 5 sold items as values corresponding to each city. Now you can use the dictionary ‘b’ to get all the information you want:

```
b['محمود آباد']
Int64Index([66484, 242685, 381617, 1042626, 67262], dtype='int64')
b['تهران']
Int64Index([294942, 36871, 51778, 45121, 8289], dtype='int64')
b['اصفهان']
Int64Index([294942, 45121, 51778, 36871, 8289], dtype='int64')
b['ساری']
Int64Index([294942, 36871, 42124, 165634, 19890], dtype='int64')
b['رشت']
Int64Index([51778, 36871, 294942, 130096, 165336], dtype='int64')
b['اهواز']
Int64Index([129574, 118375, 45121, 42124, 43346], dtype='int64')
```