DOUGLAS COLLEGE – Summer 2023

CSIS 3290 – 001 – Lab 01

- **Create a folder and rename it according to the folder structure and naming convention stated below**
- **All the files you are required to submit for the assignment should be placed inside this folder.**
- **You will lose points if you just cut and paste materials from close exercises (e.g., If I see the same comments, variable names, etc. from class exercises being using in your code).**
- **If cheating is determined (i.e., you shared your work with another student in the class), your work will a ZERO mark and you will face further consequences.**
- **Make sure to include all the necessary files to make sure that the code can run properly without producing any error**

In this lab, we will practice how to prepare and explore an unclean dataset. You need study the demo code and do your own research to make sure that you can perform all the tasks describe below.

1. Create a python notebook named as **Lab1_ABcXXXXX** with A signifies the first letter of your **first name**, Bc signifies the first two letters of your **last name** and XXXXX denotes the last five digits of your **student ID**.

2. Create a markdown cell at the top of the Jupyter notebook to state the lab, **your name and student ID** with the correct heading.

3. For each of the following section, you need to create a **markdown heading cell** followed by a few code cells to complete the tasks. Please also put some comments in each code cell.
   a. **Load the python library**. Please load all the required python libraries in this section
   b. **Read the data**. Please load the provided csv file and have a peek at the data by using the head() function. Then display the column information. Also display the summary of datatypes of your dataset.
   c. Dropping and Filling data
      - Check how many null values you have per column. Drop the column where all values are NaN. You should still have 1970 rows of data.
      - Drop the rows where all data are missing. Remember to use the inplace function or create a new dataframe
      - Notice that you have two columns that are almost similar: **COST** and **COST2**. Find the column that has more null values, that will be the candidate to be dropped:
        - If **COST** has more null than **COST2**, fill the missing values in **COST2** using the values from **COST**. Note that the cost uses different currency. Please make sure to use Canadian dollar with the rate of 1 CAD = 0.67 EUR. Drop column **COST**.
        - You need to do the opposite otherwise.
        - Hint: you should remove the "**CAD$**" and "**EUR**" string first before performing the calculation. Be careful with the NaN value. It needs to be kept as NaN using **numpy.nan**.
      - Please also drop all rows that still has nan/null values.
      - Try to have a glance at the data again using the **head()** function

   d. Changing Columns Label
   You should notice that some of the columns do not have a proper naming. Replace the whitespaces into underscore "_" and change all letters into small case letters. You can use the **str.replace()** to the df.columns or use a lambda function to perform that task.

e. Remove special characters from the data and format some column data

- Remove the % symbols from the **discount** column and make sure to remove the currency symbol and text from any column
- Transform the value of **duration** into minutes. Note that the values in the duration is in hour and minutes. To transform it to minutes, you need to remove the **h** and **m** letter, and then split the value using **split(" ")**, since it has a white space between the two values. Then calculate the duration in minutes, for example if the duration is 9h 45m, you should calculate it as 9*60 + 45 = 585
  Note: the challenge here is that some data points only have minute value. So, you need to check the length of the list from the split() operation. In addition to that, you should make sure that you change the datatype into float to perform the addition and division operation.
- Transform the value of **last_updated** into a number representing the value. You should take the last two digits of the year. For example, 6/2020 should be changed to 20.
- Change the datatypes of column **discount**, **duration** and **last_updated** into float64

f. Reduce the number of unique values in the category

- Use **`unique()`** to check all the unique values of **category** column. Use value_counts() to check the counts of those unique values. Some categories that has very small data will be removed or merged.
- Drop the rows containing the category whose value_counts() is less than 20, i.e, any data points for 'finance accounting' or 'other'
- Combine any category that contains the following word: personal, music, health, lifestyle or photography, into **personal development** category. The following can be used for that purpose. Note that the regex (?i) is used to ignore the letter case
  df.loc[df['category'].str.contains('(?i)personal|music|health|lifestyle|photography'), 'category'] = 'personal development'
- Then, replace any occurrence of the white space into underscore in the category column.
- You should only have eight course category now.
- Find out the number of course **sub_category**. Notice that we still have quite a number of course sub_category.

g. **Drop the unneeded columns and reset the index**

- Drop the unneeded **course_name** and **sub_category** columns and reset the index. Make sure to drop any column with previous index value.
- Display the dataframe info again. Make sure that except for the category column, all columns are using float64 data type.

h. Save the Cleaned Dataset

Save your cleaned dataset as Lab1_ABcXXXXX_cleaned.csv. Make sure to use index=False parameter when saving the file. Note, in order to save the csv file, you should use pd.to_csv() function.

i. Analyze the Statistics

Use the head(), describe() to display some of the data and summary statistics of the data. After that find the covariance and correlation from the dataset. Display the absolute value of correlation for column **avg_rating** in descending order.

Note on submission:

- Create a folder named as Lab1_ABcXXXXX following the naming convention.
- Put your Jupyter notebook and the original and cleaned dataset in this folder.
- Zip the file and submit it through the blackboard

LAB/ASSIGNMENT PRE-SUBMISSION CHECKLIST

- Did you follow the naming convention for your files?!
- Did you follow the naming convention for your folder?!
- Does your submission work on another computer?!
- Double check **before** submitting