DOUGLAS COLLEGE – Summer 2021

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.**

In this lab, we will practice how to prepare and explore an unclean dataset. Most of the steps to perform those actions are provided in this document. However, 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 using **`df.head()`**. After that, check how many null values you have per column using **`df.isnull().sum()`**.
   c. Dropping and Filling data
      - First, we will drop all the rows where all elements are missing. This can be done using **`df.dropna(how='all',inplace=True)`**. Remember that inplace will perform the function in the current df. If you want to keep the original df, you should make a copy of that df. Check again the number of null in the df.
      - From the previous step, you should notice that there are two columns that are almost similar: price and price2. We can see that price2 has more null values than price. So, we are going to fill the null values in price column with the values in price2. This can be done by using **`df.price.fillna(df.price2, inplace=True)`**
      - You should now drop the columns price2. Please also drop all rows that still has nan/null values.
      - After that, you should drop unneeded column such as 'VIN','name','seller_address', and 'id'.
      - 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: 'fuel type', 'exterior color', and 'interior color'. We should remove the whitespace and change it into underscore _. To change the 'fuel type' column label, use **`df.rename(columns={'fuel type': 'fuel_type'}, inplace=True)`**. Please do the same with the other columns that do not have a proper naming.

e. Reduce the number of unique values in the transmission and engine columns

- Use **`df['transmission'].unique()`** to check all the unique values of transmission. We want to limit the number of unique values. We can see that there are several transmission values that resembles 'CVT', 'Auto' and 'Manual'
- Use the following code to change anything resembles cvt (ignoring the letter case) to be changed into 'CVT
  `df.loc[df['transmission'].str.contains('(?i)CVT'), 'transmission'] = 'CVT'`
  Note: str.contains will search the values that resembles a substring we provide. '(?i)CVT' means that we want to find anything resembles CVT by ignoring the letter case (capital or small letter)
- Do the same thing for any values resembling auto and manual.
- Check the unique values in the transmission column again. After that, try to see the number of samples each unique value is represented using **`df.transmission.value_counts()`**. You should see that most of the samples are Auto, Manual and CVT. You should also see some other transmission types. We want to delete any samples whose transmission resembles ' speed', ' shift' and ' not specified'. Notice that there is a white space in front of those transmission type.

```
df.transmission.value_counts()
```

```
Auto                                  645
Manual                                 27
CVT                                    16
 10-Speed                              14
 6-Speed A/T                            8
 1-Speed Continuously Variable Ratio    1
 6-SPD SELECT SHIFT                     1
 Dual Shift Gearbox                     1
 Not Specified                          1
Name: transmission, dtype: int64
```

- We will use str.contains() to find the location of those samples and drop them. For any sample whose transmission resembles speed, the following code can be used.
  `df.drop(df.loc[df.transmission.str.contains('(?i)speed')].index, inplace=True)`
  Perform the same action for any sample whose transmission resembles speed and specified.
- Check the value_counts again, you should see that we will have 645 Auto, 27 Manual and 16 CVT samples.
  `df.transmission.value_counts()`
- Check the engine's unique values using **`df.engine.unique()`**. You should see that there are just too many different engine types. In general, anything that resembles (ignoring case) v-2, v2, I-2, I2, 2 cyl, 2-cyl, 2.X liter should be categorized as V2 engine. Therefore, you can use the following code
  `df.loc[df['engine'].str.contains('(?i)v-2|v2|I2|I-2|2 cyl|2-cyl|2.'), 'engine'] = 'V2'`
  Perform the same operation for V3, V4, V5, V6, and V8
- Check the unique values again. In addition to those V types engine we did in the previous step, we can see that some samples have engine values of '4', '6' or '8'. You need to change those samples such that their engine values should be V4, V6 or V8. Please refer any code above to perform the task.
- Reset the index with the following code **`df.reset_index(inplace=True, drop=True)`**
- Peek the dataset again using head(). The first 7 rows of the data should be like below

	price	miles	fuel_type	exterior_color	interior_color	drivetrain	transmission	engine	model_name
0	$15,999	48,054 miles	Gasoline	Ruby Red Metallic Tinted Clearcoat	Charcoal Black	RWD	Manual	V2	Mustang
1	$31,795	29,050 miles	Gasoline	Glacier White	Gray	RWD	Auto	V6	Model Unknown
2	$13,998	7 miles	Gasoline	Ingot Silver	Charcoal Black	FWD	Auto	V4	Fiesta
3	$19,237	10 miles	Gasoline	Oxford White	Ebony	FWD	Auto	V4	Fusion
4	$38,868	3 miles	Gasoline	Oxford White	Charcoal	Rear Wheel Drive	Auto	V3	Transit-350
5	$12,997	132,434 miles	Gasoline	Silver Birch Metallic	Gray	AWD	Auto	V3	Model Unknown
6	$19,431	12 miles	Gasoline	Diamond White - White	Medium Stone Cloth	Four Wheel Drive	Auto	V2	EcoSport

f. Fix the format of column miles and price

The column miles and price have characters that cannot be used in our modeling later. This include the '$', comma and 'miles' word. We need to replace those with empty string.

- For the miles column, we need to do the following. Notice the whitespace in front of the word miles.
  ```
  df['miles'] = df['miles'].str.replace(',', '')
  df['miles'] = df['miles'].str.replace(' miles', '')
  ```
- For the price column, please remove the comma and the $ sign
- Then, check the dataframe's datatype using df.dtypes command. You can see that all columns are using object (string) datatype.
  Change the datatype of the price column into numeric by **`df["price"] = pd.to_numeric(df["price"])`**.
  Do the same thing for the 'miles' column.

g. Save the Cleaned Dataset

Save your cleaned dataset as Lab01_ABcXXXXX_cleaned.csv. Your cleaned dataset should have 688 rows.

h. 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.

i. Data visualization

Please display the scatter plot relating the price and miles. In addition, please also plot the distribution (using seaborn) of 'price' and 'miles' columns.

Note on submission:

- Create a folder named as Lab01_ABcXXXXX following the naming convention.
- Put your Jupyter notebook and the 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