DOUGLAS COLLEGE – Summer 2023

CSIS 3290 – 001 – Lab 02

- **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 analyze and explore a clean 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 **Lab2_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 csv you have prepared from Lab1 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. Find out the **summary statistics** of the dataset using describe(). However, pass the following parameter to see some more detail information, percentiles=[0.01, 0.25, 0.5, 0.75, 0.99]
   d. Analyze and display some interesting data
      - Display the record(s) where the **number_ratings** is the minimum.
      - Display the record(s) where the **avg_rating** is the minimum.
      - Display the records where the **enrollment** is less than or equal to its 1% percentile.
      - What is the mean of **avg_rating** of the courses whose enrollment is less than or equal to its 1% percentile?

   e. Display the correlation between features, focusing on the **avg_rating** that will be our target for prediction. Considering the correlation values, we will choose several features that have better correlations to the avg_rating
   f. Univariate analysis
      - Display the distribution plot of the **avg_rating**. Notice that it has a long tail on the left. Since avg_rating will be our target for prediction, we want to make sure it is close to a normal distribution. Assuming that avg_rating 3.5 is the minimum cut-off, find the records where the avg_rating is less than 3.5 and drop them. Display the distribution plot of the average rating again.
      - Display the distribution plot of the **inst_rating**. Notice that it has a long tail on the left. Assuming that inst_rating 3.75 is the minimum cut-off, find the records where the inst_rating is less than 3.75 and drop them. Display the distribution plot of the inst_rating again.

- Display the distribution plot of the **enrollment**. Notice that the value of enrollment is quite big as compared to the other features. The distribution plot also has a very long tail on the right. You can either use the cube-root np.cbrt() or np.log1p() transformation to modify the data. Assuming we are using log1p transformation, create a new column in the dataframe named **log_enrollment** using np.log1p() and plot its distribution. Notice that it now has a very good distribution.
- Repeat the above process for the following columns: **number_ratings**, **inst_review**, and **inst_student**. You do not need to drop any records from the newly created log_number_ratings, log_inst_review, and log_inst_student columns.
- **Display the correlation** of the features against the avg_rating. Therefore, delete the following columns from the dataframe: **enrollment**, **number_ratings**, **inst_review**, and **inst_student.**
- Create a countplot for the **category** column

g. **Multivariate analysis**

- Display a multivariate analysis plot for **avg_rating** against **inst_rating** and record your observations as markdown text
- Display a multivariate analysis plot for **avg_rating** against any other features that has high correlation, e.g., log_number_ratings, etc, and record your observations as markdown text

h. **Create dummy_features** for the categorical column.

i. **Reset the index and save the csv file** as Lab02_prepared.csv

j. **Create dataframe for different feature selection methods**

- Looking at the correlation, choose between 6 to 8 features and save it as a **df_correlation**. Make sure to exclude the target, i.e., avg_rating
- Use the variance threshold method to select the best features and save it as **df_variance**
- Use the select K Best method with k=8 to select the best features and save it as **df_selKBest**

Note on submission:

- Create a folder named as Lab2_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