CPS 844 Lab 2: Data Preprocessing

Data preprocessing can significantly improve the quality of the data mining analysis. In this lab, you will write Python code to alleviate some data quality issues, such as missing values and duplicates. You will make use of breast cancer sample data.
Help: use what you learned during class (for example codeCh2.py which is posted on D2L) and during your previous lab. This time only, a supplementary python script with blanks is also made available to help you. However, you don't need to use it if you prefer to do it all by yourself.

Write a Python script that performs the tasks described below. Submit the .py file on D2L. Please note that if you submit your file in some other format besides .py or (.txt should you meet an issue), then your mark will at most be 60%.

To do:

1) Download and check the format of the data:

   https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data

   as well as the description of the data:

   https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names

2) Using one of the functions from the pandas library, load the 'breast-cancer-wisconsin.data' file into a DataFrame. You should get a DataFrame of 699 records and 11 attributes. (0 point)
3) You may have noticed that the data did not have headers. Assign new headers to the DataFrame. The headers should be relevant to the attribute description in the file 'breast-cancer-wisconsin.names'. (0 point)
4) Drop the 'Sample code number' attribute, because it will not bring any useful information in the analysis. (10 points)

Missing Values

5) According to the description of the data, the missing values are encoded as '?' . Replace the '?' to the numpy's constant 'NaN'. To help you with this, check these pages:
   https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.replace.html
   https://numpy.org/doc/stable/reference/constants.html
   (10 points)
6) For each attribute, count and print the number of missing values. You should observe that only the 'Bare Nuclei' attribute contains missing values, 16 in total. (10 points)
7) Discard the data points that contain missing values. (10 points)

Outliers

8) Draw a boxplot to identify the columns in the table that contain outliers. Find that plot. In your python script, add a comment and mention which attributes have outliers. Hint: call the method ‘boxplot’ on your DataFrame. (10 points)

Duplicate Data

9) Check for duplicate instances. Hint: call the method ‘duplicated’ on your DataFrame. In your python script, add a comment and mention how many records were duplicates. (10 points)
10) Drop the row duplicates. Hint: call the method ‘drop_duplicates’ on your DataFrame. (10 points)

Discretization

11) The goal is to transform a continuous-valued attribute to a categorical attribute. Start by plotting a 10-bin histogram of the attribute values 'Clump Thickness' distribution. Hint: call the method ‘hist’ on the attribute 'Clump Thickness' of your DataFrame. (10 points)
12) Discretize the Clump Thickness' attribute into 4 bins of equal width. In your python script, add a comment and mention the range of values of each category, and the number of records that belong to each of the categories. Hint: call the function ‘cut’ of the pandas library. (10 points)

Sampling

13) Randomly select 1% of the data without replacement, and save these samples into a new variable. (10 points)