ECE 445 (Fall 2020) – Notebook Exercise #1 (50 points)

Last updated: September 10, 2020

Rationale and learning expectations: Becoming familiar with different aspects of a dataset and using that familiarity to pre-process the dataset is one of the most important aspects of a machine learning pipeline. It is in this regard that this notebook exercise is meant to reinforce the concepts of dataset exploration and pre-processing for beginners in machine learning. Students attempting this exercise are expected to understand the fundamentals of dataset exploration and pre-processing at the end of it, even if programming aspects of the exercise are not fully mastered.

General Instruction: This exercise makes use of two publicly available datasets. All parts of this exercise must be done within a Notebook, with text answers (and other discussion) provided as markdown / LaTeX cells. Please make sure that the version of the notebook submitted by you has fully executed cells (i.e., submit it after a complete run of all the cells in the notebooks). In addition, submit any files that you are being asked to save as part of the exercise.

Restrictions: You can only use `numpy` and `pandas` packages within your code. Unless explicitly permitted by the instructor, you are not allowed to use any other packages or modules.

Notebook Preamble: I like to import `numpy` and `pandas` as follows (but you are allowed to use any other names of your liking):

```
import numpy as np
import pandas as pd
```

1 Heart Failure Prediction Dataset

Our first dataset is termed *Heart Failure Prediction Dataset*, which can hypothetically be used to determine the likelihood of a death by heart failure event. A machine learning model trained on such a dataset can then potentially be used by hospitals to assess the severity of patients with cardiovascular diseases. You can read further about this dataset at Kaggle using the link provided below. The dataset is stored as a `csv` file, which is also being provided to you as part of this exercise.

- **Dataset link:** `https://www.kaggle.com/andrewmvd/heart-failure-clinical-data`
- **Dataset csv filename:** `heart_failure_dataset.csv`

1.1. Load the dataset from the `csv` file as follows (you are free to choose variable names of your liking):

```
heart_df = pd.read_csv('heart_failure_dataset.csv')
```

Note that the variable `heart_df` is of type `pandas.DataFrame`.

(a) (1 point) Print the `shape`, `axes`, and `dtypes` attributes of `heart_df` dataframe.

(b) (1 point) Print first 10 rows of `heart_df` dataframe using `pandas.DataFrame.head()` function.

(c) (1 point) What is each row of `heart_df` dataframe termed within the machine learning parlance?

(d) (2 points) Based on your knowledge of the dataset and its stated usage, do you think we are dealing with an unsupervised learning problem or a supervised learning problem? Justify your answer.

(e) (1 point) How many independent variables (features, attributes, predictors, etc.) does this dataset have? List down the names of these variables.

(f) (1 point) How many dependent variables (if any) does this dataset have? List down the names of these variables.

(g) (1 point) How many of the variables in this dataset are categorical variables? List down the names of these variables.

 waheed.bajwa@rutgers.edu

(h) (1 point) What type of *encoding* do the categorical variables in the dataset follow?

(i) (1 point) How many samples in the dataset correspond to deceased patients and how many samples correspond to the remaining patients?

(j) (1 point) How many samples in the dataset correspond to women patients and how many samples correspond to male patients?

(k) (1 point) How many samples in the dataset correspond to smokers and how many samples correspond to non-smokers?

1.2. Compute pairwise correlations between variables in the dataset using `pandas.DataFrame.corr()` function.

(a) (1 point) What two variables are the most *positively* correlated with the `DEATH_EVENT` variable?

(b) (1 point) What two variables are the most *negatively* correlated with the `DEATH_EVENT` variable?

(c) (1 point) Based on your knowledge of the dataset, why do you think it makes sense that the second-most positively correlated variable with the `DEATH_EVENT` variable should have been positively correlated?

(d) (2 points) Based on your knowledge of the dataset, why do you think it makes sense that the two most negatively correlated variables with the `DEATH_EVENT` variable should have been negatively correlated?

1.3. (5 points) Write commented code cell to validate that all entries in the dataset are 'valid' and have not been *filled-in* with inconsistent values. If the code finds any invalid values then it should convert them to `NaN` and store the resulting dataframe as a `csv` file with name `heart_failure_dataset_NaNs.csv` Justify your logic by explaining it in a markdown / LaTeX cell.

1.4. (5 points) Write commented code to process the validated and potentially `NaN`-converted dataframe so that each *non-categorical* independent variable in the dataset has empirically zero mean and unit variance. This processing should *ignore* any `NaN` entries in the dataframe. Print first 20 rows of the processed dataframe.

1.5. (3 points) Write commented code to modify the processed dataframe so that the `DEATH_EVENT` variable is encoded using *one-hot encoding*. The code must be written from scratch, i.e., you cannot use a library. Store the final pre-processed dataframe as a `csv` file with name `heart_failure_dataset_processed.csv`

2 Pima Indians Diabetes Dataset

Our second dataset is termed *Pima Indians Diabetes Dataset*, which can hypothetically be used to predict the onset of diabetes based on several diagnostic measures. A machine learning model trained on such a dataset can then potentially be used by physicians to monitor their patients for early signs of diabetes. This dataset is peculiar in the sense that all patients in it are adult females, at least 21 years old, of Pima Indian heritage. You can read further about this dataset at Kaggle using the link provided below. This dataset is also stored as a `csv` file, which is being provided to you as part of this exercise.

- **Dataset link:** `https://www.kaggle.com/uciml/pima-indians-diabetes-database`
- **Dataset csv filename:** `diabetes_dataset.csv`

2.1. Load the dataset from the `csv` file into a pandas dataframe.

(a) (1 point) Print the `shape`, `axes`, and `dtypes` attributes of the dataframe.

(b) (1 point) Print first 10 rows of the dataframe using function.

(c) (2 points) Based on your knowledge of the dataset and its stated usage, do you think we are dealing with an unsupervised learning problem or a supervised learning problem? Justify your answer.

(d) (1 point) How many independent variables (features, attributes, predictors, etc.) does this dataset have? List down the names of these variables.

(e) (1 point) How many dependent variables (if any) does this dataset have? List down the names of these variables.

 waheed.bajwa@rutgers.edu

(f) (1 point) How many of the variables in this dataset are categorical variables? List down the names of these variables.

(g) (1 point) What type of *encoding* do the categorical variables in the dataset follow?

(h) (2 points) How many samples in the dataset correspond to the following age groups:

- Young adults (Ages 21–40)
- Middle-aged adults (Ages 41–60)
- Old-aged adults (Ages 61 and older)

2.2. (3 points) Write commented code cell to validate that all entries in the dataset are 'valid' and have not been *filled-in* with inconsistent values. If the code finds any invalid values then it should convert them to `NaN` and print first 20 rows of the processed dataframe. Justify your logic by explaining it in a markdown / LaTeX cell.

2.3. (2 points) Write commented code to process the validated and potentially `NaN`-converted dataframe so that each *non-categorical* independent variable in the dataset has empirically zero mean and unit variance. This processing should *ignore* any `NaN` entries in the dataframe. Print first 20 rows of the processed dataframe.

2.4. (1 point) Write commented code to modify the processed dataframe so that the `Outcome` variable is encoded using *one-hot encoding*.

2.5. (4 points) Replace the `NaN` values in the dataset for each variable with empirical median of that variable, where only the median corresponding to the same `Outcome` should be used for replacement purposes. Store the final pre-processed dataframe as a `csv` file with name `diabetes_dataset_processed.csv`

 waheed.bajwa@rutgers.edu