ECE 443 (Fall 2022) – Programming Exercise #1 (80 points)

Last updated: September 24, 2022

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 programming 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: All parts of this exercise must be done within a Notebook, with text answers (and other discussion) provided in text cells and commented code provided in code cells. Please refer to the solution template for Exercise #1 as a template for your own solution. In particular:

- Replace any text in the text cells enclosed within square brackets (e.g., `[Your answer to 1.1a goes in this cell]`) with your own text using Markdown and/or LaTeX .
- Replace any text in the code cells enclosed within pairs of three hash symbols (e.g., `### Your code for 1.1a goes in this cell ###`) with your own code.
- Unless expressly permitted in the template, **do not** edit parts of the template that are not within square brackets / three hash symbols and **do not** change the cell structure of the template.
- Make sure the submitted notebook is fully executed (i.e., submit the notebook only after a full run of all cells).

Restrictions: You are free to use `numpy`, `pandas`, `scipy.stats`, and `IPython.display` libraries / packages within your code. Unless explicitly permitted by the instructor, you are not allowed to use any other libraries, packages, or modules within your submission.

Notebook Preamble: I suggest importing the different libraries / packages / modules in the preamble as follows (but you are allowed to use any other names of your liking):

```
import numpy as np
import pandas as pd
from scipy import stats as sps
from IPython.display import display, Latex
```

1 Fetal Health Classification Dataset

Our first dataset is termed *Fetal Health Classification Dataset*, which can hypothetically be used to determine the health of a fetus in the womb using *cardiotocography* (CTG) data. A machine learning model trained on such a dataset can then potentially be used by physicians to prevent child and maternal mortality. You can read further about this dataset using the links provided below. The original dataset has been further organized and structured as a `csv` file within Kaggle, two modified variants of which are being provided to you as part of this exercise.

- **Kaggle dataset link:** `https://bit.ly/fetal-health-classification`
- **Original source link:** `https://bit.ly/uci-cardiotocography`
- **"Clean" dataset** `csv` **filename:** `fetal_health_dataset_clean.csv`
- **"Dirty" dataset** `csv` **filename:** `fetal_health_dataset_dirty.csv`

 waheed.bajwa@rutgers.edu

Clean Dataset

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

```
fetal_df = pd.read_csv('fetal_health_dataset_clean.csv')
```

Note that the variable `fetal_df` is of type `pandas.DataFrame`. Next, carry out basic exploration of data by addressing the following questions.

(a) (2 points) Based on your understanding of the dataset and its potential usage (refer to the Kaggle and source links provided above), is this a supervised machine learning task or an unsupervised machine learning task? Justify your answer as much as possible.

(b) (1 point) Print the `axes` and `dtypes` attributes of `fetal_df` dataframe.

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

(d) (1 point) Using the `shape` attribute of `fetal_df` dataframe, print the number of rows and columns within the `csv` file.

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

(f) (1 point) What is the total number of samples in this fetal health classification dataset?

(g) (2 points) What is the number of independent variables (features, attributes, predictors, etc.) in this dataset? List down the names of these variables.

(h) (2 points) What is the number of dependent variables (if any) in this dataset? List down the names of these variables.

(i) (1 point) Suppose all data in `fetal_df` dataframe corresponding to independent variables is extracted into a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. What would be the values of n and p in this case?

(j) (1 point) Suppose all data in `fetal_df` dataframe corresponding to dependent variables is extracted into a matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$. What would be the values of n and m in this case?

(k) (2 points) Based on your understanding of the different variables in the dataset (refer to the Kaggle and source links provided above), do you think the variables in the dataset are '*raw*' in the sense that they are effectively what the acquisition system output or do you think some (or all) of them have been pre-processed into transformed features? Justify your answer as much as possible.

(l) (3 points) Based on your understanding of the different variables in the dataset (refer to the Kaggle and source links provided above) as well as potential usage of functions such as `pandas.unique()` and `pandas.DataFrame.hist()`, determine the number of categorical variables in the dataset. List down the names of these variables and also justify your answer as much as possible.

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

(n) (3 points) Write code to determine how many samples in the dataset correspond to fetuses with the following health condition:

- Normal
- Suspect
- Pathological

Dirty Dataset

Load the ditry dataset from the `csv` file into a pandas dataframe as follows (you are free to choose variable names of your liking):

```
fetal_dirty_df = pd.read_csv('fetal_health_dataset_dirty.csv')
```

 waheed.bajwa@rutgers.edu

Note that unless additional arguments are passed to `pandas.read_csv()`, any empty (missing) values in the `csv` file are encoded as `NaN` in the pandas dataframe by default. Using this newly created dataframe, answer the following questions concerning the dirty version of the dataset.

1.2. Explore whether there are missing values in the dirty dataset and, if there are, determine the extent of them in the following manner.

(a) (1 point) Write code to determine whether there are any missing values in the dirty dataset by checking for the presence of `NaN` values in `fetal_dirty_df` dataframe using the `pandas.DataFrame.isna()` function.

(b) (2 points) Write code to determine the variables (if any) in the dataset that have missing values. List down the names of all these variables.

(c) (2 points) Write code to determine and print the number of samples (if any) in the dataset that have missing values.

1.3. (6 points) In addition to missing values, validation of the *logical consistency* of data samples (e.g., body temperature of a patient being above 120 degrees) is another important component of data pre-processing. Based on your understanding of the different variables in the dataset (refer to the Kaggle and source links provided above), write code to check the logical consistency of values in `fetal_dirty_df` dataframe and, if any inconsistent/invalid values are found, convert the invalid values to `NaN`. Next, print rows 91 to 100 of the processed dataframe using `pandas.DataFrame.loc()` function. Justify your logic by carefully explaining it in a text cell.

1.4. (4 points) Write 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. (6 points) Write code to replace the `NaN` values for each variable in the processed dataframe with empirical median of that variable, where only the median corresponding to the same `fetal_health` category should be used for replacement purposes.

1.6. (3 points) Write code to modify the processed dataframe so that the `fetal_health` variable is encoded using *one-hot encoding*. Save the final pre-processed dataframe as a `csv` file using `pandas.DataFrame.to_csv()` function. The file should be named `fetal_health_dataset_processed.csv` and submitted along with your notebook.

2 Heart Failure Prediction Dataset

Our second 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 using the links provided below. The dataset has been structured as a `csv` file, which is also being provided to you as part of this exercise.

- **Kaggle dataset link:** `https://bit.ly/heart-failure-clinical-data`
- **Original source link:** `https://bit.ly/uci-heart-failure-clinical-data`
- **Dataset `csv` filename:** `heart_failure_dataset.csv`

2.1. Load the dataset from the `csv` file into a pandas dataframe as follows (you are free to choose variable names of your liking):

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

Next, carry out basic exploration of data by addressing the following questions.

 waheed.bajwa@rutgers.edu

(a) (1 point) Based on your understanding of the dataset and its potential usage (refer to the Kaggle and source links provided above), is this a supervised machine learning task or an unsupervised machine learning task? Justify your answer as much as possible.

(b) (1 point) Print the `axes` and `dtypes` attributes of `heart_df` dataframe.

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

(d) (1 point) Using the `shape` attribute of `heart_df` dataframe, print the number of rows and columns within the `csv` file.

(e) (1 point) What is the total number of samples in this heart failure prediction dataset?

(f) (2 points) What is the number of independent variables (features, attributes, predictors, etc.) in this dataset? List down the names of these variables.

(g) (2 points) What is the number of dependent variables (if any) in this dataset? List down the names of these variables.

(h) (1 point) Suppose all data in `heart_df` dataframe corresponding to independent variables is extracted into a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. What would be the values of n and p in this case?

(i) (1 point) Suppose all data in `heart_df` dataframe corresponding to dependent variables is extracted into a matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$. What would be the values of n and m in this case?

(j) (3 points) Based on your understanding of the different variables in the dataset (refer to the Kaggle and source links provided above) as well as potential usage of functions such as `pandas.unique()` and `pandas.DataFrame.hist()`, determine the number of categorical variables in the dataset. List down the names of these variables and also justify your answer as much as possible.

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

(l) (1 point) Write code to determine how many samples in the dataset correspond to deceased patients and how many samples correspond to the remaining patients?

(m) (1 point) Write code to determine how many samples in the dataset correspond to women patients and how many samples correspond to male patients?

(n) (1 point) Write code to determine how many samples in the dataset correspond to smokers and how many samples correspond to non-smokers?

2.2. (4 points) Based on your understanding of the different variables in the dataset (refer to the Kaggle and the source links provided above), write code to check the logical consistency of values in `heart_df` dataframe and, if any inconsistent/invalid values are found, convert the invalid values to `NaN`. Justify your logic by carefully explaining it in a text cell.

2.3. (4 points) Write 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. (3 points) Write code to modify the processed dataframe so that the `DEATH_EVENT` variable is encoded using *one-hot encoding*. Save the final pre-processed dataframe as a `csv` file using `pandas.DataFrame.to_csv()` function. The file should be named `heart_failure_dataset_processed.csv` and submitted along with your notebook.

2.5. 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 understanding of the different variables in the dataset (refer to the Kaggle and source links provided above), why do you think it is logical that the second-most positively correlated variable with the `DEATH_EVENT` variable should indeed have a positive correlation with `DEATH_EVENT`?

(d) (2 points) Based on your understanding of the different variables in the dataset (refer to the Kaggle and source links provided above), why do you think it is logical that the two most negatively correlated variables with the `DEATH_EVENT` variable should indeed have negative correlation with `DEATH_EVENT`?

 waheed.bajwa@rutgers.edu