CMSC 435 Assignment 2

Fall 2022
(individual work; 12 pts total)

This assignment asks you to implement and evaluate two popular algorithms for the imputation of missing values using two provided datasets. You will evaluate and compare their runtime and the quality of the imputed values by comparing them with the corresponding values in the "complete" dataset.

Datasets

There are three datasets: the original dataset without missing values and two derived datasets where the missing values were introduced at two different amounts:

- *dataset_complete.csv* file is the complete dataset. It includes 10 features and 8795 objects.
- *dataset_missing01.csv* and *dataset_missing10.csv* files include the same dataset with 1% and 10% of missing values, respectively.

You will impute missing values in each of the latter two datasets and compare these imputed values to the corresponding true/correct values that are available in the *dataset_complete.csv* file to evaluate and compare different imputation algorithms. The three files are in the comma separated value (CSV) format. The first line defines the names of features and the remaining lines include the values of the corresponding 8795 objects. The features are numeric and continuous with values in [0, 1] interval.

Algorithms for missing data imputation

You will implement two algorithms for the imputation of missing values and apply each of them on the corresponding two datasets that have missing values: *dataset_missing005.csv* and *dataset_missing25.csv*.

Algorithm 1. *Mean imputation*

Missing value for a specific feature and object is imputed with the mean value computed using the complete values of this feature.

Example

	F1	F2	F3
Object 1	0.40256	0.14970	?
Object 2	0.41139	0.30140	?
Object 3	0.24752	0.32148	0.11169
Object 4	0.24609	?	0.13986
Object 5	?	0.58306	0.08910

To impute the missing value for feature F3 from object 1, we compute the mean of all complete values of F3: *mean* = (0.11169 + 0.13986 + 0.0891) / 3 = 0.11355.

	F1	F2	F3
Object 1	0.40256	0.14970	0.11355
Object 2	0.41139	0.30140	?
Object 3	0.24752	0.32148	0.11169
Object 4	0.24609	?	0.13986
Object 5	?	0.58306	0.08910

The imputed values **must not** be used to compute the means. Consequently, all missing values for a given feature are imputed with the same mean value.

	F1	F2	F3
Object 1	0.40256	0.14970	0.11355
Object 2	0.41139	0.30140	0.11355
Object 3	0.24752	0.32148	0.11169
Object 4	0.24609	?	0.13986
Object 5	?	0.58306	0.08910

Algorithm 2. *Hot deck imputation*

Missing values for features that have missing values in a given object are imputed with the values for the same features copied from another, the most similar object. First, similarity of a given object that has missing values with every other object in the dataset is computed using the Manhattan distance. The object with the smallest distance is assumed to be the most similar and its values are used for the imputation. If that object is missing some of the values that should be imputed then the second most similar object is used to impute the **remaining** missing values, and so on. In other words, you should use the first complete value that you find by screening objects by their increasing values of the distance. Given two objects $\mathbf{x} = \{x_1, x_2, \dots x_i, \dots, x_n\}$ and $\mathbf{y} = \{y_1, y_2, \dots, y_i, \dots, y_n\}$, the Manhattan distance is calculated as $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$ where n is the total number of features, x_i and y_i are values of feature i for objects **x** and **y**, respectively, and $x_i - y_i = 1$ if either x_i or y_i are missing values. The latter penalizes the use of objects that have missing values.

Example

	F1	F2	F3
Object 1	0.40256	0.14970	?
Object 2	0.41139	0.30140	?
Object 3	0.24752	0.32148	0.11169
Object 4	0.24609	?	0.13986
Object 5	?	0.58306	0.08910

To impute missing value for feature F3 from object 1, we compute distances to every other object

$d(obj1, obj2) = |0.40256 - 0.41139| + |0.14970 - 0.30140| + 1 = 1.16053$

$d(obj1, obj3) = |0.40256 - 0.24752| + |0.14970 - 0.32148| + 1 = 1.32682$

$d(obj1, obj4) = |0.40256 - 0.24609| + 1 + 1 = 2.1565$

$d(obj1, obj5) = 1 + |0.14970 - 0.58306| + 1 = 2.4336$

Since object 2, which is the most similar to object 1, has a missing value for the feature F3, the second nearest, object 3, is used and the missing value is imputed as follows

	F1	F2	F3
Object 1	0.40256	0.14970	0.11169
Object 2	0.41139	0.30140	?
Object 3	0.24752	0.32148	0.11169
Object 4	0.24609	?	0.13986
Object 5	?	0.58306	0.08910

The imputed values **must not** be used to compute the distances. In other words, all missing values for each feature are imputed based on the distances that use the dataset before the imputation. This ensures that the errors inherent in the imputed values are not propagated to compute the imputation.

Calculation of the imputation error

You will use the two datasets that were imputed with the two algorithms to calculate the corresponding four imputation errors. You will evaluate quality of these imputations based on the Mean Absolute Error (MAE) between the imputed values and the corresponding complete values that are available in the *dataset_complete.csv* file. This dataset should be used only to calculate MAE values, not to perform the imputations. The MAE values should be used to judge and compare the quality of each imputation.

Given the imputed values $\mathbf{x} = \{x_1, x_2, \dots x_i, \dots, x_N\}$ computed from a dataset that has missing values and the corresponding complete values $\mathbf{t} = \{t_1, t_2, \dots, t_i, \dots, t_N\}$ in the complete dataset, MAE is defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |x_i - t_i|$$

where N is the total number of missing values, x_i is a the imputed value in the dataset that has missing values, x_i and t_i are values for the same object and same feature in the two datasets, and $|\cdot|$ denotes the absolute value.

Example

Incomplete dataset

	F1	F2	F3	F4
Object 1	0.40256	0.14970	0.16870	?
Object 2	0.41139	0.30140	0.47033	?
Object 3	0.24752	0.32148	0.41167	0.11169
Object 4	0.24609	?	?	0.13986
Object 5	?	0.58306	0.52568	0.08910

Dataset where values were imputed using the mean imputation

	F1	F2	F3	F4
Object 1	0.40256	0.14970	0.16870	0.11355
Object 2	0.41139	0.30140	0.47033	0.11355
Object 3	0.24752	0.32148	0.41167	0.11169
Object 4	0.24609	0.33891	0.39409	0.13986
Object 5	0.32689	0.58306	0.52568	0.08910

Complete dataset

	F1	F2	F3	F4
Object 1	0.40256	0.14970	0.16870	0
Object 2	0.41139	0.30140	0.47033	0.14175
Object 3	0.24752	0.32148	0.41167	0.11169
Object 4	0.24609	0.21359	0.24071	0.13986
Object 5	0.70541	0.58306	0.52568	0.08910

Given the above imputation, the MAE is calculated as follows.

$MAE = \frac{1}{5}(|0.11355 - 0| + |0.11355 - 0.14175| + |0.33891 - 0.21359| +$
$|0.39409 - 0.24071| + |0.32689 - 0.70541|) = 0.1598$

The MAE values must be computed with precision of **four digits** after the decimal point.
The imputed values must be computed with precision of **five digits** after the decimal point.

Measurement of the runtime

You will measure and report the runtime for each of the four imputation tasks. The runtime must cover only the execution of the code that calculates imputation algorithm, including calculation of the imputed values and replacement of the missing values with the imputed values. The runtime must **exclude** reading the data (csv) files from disk, calculation of the MAE values, and writing the imputed data (csv) files to disk. You must quantify the runtime in **milliseconds**.

Implementation

Your code must perform imputation, display the four values of MAE and the four values of runtime on the screen, and save the four imputed datasets in the csv format. The **imputed datasets should be named as follows**:

Vnumber_missing01_imputed_mean.csv
Vnumber_missing01_imputed_hd.csv
Vnumber_ missing10_imputed_mean.csv
Vnumber_missing20_imputed_hd.csv

where *Vnumber* is your V number, e.g., *V12345678_missing01_imputed_mean.csv*

The MAE and runtime values should be displayed on the screen in the following format

MAE_01_mean = 0.1234
Runtime_01_mean = 124
MAE_01_hd = 0.1234
Runtime_01_hd = 124
MAE_10_mean = 0.5678
Runtime_10_mean = 56789
MAE_10_hd = 0.5678
Runtime_10_hd = 56789

You must use Java or Python 3 to implement all computations including loading the datasets from the csv files, coding the four imputation algorithms, calculation of the MAE values, printing the MAE values on the screen, and saving of the eight imputed datasets.

If you use Python 3:

- Your code should be in one source code (.py) file. You may define any number of classes and functions, but everything must be included in that file.
- You are only allowed to use NumPy (https://www.numpy.org/) and pandas (https://pandas.pydata.org) as imported libraries. They may help you with reading the csv files and working with the data.
- Your program will be tested on **Python 3.10** with the latest (as of today) versions of numpy and pandas installed. See the details of package versions here: https://github.com/sinaghadermarzi/vcu_datasci_2022F/blob/main/A2/README.md.
- This python file must successfully run on the above python environment and produce the above-mentioned outputs with the required precision. It should be run by executing the below command in the location where three input csv files are located.

```
Python3 a2.py
```

  For this, you need to make sure you read the input csv files from the current working directory.

If you use Java:

- All your code should be in one source code (.java) file. You may define any number of classes and functions, but everything must be included in that file.
- Your program will be tested on **Java 18** (latest version as of now).
- This java file must successfully compile and run with above version of Java and produce the above-mentioned outputs with the required precision. It should be run by executing the below commands in the location where three input csv files are located.

```
javac a2.java
java a2
```

  For this, you need to make sure your main class is named `a2`, you don't declare a package in `a2.java`, and you read the input csv files from the current working directory.

Deliverables

1. Java or Python source code in a single .java or .py file. The file **must be named** a2.java or a2.py.
2. Answers to the following five questions:
   2.1. What are the four MAE and the four runtime values? Copy the output from the screen.
   2.2. Which of the four results has the smallest error? Briefly explain **why** this result (i.e., make sure to consider both the corresponding **algorithm and dataset**) is better than the other three results.
   2.3. Give the computational complexity of the mean and the hot deck algorithms as a function of the number of objects *n* (use the big O notation). Do the runtime values that you measured agree with the computational complexity?
   2.4. Which of the four imputation tasks (i.e., make sure to consider both the corresponding **algorithm and dataset**) requires the longest runtime? Why is this runtime longer than the runtime of the three other tasks?
   2.5. Consider a median imputation algorithm, i.e., missing value for a specific feature and object is imputed with the median value computed using the complete values of this feature. Give the computational complexity of the median imputation algorithm as a function of the number of objects *n* (use the big O notation). Would the median imputation be **faster, slower and similar** in speed when compared to the mean imputation on our datasets? Would the median imputation be **faster, slower and similar** in speed when compared to the hot deck imputation on our datasets? Note: you do not have to implement the median imputation algorithm to answer this question.

Notes

- Achieving the lowest runtime (i.e., providing highly efficient implementation) is not necessary. The answers to the questions rely on comparing the runtime values in relative terms (when compared with each other), not as absolute values, since the absolute values depend on the efficiency of the code, choice of the programming language, and the hardware used. However, **your code should complete all computations in a matter of seconds or minutes, not hours**.
- Do not procrastinate and start early – **this assignment requires a substantial amount of time and effort**. Late submissions will be subject to deductions: 15% in first 12 hours and 30% for between 12 and 48 hours. We will not accept submissions that are over 48 hours late.
- We will check your source code, verify if it runs correctly, validate the results on the screen and in the files, and mark the answers to the five questions.
- We will **deduct** points if the files names and/or the outputs on the screen do not follow the above-defined format.
- We will check for **plagiarism**. Write your own code and provide your own answers.

Due Date

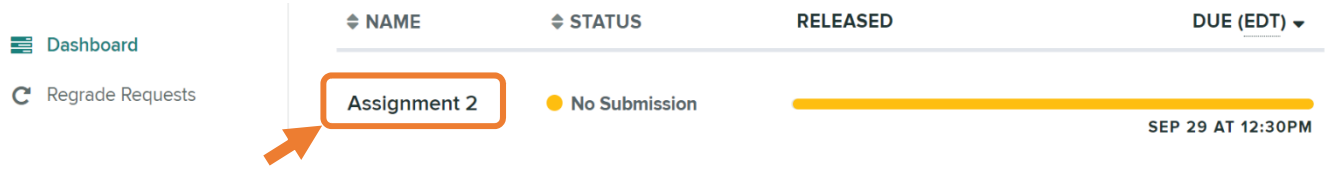
Your assignment must be received before 12:30 pm Eastern Time on September 29 (Thursday), 2022. Submissions must be done using Gradescope. Use the below instructions.

Signing up for Gradescope

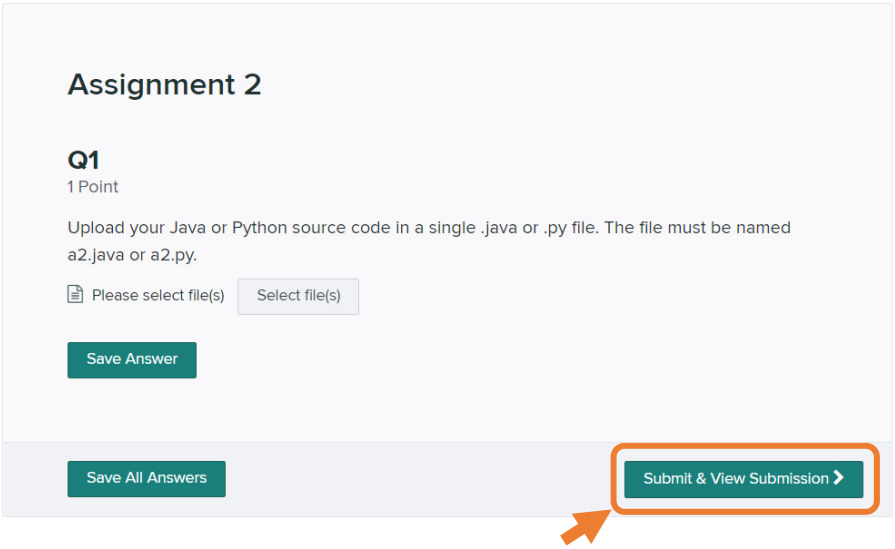
If you already have Gradescope account using your VCU email then you should be able to see the course (CMSC435 INTRODUCTION TO DATA SCIENCE) in your dashboard and find the assignment-2 in that course. Alternatively, you can go to the https://www.gradescope.com and use the signup button in the homepage. If you go through the Gradescope website, make sure to sign up using your official VCU student email and using the button "sign up as a student". If you have any problem signing up in Gradescope or finding the assignment in Gradescope the please contact ghadermarzis@vcu.edu.

Submitting assignment

Once you accessed the course in Gradescope successfully, click on “Assignment 2” in the dashboard



After answering the questions, you must click on the “Submit and View” button, which is at the bottom of the page, before the deadline to submit your assignment on-time. The “Save answer” or “Save All Answers” buttons only save your answers and do not submit the assignment.



After clicking the submit you will be able to review your submission and double check that it is complete. You can submit multiple times and only your last submission will be graded.