FIT5196 Data wrangling S1 2018

Dashboard ▶ Faculty of Information Technology ▶ Active Units ▶ FIT Units ▶ FIT S1 2018 ▶ FIT5196-S1-2018 ▶ My Assessments ▶ Assessment 2: Cleansing and Integrating Raw Data

Assessment 2: Cleansing and Integrating Raw Data

Please carefully review all the requirements below to ensure you have a good understanding of what is required for your assessment.

1. Due Date

This specific due date and time can be **viewed below in the Grading Summary**.

2. Assessment Description

For this assessment, you are required to write Python (Python 2/3) code to analyze existing datasets, find and fix problems in those datasets and integrate two datasets which are in the different format. This assessment contains **four major tasks** that are specified as follows:

Task 1. Auditing and Cleansing the Job dataset

Input and output of this task are shown below:

Input	Output	Jupyter notebook
dataset1_with_error.csv	dataset1_solution.csv	task1_ass2.ipynb

The dataset description is shown below:

COLUMN	DESCRIPTION
Id	8 digit Id of the job advertisement
Title	Title of the advertised job position
Location	Location of the advertised job position
ContractType	The contract type of the advertised job position, could be full-time, part-time or non-specified.
ContractTime	The contract time of the advertised job position, could be permanent, contract or non-specified.
Company	Company (employer) of the advertised job position
Category	The Category of the advertised job position, e.g., IT jobs, Engineering Jobs, etc.
Salary per annum	Annual Salary of the advertised job position, e.g., 80000
OpenDate	The opening time for applying for the advertised job position, e.g., 20120104T150000, means 3pm, 4th January 2012.
CloseDate	The closing time for applying for the advertised job position, e.g., 20120104T150000, means 3pm, 4th January 2012.
SourceName	The website where the job position is advertised.

In this task, you are required to inspect and audit the data (**dataset1_with_error.cs**v) to identify the data problems, and then fix the problems. Different generic and major data problems could be found in the data **might** include:

- Lexical errors
- Irregularities
- Violations of the Integrity constraint.
- Inconsistency

In the end, save the error-free dataset in dataset1_solution.csv. The number of records in your solution should be the same as the number of those in the input file.

Task 2. Integrating the Job datasets:

Input and output of this task are shown below:

Input	Output	Jupyter notebook
dataset2_integration.csv dataset1_solution.csv	dataset1_dataset2_solution.csv	task2_ass2.ipynb

To complete this task successfully, you are required to do the following:

Step 1: Resolving schema conflicts and merging data

Inspect **dataset1(dataset1_solution.csv) and dataset2 (dataset2_integration.csv)** schema in order to find any schema conflicts.

1. Write Python code for data inspection to find schema conflicts.
2. Choose and implement an appropriate method for resolving the problem.
3. You will need to adapt the schema in **dataset1** for your global schema as much as you could (please **AVOID** changing the attribute names).
4. Write your Python code to implement the semantic mapping and merge between **dataset1** and **datset2** to produce one unified table

Step 2: Resolving data conflicts:

Inspect tuples and instances for data conflicts in the unified table. In this step, you are required to do the following:

1. Use Pandas libraries to detect and resolve duplication in the unified table.
2. Find a proper global key for the integrated job data and explain your key in the notebook.

Task 3. Finding missing value and fill in the reasonable values

Input and output of this task are shown below:

Input	Output	Jupyter notebook
dataset3_with_missing.csv	dataset3_solution.csv	task3_ass2.ipynb

The dataset description is shown below:

ATTRIBUTE	DESCRIPTION
Id	Sale Id
date	Date of the property sold, e.g., 20140502T000000
price	Property sold price
bedrooms	Number of bedrooms
bathrooms	Number of bathrooms, the value of which can be either an integer or a fraction ending with .25, .5, and .75. For example, 0.5 accounts for a room with a toilet but no shower
sqft_living	Square footage of the property's interior living space
sqft_lot	Square footage of the land space
floors	Number of floors
waterfront	Whether the property was overlooking the waterfront or not
view	An index from 0 to 4 of how good the view of the property was
condition	An index from 1 to 5 on the condition of the property.
sqft_above	The square footage of the interior living space that is above ground level
sqft_basement	The square footage of the interior living space that is below ground level
yr_built	The year the property was initially built
yr_renovated	The year of the property's last renovation
zipcode	The zip code area where the property is, which contains state and zip code, separated by a space.
lat	Latitude of the property
long	Longitude of the property

In this task, you are required to find impute all the missing values by inspecting and analyzing the dataset3(dataset3_with_missing.csv). In the end, save the fixed table in **dataset3_solution.csv**.

Task 4. Finding the outliers

Input and output of this task are shown below:

Input	Output	Jupyter notebook
dataset4_with_outliers.csv	dataset4_solution.csv	task4_ass2.ipynb

This dataset has the same description as in Task 3.

In this task, you are required to identify the outliers and **delete those rows with outliers** by analyzing the dataset4 **(dataset4_with_outliers.csv)**

In summary:

dataset1, dataset2, dataset3 and dataset4 are provided, and the download links are embedded in the following table.

Task	Input	Output	Jupyter notebook
Task 1	dataset1_with_error.csv	dataset1_solution.csv	task1_ass2.ipynb
Task 2	dataset2_integration.csv dataset1_solution.csv	dataset1_dataset2_solution.csv	task2_ass2.ipynb
Task 3	dataset3_with_missing.csv	dataset3_solution.csv	task3_ass2.ipynb
Task 4	dataset4_with_outliers.csv	dataset4_solution.csv	task4_ass2.ipynb

This is an individual assignment and worth 40% of your total mark for FIT5196.

3. Assessment Resources

Before you start writing your code, you will need to read the following materials:

- While using Jupyter Notebook, you should consider the use of different cells to make notes as you go. However, be precise and concise, please do not include things that are not relevant to the tasks in your notebook.
- Standards for commenting code are available online (e.g. **https://www.python.org/dev/peps/pep-0008/#comments).** You must ensure that you can clearly explain what your code does with clear comments.
- The work required to finish this assessment should be your own. If you use resources elsewhere, make sure that you properly acknowledge them in your notebook.You may need to review the FIT citation style tutorial to make you're familiar with appropriate citing and referencing for this assessment. Also, review the demystifying citing and referencing for help.

and download the data file:

- dataset1_with_error.csv
- dataset2_integration.csv
- dataset3_with_missing.csv
- dataset4_with_outliers.csv

4. Assessment Criteria

The following outlines the criteria which you will be assessed against.

4.1 Grading Rubric

Specific grading details for this assessment are:

1. The submitted scripts in the notebook should work **without any errors** and must give **the correct results**. If the submitted notebook cannot be run by the assessor, which will be double-checked by the head tutor and the lecturer, zero marks will then be given to the corresponding task.
   - task 1: 7 out of 40
   - task 2: 7 out of 40
   - task 3: 10 out of 40
   - task 4: 12 out of 40
2. The code should be well structured and properly commented. (2 out of 40)
3. The notebook should be structured in a logical way so that it clearly shows how students finish the tasks in the assessment. (2 out of 40)
4. Criteria 2 and 3 will be assessed **if and only if** the mark for criteria 1 is greater than and equal to 28.

4.2 Penalties

- Late submission: for all assessment items handed in after the official due date, and without an approved extension, a 10% penalty applies to the student's mark for each day after the due date (including weekends, and public holidays) for up to 5 days. **Assessment items handed in after 5 days will not be considered!**
- Submission: please do follow **Section 5 How to Submit** to submit your assignment. Otherwise, a 5% penalty will be applied.

5. How to Submit

Once you have completed your work, take the following steps to submit your work.

1. Only one zip file needs to be submitted named as [**surname_studentID]_ass2.zip**: In total, there are 8 files. It includes 4 tasks with two files each, one is the output dataset and the other is the Jupyter notebook, the naming of the files must follow the names in the below table.

Output	Jupyter notebook
dataset1_solution.csv	task1_ass2.ipynb
dataset1_dataset2_solution.csv	task2_ass2.ipynb
dataset3_solution.csv	task3_ass2.ipynb
dataset4_solution.csv	task4_ass2.ipynb