RMIT

Computer Science & IT, School of Science

COSC2670/COSC2738 — Practical Data Science

Assignment 1: Data Cleaning and Summarising

Due: 23:59, Thursday, the 12th of April, 2018 (week 6)

This assignment is worth 15% of your overall mark.

Introduction

In this assignment, you will examine a data file and carry out the first steps of the data science process, including the cleaning and exploration of data.

You will need to develop and implement appropriate steps, in IPython, to load a data file into memory, clean, process, and analyse it.

This assignment is intended to give you practical experience with the typical first steps of the data science process.

The "Practical Data Science" Canvas contains further announcements and a discussion board for this assignment. Please be sure to check these on a regular basis – it is your responsibility to stay informed with regards to any announcements or changes. Login through `https://rmit.instructure.com/`.

Where to Develop Your Code

You are encouraged to develop and test your code in two environments: **Jupyter Notebook on Lab PCs** and **Teaching Servers**.

Jupyter Notebook on Lab PCs

On Lab Computer, you can find Jupyter Notebook via:

Start → All Programs → Anaconda2 (64-bit) → Jupyter Notebook

Then,

- Select New → Python 2
- The new created '*.ipynd' is created at the following location:
  - C:\Users\sXXXXXXX
  - where sXXXXXXX should be replaced with a string consisting of the letter "s" followed by your student number.

Teaching Servers

Three CSIT teaching servers are available for your use:

`(titan|saturn|jupiter).csit.rmit.edu.au`.

Details for how to access these servers are available in ‘‘`Extra: Run Anaconda on RMIT Coreteaching Servers`’’ under the `Modules/Week2: Data Curation` section of the course Canvas. You are encouraged to develop your code on these machines.

If you choose to develop your code elsewhere, it is your responsibility to ensure that your assignment submission can be successfully run using the version of IPython installed on Lab PCs or `(titan|saturn|jupiter).csit.rmit.edu.au`, as this is where your code will be run for marking purposes.

Important: You are required to make regular backups of all of your work. This is good practice, no matter where you are developing your assignment solutions.

Plagiarism

RMIT University takes plagiarism very seriously. All assignments will be checked with plagiarism-detection software; any student found to have plagiarised will be subject to disciplinary action as described in the course guide. Plagiarism includes submitting code that is not your own or submitting text that is not your own. Allowing others to copy your work is also plagiarism. All plagiarism will be penalised; there are no exceptions and no excuses. For further information, please see the *Academic Integrity* information at `http://www1.rmit.edu.au/academicintegrity`.

General Requirements

This section contains information about the general requirements that your assignment must meet. *Please read all requirements carefully before you start.*

- You *must* do the analysis in IPython.
- You *must* include a plain text file called “readme.txt” with your submission. This file should include your name and student ID, and instructions for how to execute your submitted script files.
- Parts of this assignment will include a written report, this *must* be in *PDF* format.
- Please ensure that your submission follows the file naming rules specified in the tasks below. File names are case sensitive, i.e. if it is specified that the file name is `gryphon`, then that is exactly the file name you should submit; `Gryphon`, `GRYPHON`, `griffin`, and anything else but `gryphon` will be rejected.

Task 1: Data Preparation (5%)

Have a look at the file `Bank.csv`, which is available in Canvas under the `Assignments/Assignment 1` section of the course Canvas.

The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

The data set includes the following attributes:

1. **age** (numeric)
2. **job**: type of job (categorical: 'admin', 'blue-collar', 'entrepreneur', 'housemaid', etc)
3. **marital**: marital status (categorical: 'divorced', 'married', 'single','unknown'; note: 'divorced' means divorced or widowed)
4. **education** (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', etc)
5. **default**: has credit in default? (categorical: 'no','yes','unknown')
6. **housing**: has housing loan? (categorical: 'no','yes','unknown')
7. **loan**: has personal loan? (categorical: 'no','yes','unknown')

\# *Attributes related to the last contact of the current campaign:*

8. **contact**: contact communication type (categorical: 'cellular','telephone')
9. **month**: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10. **day_of_week**: last contact day of the week (categorical: 'mon', 'tue', 'wed', 'thu', 'fri')
11. **duration**: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call, y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

\# *Other attributes:*

12. **campaign**: number of contacts performed during this campaign and for this client (numeric, includes last contact)

13. **pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14. **previous**: number of contacts performed before this campaign and for this client (numeric)
15. **poutcome**: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

Social and economic context attributes:

16. **emp.var.rate**: employment variation rate - quarterly indicator (numeric)
17. **cons.price.idx**: consumer price index - monthly indicator (numeric)
18. **cons.conf.idx**: consumer confidence index - monthly indicator (numeric)
19. **euribor3m**: euribor 3 month rate - daily indicator (numeric)
20. **nr.employed**: number of employees - quarterly indicator (numeric)

Output variable (desired target):

21. **y**: has the client subscribed a term deposit? (binary: 'yes','no')

Being a careful data scientist, you know that it is vital to carefully check any available data before starting to analyse it. Your task is to prepare the provided data for analysis, by carrying out the following steps:

1. Load the CSV data from the file. You need to use an appropriate pandas function to load the csv data, and make use of the correct arguments including *sep*, *decimal*, *header*, *names*, if needed.
2. Check whether the loaded data is equivalent to the data in the source (CSV) file. That is, you will need to ensure that the loaded data has appropriate data types assigned, or take steps to ensure that the appropriate types are used.
3. Check whether there are *typos* in the data. If there are any typos, correct them by using masks.
4. Check whether there are instances of *extra whitespaces* in the data, and if so, demonstrate how to remove them by calling on an appropriate function.
5. Demonstrate how to cast text data to lower-case, using an appropriate function.
6. Design and run a small test-suite, consisting of a series of sanity checks to test for the presence of impossible values for each attribute.

7. Check whether the loaded data has any *missing values*. If so, use an appropriate function to replace them with the *column-wise* mean value.

Task 2: Data Exploration (5%)

Explore the provided data based on the following steps:

1. Create a visualization for each column by producing an appropriate type of graph.

    - You should explore each column with at least one type of graph, but you can explore with more than one type, including histograms, barcharts, pie graphs, or boxplots.
    - Format each graph carefully. You need to include appropriate labels on the x-axis and y-axis, a title, and a legend. The fonts should be sized for good readability. Components of the graphs should be coloured appropriately, if applicable.

2. Explore the relationships between columns. You may choose which pairs of columns to focus on, but you need to generate at least 3 visualisations for this subtask. These should address a plausible hypothesis for the data concerned. For example, you might wonder: is there a relationship between the `age` and the `cons.price.idx`? An appropriate visualisation for this could be to graph `age` against `cons.price.idx`.

3. Build a *scatter matrix* for all numerical columns.

Task 3: Report (5%)

Write your report and save it in a file called `report.pdf` (it must be in PDF format) and answer the questions below. Remember to clearly cite any sources (including books, research papers, course notes, etc.) that you referred to while designing aspects of your programs.

- Create a heading called "Data Preparation" in your report.
- For each numbered step in Task 1 above, create a sub-section with corresponding numbering, and provide a brief explanation of how you addressed the task, and explain any choices that you made (if appropriate). As part of this exercise, you must specifically list any data rows that you changed.
- Create a heading called "Data Exploration" in your report.
- For each numbered step in Task 2 above, create a sub-section with corresponding numbering.

- In subsection 1, include *all* of your graphs from Task 2, Step 1. Under each graph, include a brief explanation of why you chose this graph type(s) to represent the data in a particular column.
- In subsection 2, include your plots from Task 2, Step 2. With each plot, state the hypothesis that you are investigating. Then, briefly discuss any interesting relationships (or lack of relationships) that you can observe from your visualisation.
- In subsection 3, present your scatter matrix.

Optional Extension: Analysis of Missing Values (Up to 1.5% bonus marks for practical component)

ONLY attempt this section if you have completed all previous sections of the assignment.

In Task 1, Step 7, you were asked to deal with missing values by including the column-wise mean.

Now, your task is to deal with the missing values in the data sets with other options:

- replacing them with a fixed value
- replacing with the median value (column-wise)
- ignoring all observations containing missing values

For each of these approaches, choose a data column, and produce a new graph (corresponding to the initial graph that you produced in Task 2, Step 1).

In your `report.pdf` file, create a heading called "Extension". In this section, include your three graphs. Under each one, briefly discuss the impact that the different approachs to dealing with missing values have on what you observe from the visualisation.

What to Submit, When, and How

The assignment is due at

23:59, Thursday, the 12th of April, 2018 (week 6).

Assignments submitted after this time will be subject to standard late submission penalties.

There are three files you need to submit:

- Notebook file containing your python commands for Task 1, 'Task1.ipynb'.
- Notebook file containing your python commands for Task 2, 'Task2.ipynb'.

For the notebook files, please make sure to clean them and remove any unnecessary lines of code (cells). Follow these steps before submission:

1. Main menu → Kernel → Restart & Run All
2. Wait till you see the output displayed properly. You should see all the data printed and graphs displayed.

- Your `report.pdf` file.

They must be submitted as ONE single zip file, named as your student number (for example, 1234567.zip if your student ID is s1234567). The zip file must be submitted in Canvas:

Assignments/Assignment 1.

Please do NOT submit other unnecessary files.