FIT5196-S2-2019 assessment 3

This is a group assessment and worth 30% of your total mark for FIT5196.

Due date: **11:55 pm, 7 October 2019.**

Data Cleansing (%60)

For this assessment, you are required to write Python (Python 2/3) code to analyze your dataset, find and fix the problems in the data. The input and output of this task are shown below:

Table 1. The input and output of the task

Input	Output	Other Deliverables
<GroupName>_dirty_data.csv <GroupName>_outlier_data.csv <GroupName>_missing_data.csv	<GroupName>_dirty_data_solution.csv <GroupName>_outlier_data_solution.csv <GroupName>_missing_data_solution.csv	<GroupName>_ass3.ipynb <GroupName>_ref-diary.pdf <GroupName>_dec_form.pdf

Note1: All files must be zipped into a file named *<GroupName>_ass3.zip*

Note2: <GroupName> is to be replaced with your group name exactly as mentioned in the given datasets (example: Group001)

Note3: Each group can find their three input files here

Exploring and understanding the data is one of the most important parts of the data wrangling process. You are required to perform graphical and/or non-graphical EDA methods to understand the data first and then find the data problems. You are required to:

- Detect and fix errors in *<GroupName>_dirty_data.csv*
- Detect and **remove** outlier rows in *<GroupName>_outlier_data.csv*
  (outliers are to be found w.r.t. *delivery_fee* attribute)
- Impute the missing values in *<GroupName>_missing_data.csv*

As a starting point, here is what we know about the dataset in hand:

The dataset contains Food Delivery data from a restaurant in Melbourne, Australia. The restaurant has three branches around CBD area. All three branches share the same menu but they have different management so they operate differently.

Each instance of the data represents a single order from said restaurant. The description of each data column is shown in Table 2.

Table 2. Description of the columns

COLUMN	DESCRIPTION
order_id	A unique id for each order
date	The date the order was made, given in YYYY-MM-DD format
time	The time the order was made, given in hh:mm:ss format
order_type	A categorical attribute representing the different types of orders namely: Breakfast, Lunch or Dinner
branch_code	A categorical attribute representing the branch code in which the order was made. Branch information is given in the *branches.csv* file.
order_items	A list of tuples representing the order items: first element of the tuple is the item ordered, and the second element is the quantity ordered for such item.
order_price	A float value representing the order total price
customer_lat	Latitude of the customer coming from the *nodes.csv* file
customer_lon	Longitude of the customer coming from the *nodes.csv* file
customerHasloyalty?	A logical variable denoting whether the customer has a loyalty card with the restaurant (1 if the customer has loyalty and 0 otherwise)
distance_to_customer_KM	A float representing the shortest distance, in kilometers, between the branch and the customer nodes with respect to the *nodes.csv* and the *edges.csv* files. Dijkstra algorithm can be used to find the shortest path between two nodes in a graph. Reading materials can be found here.
delivery_fee	A float representing the delivery fee of the order

Notes:

1. The output *csv* files **must** have the exact same columns as the input.
2. There is at least one anomaly in the dataset from each category of the data anomalies (i.e., syntactic, semantic, and coverage).
3. In the file *<GroupName>_dirty_data.csv,* any row can carry no more than one anomaly. (i.e. there can only be one anomaly in a single row and all anomalies are fixable)
4. There are no data anomalies in the file *<GroupName>_outlier_data.csv,* only outliers. Similarly, there are no data anomalies other than missing value problems in the file *<GroupName>_missing_data.csv*

5. There are three types of meals:
    a. Breakfast - served during morning (8am - 12pm),
    b. Lunch - served during afternoon (12:00:01pm - 4pm)
    c. Dinner - served during evening (4:00:01pm - 8pm)

    Each meal has a distinct set of items in the menu (ex: breakfast items can't be served during lunch or dinner and so on).
6. A useful python package to solve a linear system of equations is numpy.linalg

7. Delivery fee is calculated using a different method for each branch.
    The fee depends linearly (but in different ways for each branch) on:
    a. weekend or weekday (1 or 0) - as a continuous variable
    b. time of the day (morning 0, afternoon 1, evening 2) - as a continuous variable
    c. distance between branch and customer

    If a customer has loyalty, they get a 50% discount on delivery fee

8. The restaurant uses Djikstra algorithm to calculate the shortest distance between customer and restaurant. (explore **networkx** python package for this or alternatively find a way to implement the algorithm yourself)
9. As EDA is part of this assessment, no further information will be given publicly regarding the data. However, you can brainstorm with the teaching team during tutorials and consultation sessions.

Methodology (%25)

The report should demonstrate the methodology (including all steps) to achieve the correct results.

Documentation (%15)

The cleaning task must be explained in a well-formatted report (with appropriate sections and subsections). Please remember that the report must explain the complete EDA to examine the data, your methodology to find the data anomalies and the suggested approach to fix those anomalies.