FIT5196-S2-2020 assessment 2

This is an individual assessment and worth 35% of your total mark for FIT5196.

Due date: **11:55 pm, 11 October 2020.**

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	Output Notebook
<student_id>_dirty_data.csv <student_id>_outlier_data.csv <student_id>_missing_data.csv	<student_id>_dirty_data_solution.csv <student_id>_outlier_data_solution.csv <student_id>_missing_data_solution.csv	<student_id>_ass2.ipynb <student_id>_ass2.pdf

Note1: You should submit a zip file and a pdf file which will be used for plagiarism check.

1. **The csv files and the ipynb file must be zipped into a file named** <student_id>_ass2.zip.
2. **The pdf file should be exported from the** <student_id>_ass2.ipynb **without any cell output (please only keep the markdown notes and scripts in the pdf file). The pdf file is named** <student_id>_ass2.pdf.

Note2: <student_id> is to be replaced with your student id

Note3: Students can find their three input files here based on their student_id

Note 4: An interview is required for this assessment. You will need to explain your solution and answer questions from a teaching team member.

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 and fix the data problems. You are required to:

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

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

The dataset contains transactional retail data from an online electronics store (DigiCO) located in Melbourne, Australia[1]. The store operation is exclusively online, and it has three warehouses around Melbourne from which goods are delivered to customers.

Each instance of the data represents a single order from said store. 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
customer_id	A unique id for each customer
date	The date the order was made, given in YYYY-MM-DD format
nearest_warehouse	A string denoting the name of the nearest warehouse to the customer
shopping_cart	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 denoting the order price in AUD. The order price is the price of items before any discounts and/or delivery charges are applied.
customer_lat	Latitude of the customer's location
customer_long	Longitude of the customer's location
coupon_discount	An integer denoting the percentage discount to be applied to the order_price.
distance_to_nearest_warehouse	A float representing the arc distance, in kilometres, between the customer and the nearest warehouse to him/her. (radius of earth: 6378 KM)
delivery_charges	A float representing the delivery charges of the order
order_total	A float denoting the total of the order in AUD after all discounts and/or delivery charges are applied.
season	A string denoting the season in which the order was placed. Refer to this link for details about how seasons are defined.

[1] The dataset is fictional

is_expedited_delivery	A boolean denoting whether the customer has requested an expedited delivery
latest_customer_review	A string representing the latest customer review on his/her most recent order
is_happy_customer	A boolean denoting whether the customer is a happy customer or had an issue with his/her last order.

Notes:

1. The output *csv* files **must** have the exact same columns as the input. Any misspelling or mismatch will lead to a malfunction of the auto-marker which will in turn lead to losing marks.
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 *<student_id>_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, if there is no possible way to fix it, it is not an anomaly)
4. There are no data anomalies in the file *<student_id>_outlier_data.csv,* only outliers. Similarly, there are no data anomalies other than missing value problems in the file *<student_id>_missing_data.csv*
5. The retail store has three different warehouses in Melbourne (see warehouses.csv for their locations)
6. The retail store focuses only on 10 branded items and sells them at competitive prices.
7. A useful python package to solve linear equations is numpy.linalg
8. The store has different business rules depending on the season to match the different demands of each season. For example, delivery charge is calculated using a linear model which differs depending on the season. The model depends linearly (but in different ways for each season) on:
    1. Distance between customer and nearest warehouse
    2. Whether the customer wants an expedited delivery
    3. Whether the customer was happy with his/her last purchase (if no previous purchase, it is assumed that the customer is happy)
9. To check whether a customer is happy with their last order, the customer's latest review is classified using a sentiment analysis classifier. *SentimentIntensityAnalyzer* from nltk.sentiment.vader is used to obtain the polarity score. A sentiment is considered positive if it has a 'compound' polarity score of 0.05 or higher and is considered negative otherwise. Refer to this link for more details on how to use this module.
10. If the customer provided a coupon during purchase, the coupon discount percentage will be applied to the order price before adding the delivery charges (i.e. the delivery charges will never be discounted).
11. Also, we know that the following attributes are always correct (i.e. don't look for any errors in dirty data for them):
    1. coupon_discount
    2. delivery_charges
    3. The ordered quantity values in the shopping_cart attribute
12. 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.