Huong Giang Bui

Student ID: 37306207

COSC480 – Computer Programming

Due date: 21/10/2022

Python Project

SALES MANAGEMENT PROGRAMME

1. Introduction

Throughout five years working as a Sales Representatives for a fertiliser company in Vietnam, I understand how important an effective sales process management is. With the idea of building an efficient sales management programme, this project was born. The programme is designed for Area Sale Managers for fertilizer companies, which specialised in managing sales in two provinces (I used Ninh Thuan and Khanh Hoa in Vietnam for my project). With this programme, the users can access every single client or product information with their ID number and have a visual look at how they perform throughout the year. The programme also allows users to sort out the top best-selling products for a particular client and the top best buyers for a specific product.

Moreover, this programme facilitates the annual year report so that the users can make appropriate decisions on which months/ clients/products the company should focus on to boost productivity for the sales department. In particular, the report functions allows users to view pretty charts of sales performance by province, category, or month. Also, the programme returns a list of "diamond clients" with an adjustable revenue threshold.

2. How to Interact with programme

2.1. Dataset preparation:

- You will need to have the CSV files as below:
- ***Client_data.csv file***, which contains essential information about clients. The file has six columns: Client ID, Store name, Owner, Phone number, email, and province.
- ***Product_data.csv file***, which contains essential information about products. There are five columns in the file: Product ID, Category, Product Name, Specification, Unit, Price (NZD)
- **Order_data.csv file**, which contains the order information of the year. There are six columns in the file:

- ORDER ID: Order ID is unique for every order.
- Client ID: The client ID is taken from client_data.csv
- Product ID: The product ID is taken from product_data.csv
- Date: the day of the order (dd/mm/yyyy)
- Price: the price of the product which is converted to NZD
- Quantity: number of the unit the order has
- Total: the total amount of money for that order.

- ***Monthly_target.csv file***, which contains the target amount of revenues (in NZD), months (integer from 1 to 12)

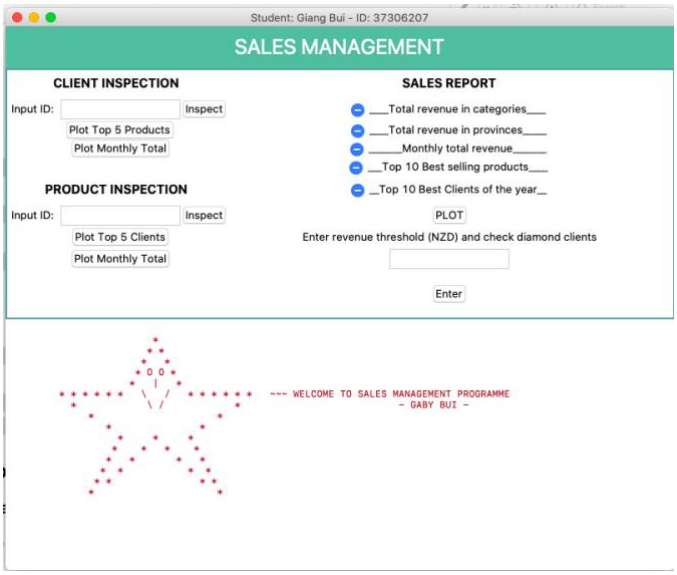
2.2. Run the program:

This programme should be able to run on any Python IDE application. In my situation, I have chosen Pycharm as the integrated development environment with fine interface experience and handy functions facilitating the programming process.

The programme also is supported by some Python libraries/modules. Therefore, installing all of them is required, including Pandas, Numpy, Matplotlib, Prettytable, and Tkinter.

Also, we need to make sure all the supported modules are in the same directory with 'main.py': 'client_module.py', 'product_module.py', 'report_module.py', 'guimodule.py', and 'art.py'.

To run the programme, we execute the main.py file by choosing run\run 'main'. After that, the Tkinter GUI window will pop up as below:



Some widgets are displayed in the **top left of the window**, allowing users to input the "Client ID" or "Product ID". If the users input the wrong "Client ID" or "Product ID", there is a "warning message" will display on the result area and asks users to input the ID again. There are three buttons for each client/product inspection feature with different purposes:

- "*Inspect*" button links with command of printing out the information of a particular client/product item. The layout is supported by Prettytable Module installed from the Python library.
- "*Plot Top 5 Products*" and "Plot Top 5 Clients" link with command of plotting bar charts for the top 5 best-selling products for client object and top 5 best buyers for product object. The plot can be adjusted or downloaded by clicking on the bottom left buttons of the plot window.
- "*Plot monthly Total*" link with command of plotting a bar chart for a particular object's monthly total revenue. Similarly, those charts also can be adjusted or downloaded by clicking on the bottom left buttons of the plot window.

The annual report can be retrieved by making commands to the programme. On **the top right corner of the Tkinter window**, there is a range of plotting options that users can choose from by clicking on the radio buttons on the screen.

- The results of "*Total revenue in categories/provinces*" are pie charts that display the total revenue of each category/province. Those pie charts also can be adjusted or downloaded as the users wish to do so.
- Monthly total revenues are plotted with the option "Monthly total revenue". This chart allows users to inspect which months hit the target and which did not. The plot is a combination of a bar and a line graph. The bars indicate the total monthly revenues, and the lines indicate the target being set from the beginning of the year.
- The "Top 10 best-selling products"/ "Top 10 best clients of the year" button allow users to inspect the top 10 products that clients are interested in and the top 10 Clients making

the most benefit for the company in that specific market. This will help the company have a good strategy on which clients/products to focus on to boost sales productivity in the inspected market.

Below the plotting options are the "**Diamond clients inspection**". When inputting a threshold of the year's total revenue, the programme will retrieve a list of clients who achieved that target. The client list is well-organized using the pretty table from the PrettyTable module from Python Library.

3. Development process

3.1. Tool and libraries used in the programme:

The programme is supported by some libraries, including: pandas, NumPy, matplotlib.pyplot, prettytable, and tkinter.

- **Pandas**: some CSV data files were converted to pandas data frames, allowing programme to read files and process data efficiently.
- **Numpy**: facilitates programme calculation.
- **Matplotlib.pyplot** is used for plotting graphs.
- **Prettytable** is used to refine the data tables printed on the console area.
- **Tkinter** is used to build the graphic user interface for the programme.

3.2. Programme's structure:

To make the code easy to read and retrieve, the programme is approached by Object Oriented Programme method, and the code is organised into six modules: "**client_module.py**", "**product_module.py**", "**report_model.py**", "**art.py**", "**guimodule.py**" and "**main.py**". In addition, there are 4 CSV files including in the same directory with those python files, which are "**client_data.csv**", "**product_data.csv**", "**order_data.csv**", "**monthly_target.csv**".

Module **"client_module.py"** only works with the class **"Client",** which is the blueprint for client object . The class has four attributes: *id, name, phone, and province*, which are used for the methods:

- Method "*client_details*": organises and returns a table of the client's details.
- Method "*add_order*": fetches all orders of client object from order data and returns client's order data
- Method "*total_overturn*": calculates total overturn of a particular client object
- Method "*plot_top_5_products*": Plots bar chart from client's top 5 best-selling products
- Method "*plot_monthly_total*": plots monthly total for client overturn

Module **"product_module.py"** only works with the class **"Product",** which is the blueprint for product object**.** The class has six attributes: *product_id, category, name, specification, unit, and price,* which are used for the methods:

- Method "*product_details*": organises and returns a table of product details
- Method "*add_order*": fetches all orders of product object from order data and returns order data for the product
- Method "*total_overturn*": calculates total overturn of product object
- Method "*plot_top_5_clients*": Plots top 5 best-sellers for the product
- Method "*plot_monthly_total*": plots monthly total for the product object

Module "**report_module.py**" only works with class "Report" which is the blueprint for report object. The class has four attributes: *order_data, product_list, client_list, and monthly_target*, which are used for the methods:

- Method "*client_id_list*": gets list of client's IDs
- Method "*product_id_list*": gets list of product's IDs
- Method "*category_revenue*": calculates total revenue for each product's category
- Method "*province_revenue*": calculates total revenue for each province
- Method "*autopct_format*": creates pct values for bar plotting
- Method "*plot_category_revenue*": plots bar chart for category revenues
- Method "*plot_province_revenue*": plots piechart based on province total revenue
- Method "*plot_monthly_revenue*": plots bar chart for monthly revenue
- Method "*plot_top_10_clients*": plots Top 10 Clients of the year:
- Method "*plot_top_10_Products*": plots Top 10 Products of the year
- Method "*print_diamond_clients*": returns a list of clients with revenue equal to or more than the threshold

Module **"guimodule.py"** only works with the class **"SalesManagementGui".** The __init__ function sets up the frames and some widgets on those frames, such as texts, labels, entries, buttons, and radio buttons. The window frame (root) contains 'header' and 'body' frames. 'body' frame contains 'top_body' and 'bottom_body' frames. Buttons are integrated with commands such as *inspecting a client/product* and *plotting bar graphs, line graphs, and pie charts.*

Methods of the class "SalesManagementGui":

- Method "*print_client_info*": Prints client information
- Method "*get_client_info*": gets client information
- Method "*plot_top5_product*": Plots top 5 products of a particular client
- Method "*plot_client_monthly_total*": Plots monthly total from the client inspected
- Method "*print_product_info*": Prints product information
- Method "*get_product_info*": gets product information
- Method "*plot_top5_clients*": Plots top 5 clients buying this product
- Method "*plot_product_monthly_total*": Plots monthly total from the product inspected
- Method "*plot_annual_report*": plots total revenues in categories, months, provinces, top 10 products, or top 10 Method Clients:

- Method "*print_vip_client*": Prints VIP client information with the threshold input from users.

Module **"main.py"** can use all codes from all modules above by importing all of them into it. "main.py" including some functions working with csv file:

- Function "*read_file*" takes a csv file and returns a pandas data frame. We used this function for handling "order_data.csv" and "monthly_target.csv" files.
- Function "*read_client*" takes a client csv file and returns a list of client objects
- Function "*read_product*" takes a product csv file and return a list of product objects
- Function "*return_object"* retrieves an object in object list by object ID attribute.
- In the function "**main**", we call out the functions above to get what we need:

    - We got a client object list by calling function "read_client" with client_data.csv as an argument.
    - We got a product object list by calling function "read_product" with product_data.csv as an argument.
    - We turned order_data.csv and monthly_target.csv into pandas data frames by calling function "read_file" with order_data.csv and monthly_target.csv as an argument respectively.
    - We got report object by calling Report class with the client object list, product object list, pandas order data, pandas monthly_target as arguments.
    - We used for loops and method total_overturn() for calculating each object (client or product) to assign the total revenue for each object.
    - When Everything were ready, we proceeded creating salesgui object from SalesManagementGui with all the arguments that we have.

    The programme is ready to go.

4. Things that went well

The datasets were real from my previous job in Vietnam. Therefore, what I needed to do was reprocessing them to be compatible with my function.

The coding structure was handled efficiently; I used the OOP approach for the project, so when the code line grew significantly, the programme still worked well, and the bugs were limited.

Pandas helped data reading, processing, and exporting processes become easier with simple code.

Plotting was handled well, with the foundation from Learning Modules in the COSC480 course.

Tkinter facilitates user experience with interactive widgets such as text, labels, and buttons.

5. The challenge that I exposed to:

The biggest challenge that I was exposed to is from building Tkinter GUI. Some functions worked not so well and were error-prone when the structure of the front end became more complicated. I solved the problems by simplifying the GUI structure to limit errors.

6. Future development

The more I put myself into this project, the more ideas popped into my mind. It would be beautiful to have more time to refine and update current and new functions and make the programme more sophisticated. Some potential features are: making the programme flexible with any business, any location; exporting the result into CSV files; creating functions with filter clients who achieve specific promotion programs from a company, etc.