

Python project- Describing Use and Design of Program

1. Short introduction about the project

This project is mainly focus on coding for ABC classification (80/20 rule) and apply it for analysing specific inventory data of raw materials of a FMCG company. The ABC analysis provide a mechanism for identifying items that will have a significant impact on overall inventory cost. Based on that, the organization will apply appropriate order patterns to avoid high inventory cost for the warehouse. This model can help organization to group the inventory into three categories (A, B and C) in order of their importance levels, then build a proper management model for each category. For example, 'A' items consume high value of organization (let's say: 20% of the number of items will account for 80% of the inventory value), therefore a frequent value analysis and a frequent order pattern (e.g. 'just-in-time' or 1-week order) are required for these kind of items (Wikipedia).

2. Tools and libraries imported in my program

I used Pandas, Numpy, Matplotlib and Tkinter in my program.

It's also need to install Pandas through Command Prompt before running the Program by script "pip install pandas"

3. Guidance for running program "Stock classification"

3.1 Open the program

If you open the program by Spyder (Anaconda), it's fine. If you open the program by Wing IDE 101 and click green button but nothing pop out, you need to click the "Lady bug" button to start the program rather than Run button. All functions in program will perform the same with Run function.

Reason: As I investigated, there's a clash between pandas and tkinter in Wing.

3.2 Data source:

The source of data is in file "Data_stock.csv". Make sure you save this source file the same folder with the Python Program file.

3.3 Other steps are follow the flow of GUI tkinter window:

+ Step 1: Click the button "See my stock status" to view some descriptive statistics of the stock data like: count the number of items, sum of total stock value, mean stock value (the average stock value), max stock (the value of the most expensive item), min stock (the value of the cheapest item).

If the user have new items, they can update the list of these items in "Data_stock.csv", then these descriptive statistics parameters will change.

+ Step 2: For ABC classification: require user to input the threshold of each class.

For example: class A accounts for 70% of stock value (please input: 0.7), class B accounts for 25% of stock value (input: 0.25). Class C is extrapolated by $(1 - 0.7 - 0.25) = 0.05 \sim 5\%$ of stock value

Then click the button "Stock classification" to see the results after classifying stock into 3 classes.

Error handling for user's input: If the user input threshold which is larger than 1, the ERROR will be appeared in red colour ("ERROR: Threshold value must less than 1!")

If the user input threshold class A and threshold class B smaller than 1 but the sum of these threshold is greater than 1, the ERROR will be appeared in red colour ("ERROR: Sum of threshold values for class A, B, C must be equal to 1!").

+ **Step 3:** If the user want to export data to csv file, click the button "Export ABC classification file".

One csv file named "export_data_frame.csv" will be appeared in the same folder of the Program file.

Many new columns added in this file, including the column named "ABC_classification". The warehouse manager can based on this file to know which item is particularly classified to which class and have the proper purchase policy of each item.

+ **Step 4:** click the button "View Pareto chart" to view the Pareto chart of ABC classification.

4. Development process

4.1 How this is done?

My code followed the Object Oriented Programming paradigm. I broke down my program into 3 classes:

Class "Descriptive_stats" which represents the original stock file. The attribute of this class is the file name. In this class, I have a method to calculate the descriptive statistics for the data in the file name.

Class "ABC_classification" which represents the file with classification information in there. The attributes of this class are: the file name and the threshold for each class. This will be a good input for starting to classify stock data. In this class, I have 5 methods:

+ Method "ABC_classification" to build the rule of ABC classification: classified stock based on thresholds.

+ Method "new_dataframe": call the method "ABC_classification" and assign each item in the dataset to class A, B, C. In this step, there are 4 supplemental steps:

Step 1: Sort the data by stock value using built-in function `sort_values()` in pandas

Step 2: Make a cumulative stock value using built-in function `cumsum()`

Step 3: Convert a cumulative stock value into a percentage of grand totals.

Step 4: Divide the inventory into ABC classes by calling the method "self.ABC_classification".

Then, create a new data frame including new columns which were created in step 2, 3, 4.

+ Method "results": call the method "new_dataframe" to get the needed data, then calculate the total stock value for each class using "for" loop.

+ Method "export_results" to export dataframe after ABC classification to file csv output.

+ Method "__str__": returns the formatted string represents of the ABC_classification object. In this step, we call the method "results" and method "new_dataframe" to get the data that we want to extract and pop out in class GUI

Class "StockGui" which use for user interface. In this class, we have the `__init__` function which includes the header frame, main frame, buttons, label, Entry box which needed for the user interface. We have 4 buttons. To bind the events for these 4 buttons, and have 4 functions which call as methods for class StockGui:

- + Method “stock_status”: The event handler for clicks on the button stock_status. The purpose: to see current stock status based on descriptive statistics. This method will call “**Descriptive_stats**” class.
- + Method “ABC_results”: The event handler for clicks on the ABC results. The purpose: to get threshold values from user input and print results after classification. This event handler also handling Exception in case user input wrong. The method will call class “**ABC_classification**”.
- + Method “save_file”: The event handler for clicks on the button “Export ABC classification file”. The output will be a new Excel file with result of ABC classification for each item. This method will call method “export_results” from class “**ABC_classification**”.
- + Method “plot_graph”: The event handler for clicks on the button “View Pareto chart”. The purpose: to plot the Pareto chart. This method will call the method “new_dataframe” to get the data for plotting a chart.

Last but not least, the main function at the end of the program for set up the GUI and run it. We will call “**StockGui**” to trigger the tk window and when we click in those buttons, these buttons will call to relevant classes and methods.

4.2 Issues that I had and the solutions

+ Issue 1: At beginning of the program, I used Pandas for data analysis. At that time, the teacher feedback that my program is static.

➔ Solution: As teacher’s suggestion, I used GUI to get interaction with users about required threshold and use that threshold as an important input for ABC classification.

+ Issue 2: When building GUI for user interface, I came to another issue because I didn’t know how to integrate Pandas with GUI. Questions are: How to get user’s inputs from GUI, then using that values as an input for data analysing using pandas and pop out results from new data frame in GUI? How GUI and Pandas play together?

➔ Solution: lab 11 is my rescuer. Object Oriented Programming is the solution. Considering “Stock file” as a class, we have 2 benefits:

- + Each class have its own attributes. This allows us access different attributes and different iterations of the dataset.
- + Pandas can help us to create new calculated columns for the dataset. Then, OOP can help us to reuse these columns easier since they now exist as a method in the Class.

5. Future development

The versatile of ABC analysis: not only using in inventory management, ABC classification can also apply to many other fields such as: Customer service (find the most valuable customers based on their sale volumes/sale revenue); Shopping (find the most valuable products in grocery store); Quality problems (what are the major quality problems. In each problem, what is the main causes. ABC analysis allows managers to strictly deal with main causes of problems). New features could be added such as: adding different tabs in Tkwindow to process additional 3 applications.