**Kirolos Farouk**

**Omar Mostafa Abdelhady**

**Hashem Zayed**

**Farah ahmed**

**Nour mostafa**

**Final project python**

* We started by importing the libraries that we need then we read the csv file.
* After that we created a class called “data processor” with 2 methods(functions). The first method is called load\_data() it loads the csv file and stores the data in a variable called data then returns data. Second method is called clean\_data() it takes data as a parameter it removes null values then return cleaned data .We create an instance called “Data\_Processor” that calls the class DataProcessor() and then creates another instance that holds the loaded data and then finally an instance called cleaned\_data that carries the cleaned data(without null values).
* Then we created another class called ‘DataAnalyzer’.We created a constructor that holds the cleaned data in a variable called self.cleaned\_data.’The DataAnalyzer’ class has 4 methods: get\_average(), get\_mode(), get\_distribution(), and get\_median().we will start with
  + the get\_average() method , it takes a ‘column’ parameter and calculates the average (mean) of that column from the ‘self.cleaned\_data’ DataFrame; then, It returns the calculated average.
  + The get\_mode() method takes a ‘column’ parameter and calculates the mode (most frequently occurring value) of that column from the ‘self.cleaned\_data’ Data Frame and it returns the calculated mode.
  + The get\_distribution() method takes a ‘column’ as a parameter then calculates the standard deviation of the column from the ‘self.cleaned\_data’ Data Frame after that it returns the calculated standard deviation.
  + The get\_median() method takes a ‘column’ parameter and calculates the median (middle value) of that column from the ‘self.cleaned\_data ‘DataFrame. It returns the calculated median.
* We have created instances for that class in order to call the methods:
  + Analyzer: that calls the DataAnalyzer itself taking the cleaned data as a parameter.
  + Average: that calls the average function in the class and applies it on the “Age” column as an example.
  + mode: that calls the mode function in the class and applies it on the “Age” column as an example.
  + Dist: That calls the distribution function in the class and applies it on the “Rating“ column as an example.
  + Median: that calls the median function in the class and applies it on the “Age” column as an example.
* The last class that we created is the ‘DataVisualizer’ class having the same constructor as the previous class that contains the cleaned data in a variable. ‘DataVisualizer’ class has 4 methods: plot\_line\_chart(), plot\_distribution(), plot\_pie(), and plot\_scatter().
  + The plot\_line\_chart() method takes parameters X, Y, title, and color. It uses seaborn (sns) and matplotlib (plt) libraries to create a line chart based on the specified columns X and Y from the self.cleaned\_data DataFrame. It sets the chart's title, saves the chart as an image file, and displays the chart.
  + The plot\_distribution() method takes parameters column, title, palette, and kde (Kernel Density Estimation) flag. It uses seaborn (sns) and matplotlib (plt) libraries to create a histogram plot of the specified column from the self.cleaned\_data DataFrame. It sets the plot's title, saves the plot as an image file, and displays the plot. The kde parameter determines whether to include a kernel density plot or not.
  + The plot\_pie() method takes parameters column and title. The variable myLabels retrieves the unique values from the specified column in the self.cleaned\_data DataFrame and assigns them to the myLabels variable. These unique values will be used as labels in the pie chart. Then variable values loops in myLabels to count all the unique values. It uses matplotlib (plt) to create a pie chart based on the unique values of the specified column from the self.cleaned\_data DataFrame. It sets the chart's title, saves the chart as an image file with the filename based on the title parameter, and displays the chart.
  + The plot\_scatter() method takes parameters column\_x, column\_y, title, and palette. It uses seaborn (sns) and matplotlib (plt) libraries to create a scatter plot based on the specified column\_x and column\_y from the self.cleaned\_data DataFrame. It sets the plot's title, saves the plot as an image file, and displays the plot.

•As for the class instances:.

* we created the data variable that loads the data
* cleaned\_data variable that calls the cleaned data from the Data\_Processor class,
* Visualizer variable that calls the the class and taking the cleaned data as a parameter
* Plot\_line variable that calls the plot\_line\_chart function from the Visualizer class taking the “Age” and “Positive feedback” as an example and the title of the chart.
* Plot\_Distribution variable that calls the plot Distribution function from the Visualizer class taking the “Age” as an example , the title of the chart and the palette name.
* Scatter\_plot variable that calls the plot\_scatter function from the Visualizer class taking the “Age” and “Positive feedback” as an example and the title of the scatter.
* Plot\_pie variable that calls the plot pie function from the Visualizer class taking the “Rating” , as an example , and the title of the plot as parameters
* **Summary:**

Our code performs data processing, analysis, and visualization tasks on a dataset containing women's clothing e-commerce reviews.

* **GitHub Link:**

https://github.com/Kirolos992/Python-Project

**A picture containing screenshot, colorfulness, design

Description automatically generatedA picture containing screenshot, design

Description automatically generatedA red line on a white background

Description automatically generated with low confidenceVisualization**

**A picture containing colorfulness, circle, graphics

Description automatically generated**