

**eT3 - Tomorrow Information Technology**

**Data Science Task**

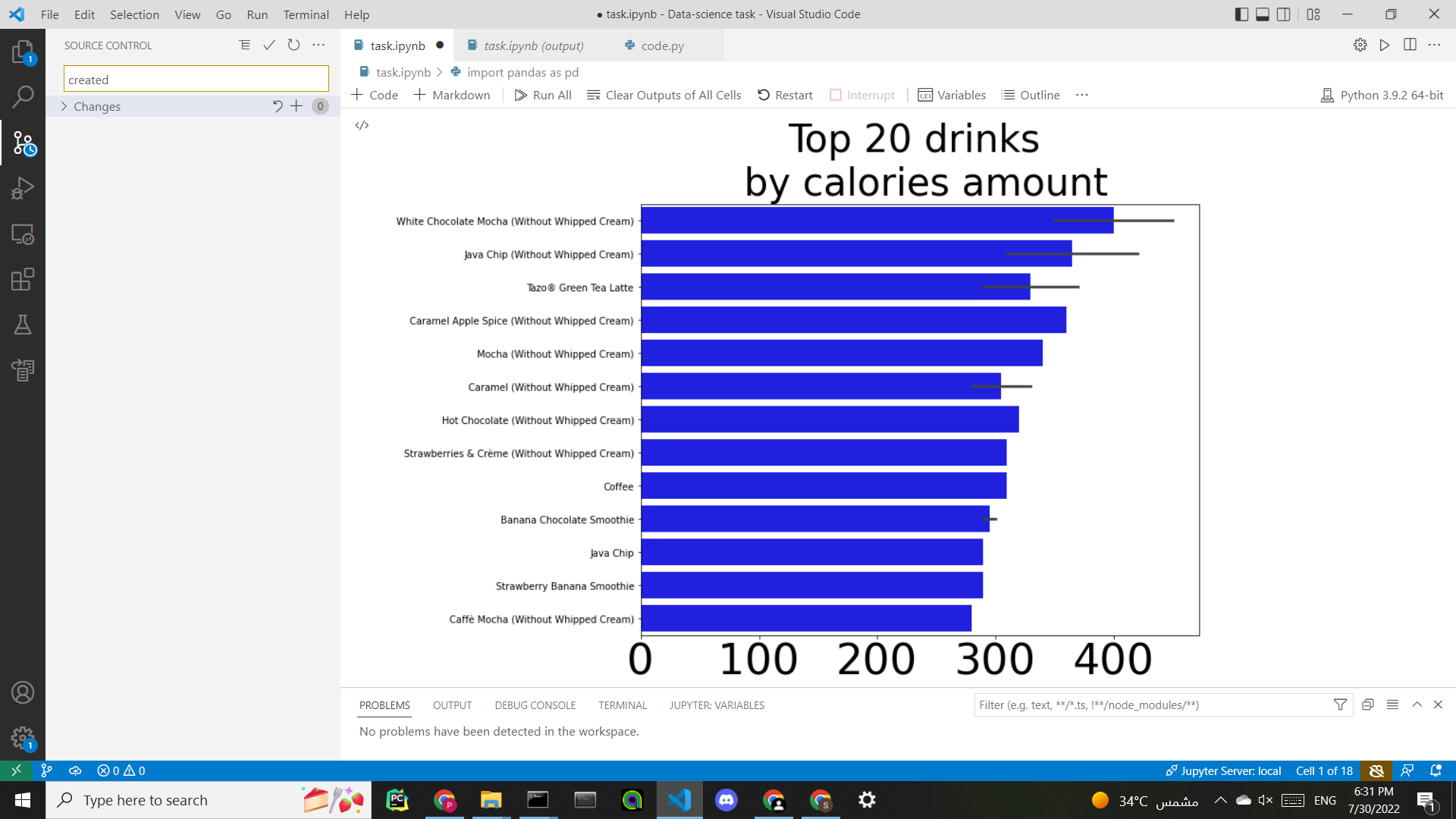
I implemented the task using Python with Jupyter Notebook and the help of the libraries mentioned in the readme file on the repo.

* I imported the dataset into a dataframe using pandas ‘read\_csv() function’.
  + - I checked the different **column types** and separated **numerical columns** from **categorical columns** to be able to process and clean them.
    - I checked the **unique** **values** using ‘unique() ‘ for each column to point out invalid values and converted them to nan.
    - Some columns had **percentages** which I converted to numerical after striping the symbol "%".
    - **Regarding duplicates**, I dropped duplicates using this function data.drop\_duplicates(subset=categorical\_cols, keep='first', inplace=False, ignore\_index=False) and the criteria for a duplicate was assumed to be a drink that has the same beverage name, beverage category and beverage prep.
    - I plotted the data using a box plot to be able to check for **outliers**, but since removing outliers was not mentioned explicitly in the task description, I kept the outliers, you can find all plots in the notebook and here is an example in the figure below for the outliers in column ‘Protein (g)’ .

Graphical user interface, text, application

Description automatically generated

* + I check for na and null values to fill them, I only found na values in the numerical data and filled them with the mean value, for example in column Caffiene (mg), I calculated the mean of all non-empty values and filled the empty values with the calculated mean.
    - I dropped all columns except for, Beverage, Calories and Sugars (g)
    - I sorted the data based on Calories and created a bar plot for the highest 20 drinks in calories and then did the same for Sugars.
    - I found that the highest in calories was **White Chocolate Mocha (Without Whipped Cream)**

****

* + - And the highest in Sugars was **Java Chip (Without Whipped Cream)**

**Chart

Description automatically generated**