gn417126 - Christian Gang Liu - B000415613

**Assignment 4**

on

**CSCI 6610 Visual Analytics**

Submitted by

**Christian Gang Liu (B000415613/gn417126@dal.ca)**

**…………………………**

In partial fulfillment of the requirements for the Course CSCI 6610 Human Computer Interaction

**Computer Science**

**2019-11-09**

# DEPARTMENT OF Computer Science

**Present to Teaching Assistant : Mateus Pereira; Leonardo Milhomem Franco Christino**

**&**

**Professor : Fernando Paulovich, PhD**



# DEPARTMENT OF Computer Science

**Important Notes: because I am principle programmer of our “outliers” group project implementation, so you might find similar coding between my assignment 4 and project.**

1. **Project technical used:**

General speaking, the project is based on JS and Python 3.4+ as required by assignment 4 instruction.

1, Flask: A python framework used for quick REST services establishment;

2, D3: A Javascript library used for generating the interactive ML graphic charts;

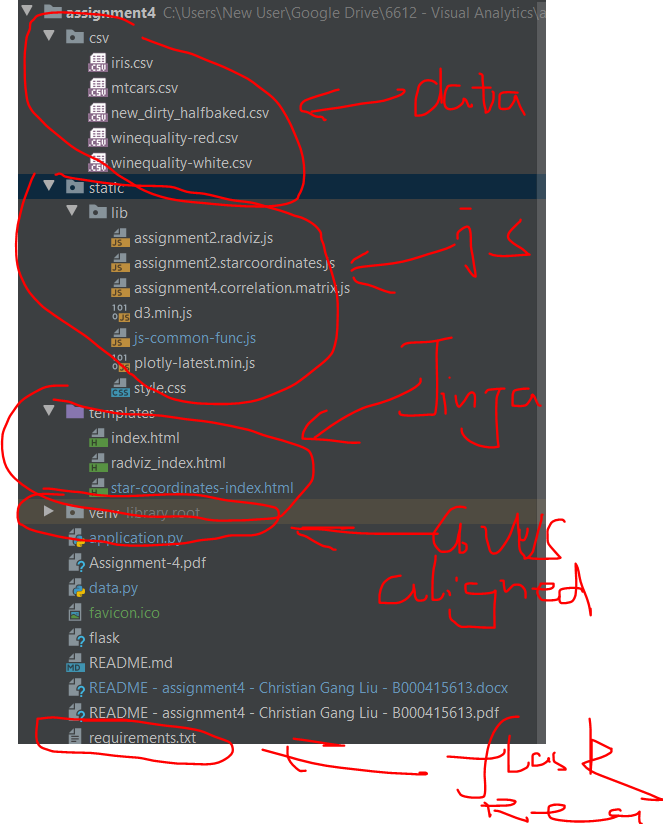
3, Plotly: A popular javascript library which based on D3 provides the complicated chart view (here I only used it for show the tabular table view on the page, this requirement is not asked by assignment, I used it only to demonstrate how data comes from backend framework (Flask)

4, Others: JQuery / Bootstrap (only for UI layout and event triggering)

5, IDE: PyCharm

6, AWS (extra work): public cloud deployment

1. **Project structure:**



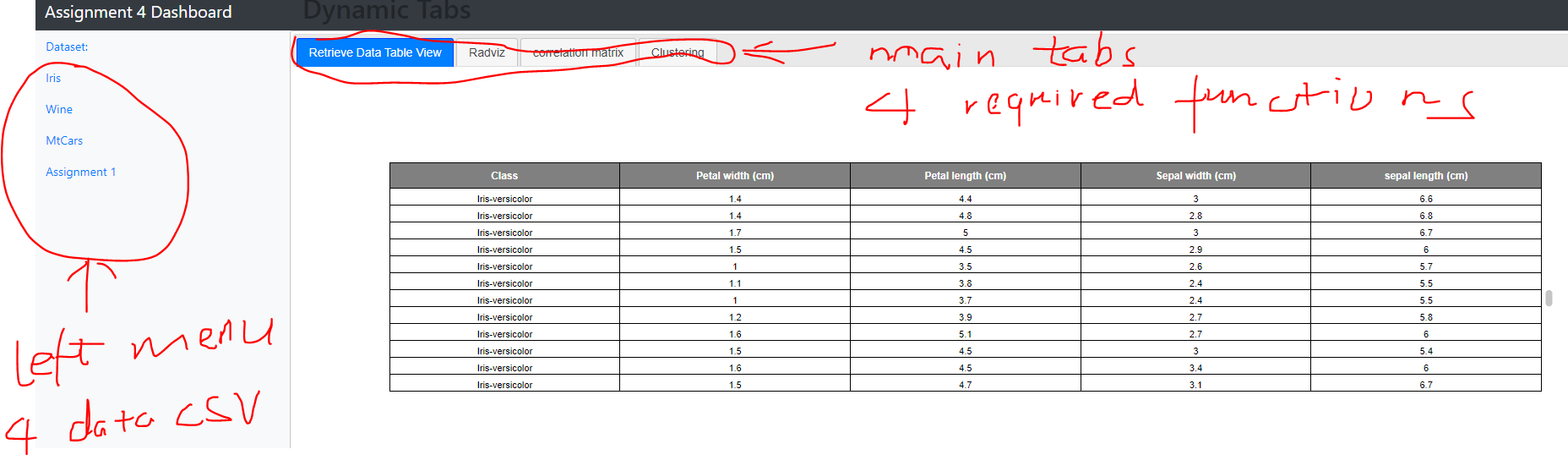
As you can see above, I constructed project structure to comply with the AWS elastic sever standardization, in order to deploy on AWS for public access.

I utilized FLASK framework to quickly establish this lightweight project, like what I did for my group project as well (<http://flask-env.zsewpnnzda.us-east-2.elasticbeanstalk.com/>).

In order to simplify the assignment 4 review work. I planned to deploy it on AWS for quickly reviewing. Anyhow you can always open my project source folder to execute command: python application.py to exam it on local.

1. **General UI introduction (the details will be described specifically to each of questions of assignment 4):**

UI layout introduction:



1. Left Menu: they are four different tables which are required by assignment 4:
2. Iris
3. Wine
4. Mtcars
5. Assignment 1 dataset (bonus)
6. Main Menu:
7. Tabular view on Html page (extra work): it shows tabular view directly from JSON response of Flask REST services:
8. Radviz: The assignment 2 functionality for each of tables;
9. Correlation Matrix: Heatmap of each table;
10. Clustering: there will be a comparison between original labels (classification) and clustering by K-means, as well as a measurement score showing on the page.

Notes: the Bonus requirement of assignment 1 dataset will share the same functionalities.

1. **The specific answers to the assignment 4 questions:**

# [100 (+30)] Requirements:

Code will be marked based on functionality, structure, reusability, best practices, and documentation.

1. [10 Marks] Create a backend that will provide the data and metadata that can be used to display the visualization
   1. Use an HTTP request to retrieve data from the back-end and use it to generate the visualization on the front-end.
   2. Tip: You can return, along with the data, some metadata like column names or other information that could be useful to handle and/or display the data in the front-end.
2. [20 Marks] Add an option on the interface to choose a different dataset (iris or winequality)
   1. The backend will return the new dataset
   2. You can keep a state on the front-end and send it on every request to identify the current dataset in use.
3. [20 Marks] When hovering an instance of a given cluster, show (as a tooltip or in other available space) the correlation matrix for instances of that cluster.
   1. The correlation matrix should be calculated and returned by the back-end.
   2. This should be displayed on the front-end through a color matrix. See example of such matrix below (Tip: feel free to use libraries to help you):



1. [40 Marks] Implement a button that requests the backend to clusterize the data using one of:

K-Means or DBScan

* 1. You should color the instances using the clustering information
  2. Add a switch button to choose between the color modes: cluster colors or class-based colors.
  3. The clusterization should be performed on the same dataset currently seem in the visualization.
  4. You just need to implement for one clusterization algorithm.
  5. You may use existing implementations of the clustering algorithms.

1. [10 Marks] Add one (or more) options to configure the parameters of the clustering algorithm
   1. Clicking the button should make a new clusterization with the new parameters and update the colors on the visualization.
2. [+30 Bonus Marks] Add an option on the interface to choose to see the preprocessed dataset generated by your A1 assignment.
   1. RadViz/StarCoordinates should only show the numerical columns as anchor points
   2. The Categorical columns should be shown as the color of the plot. Make an input box selector in the interface to choose the categorical column to be shown as the color.