Assignment 4
on
CSCI 6610 Visual Analytics

Christian Gang Liu (B000415613)

November 29, 2019

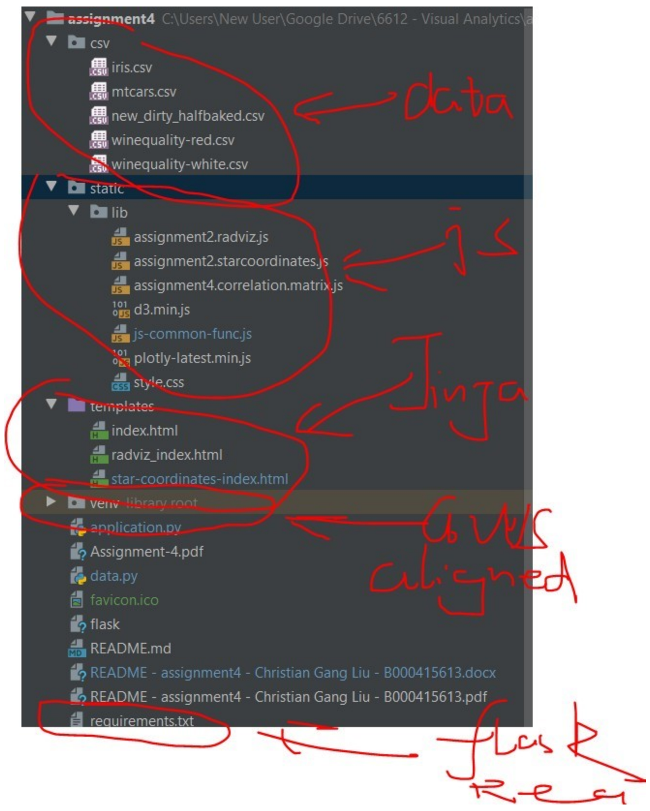
Abstract

Important Notes: because I am principle programmer of our "outliers" group project implementation, so you might find similar coding on our project as well.

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

   - Flask: A python framework used for quick REST services establishment;
   - D3: A Javascript library used for generating the interactive ML graphic charts;
   - 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);
   - Others: JQuery / Bootstrap (only for UI layout and event triggering)
   - IDE: PyCharm;
   - AWS (extra work): public cloud deployment:$http://assignment4-dev.us-east-2.elasticbeanstalk.com/$

2. Project structure:



```
▼ assignment4 C:\Users\New User\Google Drive\6612 - Visual Analytics\a
  ▼ csv
      iris.csv
      mtcars.csv
      new_dirty_halfbaked.csv
      winequality-red.csv
      winequality-white.csv
  ▼ static
    ▼ lib
        assignment2.radviz.js
        assignment2.starcoordinates.js
        assignment4.correlation.matrix.js
        d3.min.js
        js-common-func.js
        plotly-latest.min.js
        style.css
  ▼ templates
      index.html
      radviz_index.html
      star-coordinates-index.html
  ▶ venv library root
    application.py
    Assignment-4.pdf
    data.py
    favicon.ico
    flask
    README.md
    README - assignment4 - Christian Gang Liu - B000415613.docx
    README - assignment4 - Christian Gang Liu - B000415613.pdf
    requirements.txt
```

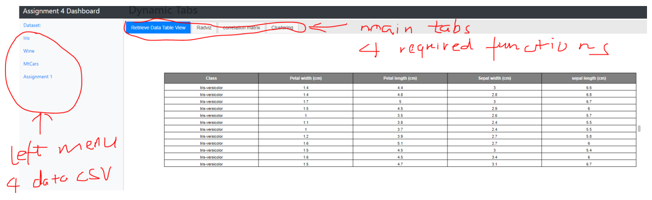
As you can see above, I constructed project structure to comply with the AWS elastic sever standardization, in order to deploy on AWS for public access.

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.

3. Execution: the execution instruction is also mentioned in the README file under project folder.

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



Class	Petal width (cm)	Petal length (cm)	Sepal width (cm)	sepal length (cm)
Iris-versicolor	1.4	4.4	3	6.6
Iris-versicolor	1.4	4.8	2.8	6.8
Iris-versicolor	1.7	5	3	6.7
Iris-versicolor	1.5	4.5	2.9	6
Iris-versicolor	1	3.5	2.6	5.7
Iris-versicolor	1.1	3.8	2.4	5.5
Iris-versicolor	1	3.7	2.4	5.5
Iris-versicolor	1.2	3.9	2.7	5.8
Iris-versicolor	1.6	5.1	2.7	6
Iris-versicolor	1.5	4.5	3	5.4
Iris-versicolor	1.6	4.5	3.4	6
Iris-versicolor	1.5	4.7	3.1	6.7

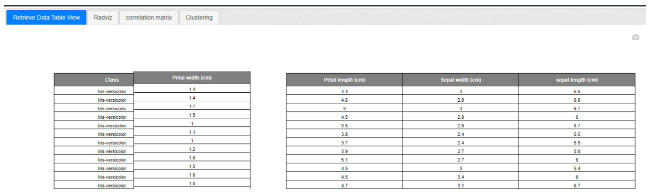
- Left Menu: they are four different tables which are required by assignment 4:
  - Iris
  - Wine
  - Mtcars
  - Assignment 1 dataset (bonus)
- Main Menu:
  - Tabular view on Html page (extra work): it shows tabular view directly from JSON response of Flask REST services:
  - Radviz: The assignment 2 functionality for each of tables;
  - Correlation Matrix: Heatmap of each table;
  - 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.

0.1 Assignment 4 Qeustions And Answers

The specific answers to the assignment 4 questions:

1 *[10 Marks] Create a backend that will provide the data and metadata that can be used to display the visualization a. Use an HTTP request to retrieve data from the back-end and use it to generate the visualization on the front-end. b. 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.*

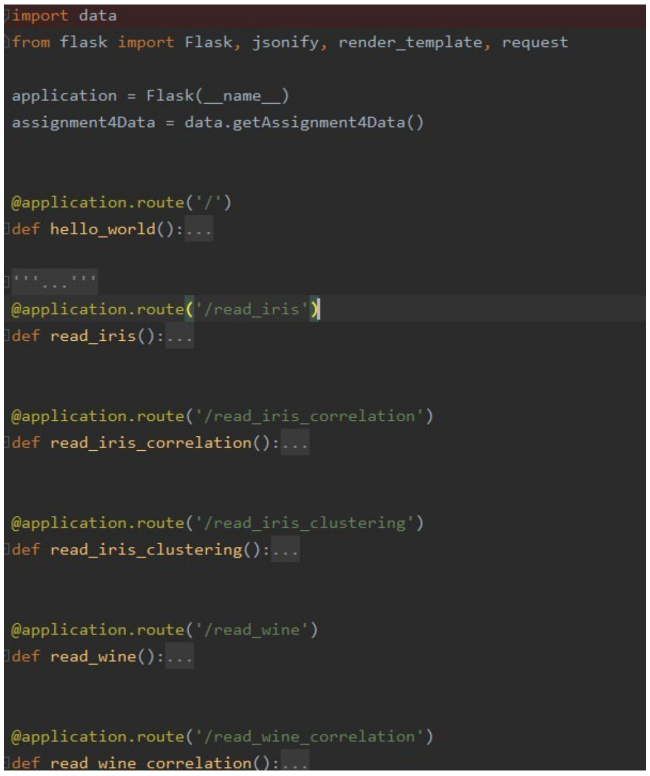
answer: Front-end (JS retrievals JSON data, sends to html page):



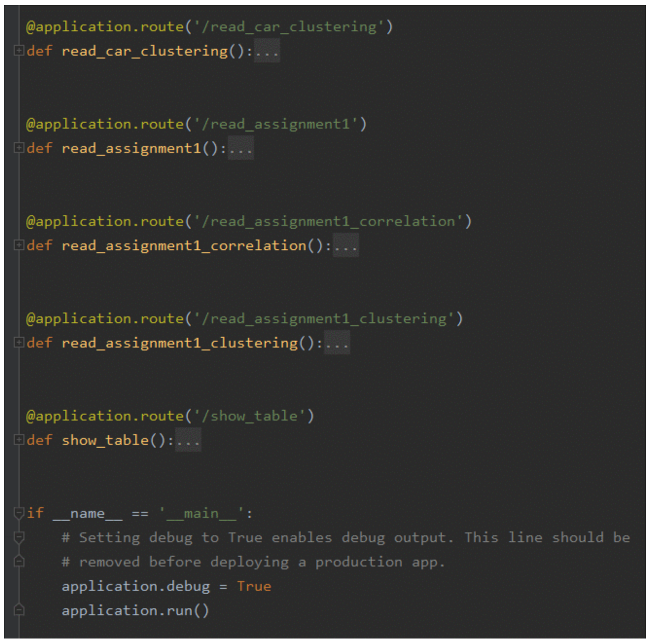
Class	Petal width (cm)
Iris-versicolor	1.4
Iris-versicolor	1.4
Iris-versicolor	1.7
Iris-versicolor	1.5
Iris-versicolor	1
Iris-versicolor	1.1
Iris-versicolor	1
Iris-versicolor	1.2
Iris-versicolor	1.6
Iris-versicolor	1.5
Iris-versicolor	1.6
Iris-versicolor	1.5

Petal length (cm)	Sepal width (cm)	sepal length (cm)
4.4	3	6.6
4.8	2.8	6.8
5	3	6.7
4.5	2.9	6
3.5	2.6	5.7
3.8	2.4	5.5
3.7	2.4	5.5
3.9	2.7	5.8
5.1	2.7	6
4.5	3	5.4
4.5	3.4	6
4.7	3.1	6.7

Back-end Flask WSGI endpoint (REST services) – Code snippets:

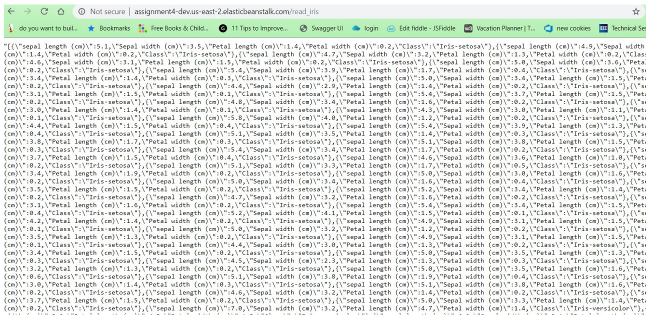


```
import data
from flask import Flask, jsonify, render_template, request

application = Flask(__name__)
assignment4Data = data.getAssignment4Data()


@application.route('/')
def hello_world():...

'''...'''
@application.route('/read_iris')
def read_iris():...


@application.route('/read_iris_correlation')
def read_iris_correlation():...


@application.route('/read_iris_clustering')
def read_iris_clustering():...


@application.route('/read_wine')
def read_wine():...


@application.route('/read_wine_correlation')
def read_wine_correlation():...
```



```python
@application.route('/read_car_clustering')
def read_car_clustering():...


@application.route('/read_assignment1')
def read_assignment1():...


@application.route('/read_assignment1_correlation')
def read_assignment1_correlation():...


@application.route('/read_assignment1_clustering')
def read_assignment1_clustering():...


@application.route('/show_table')
def show_table():...


if __name__ == '__main__':
    # Setting debug to True enables debug output. This line should be
    # removed before deploying a production app.
    application.debug = True
    application.run()
```

The returning JSON sample like this: $(http://assignment4-dev.us-east-2.elasticbeanstalk.com/read\_iris)$



```
"[{\"sepal length (cm)\":5.1,\"Sepal width (cm)\":3.5,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.9,\"Sepal width
(cm)\":1.4,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.7,\"Sepal width (cm)\":3.2,\"Petal length (cm)\":1.3,\"Petal width (cm)\":0.2,\"Clas
(cm)\":4.6,\"Sepal width (cm)\":3.1,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.6,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.4,\"Sepal width (cm)\":3.9,\"Petal length (cm)\":1.7,\"Petal width (cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.4,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.4,\"Petal length (cm)\":1.5,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.4,\"Sepal width (cm)\":2.9,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.1,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.1,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.4,\"Sepal width (cm)\":3.7,\"Petal length (cm)\":1.5,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.8,\"Sepal width (cm)\":3.4,\"Petal length (cm)\":1.6,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.0,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.1,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.3,\"Sepal width (cm)\":3.0,\"Petal length (cm)\":1.1,\"Peta
(cm)\":0.1,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.8,\"Sepal width (cm)\":4.0,\"Petal length (cm)\":1.2,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":4.4,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.4,\"Sepal width (cm)\":3.9,\"Petal length (cm)\":1.3,\"Peta
(cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.1,\"Sepal width (cm)\":3.5,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.8,\"Petal length (cm)\":1.7,\"Petal width (cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.1,\"Sepal width (cm)\":3.8,\"Petal length (cm)\":1.5,\"Peta
(cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.4,\"Sepal width (cm)\":3.4,\"Petal length (cm)\":1.7,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.7,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.6,\"Sepal width (cm)\":3.6,\"Petal length (cm)\":1.0,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.1,\"Sepal width (cm)\":3.3,\"Petal length (cm)\":1.7,\"Petal width (cm)\":0.5,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.4,\"Petal length (cm)\":1.9,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.0,\"Petal length (cm)\":1.6,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.4,\"Petal length (cm)\":1.6,\"Petal width (cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.5,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.2,\"Sepal width (cm)\":3.4,\"Petal length (cm)\":1.4,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.7,\"Sepal width (cm)\":3.2,\"Petal length (cm)\":1.6,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.1,\"Petal length (cm)\":1.6,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.4,\"Sepal width (cm)\":3.4,\"Petal length (cm)\":1.5,\"Peta
(cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.2,\"Sepal width (cm)\":4.1,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.1,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":4.2,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.9,\"Sepal width (cm)\":3.1,\"Petal length (cm)\":1.5,\"Peta
(cm)\":0.1,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.2,\"Petal length (cm)\":1.2,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.5,\"Petal length (cm)\":1.3,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.9,\"Sepal width (cm)\":3.1,\"Petal length (cm)\":1.5,\"Peta
(cm)\":0.1,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.4,\"Sepal width (cm)\":3.0,\"Petal length (cm)\":1.3,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.4,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.5,\"Petal length (cm)\":1.3,\"Peta
(cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.5,\"Sepal width (cm)\":2.3,\"Petal length (cm)\":1.3,\"Petal width (cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.2,\"Petal length (cm)\":1.3,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.5,\"Petal length (cm)\":1.6,\"Peta
(cm)\":0.6,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.1,\"Sepal width (cm)\":3.8,\"Petal length (cm)\":1.9,\"Petal width (cm)\":0.4,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.0,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.3,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.1,\"Sepal width (cm)\":3.8,\"Petal length (cm)\":1.6,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":4.6,\"Sepal width (cm)\":3.2,\"Petal length (cm)\":1.4,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"se
(cm)\":3.7,\"Petal length (cm)\":1.5,\"Petal width (cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":5.0,\"Sepal width (cm)\":3.3,\"Petal length (cm)\":1.4,\"Peta
(cm)\":0.2,\"Class\":\"Iris-setosa\"},{\"sepal length (cm)\":7.0,\"Sepal width (cm)\":3.2,\"Petal length (cm)\":4.7,\"Petal width (cm)\":1.4,\"Class\":\"Iris-versicolor\"},
```

The list of REST services showing below (hostname is http://assignment4-dev.us-east- 2.elasticbeanstalk.com):

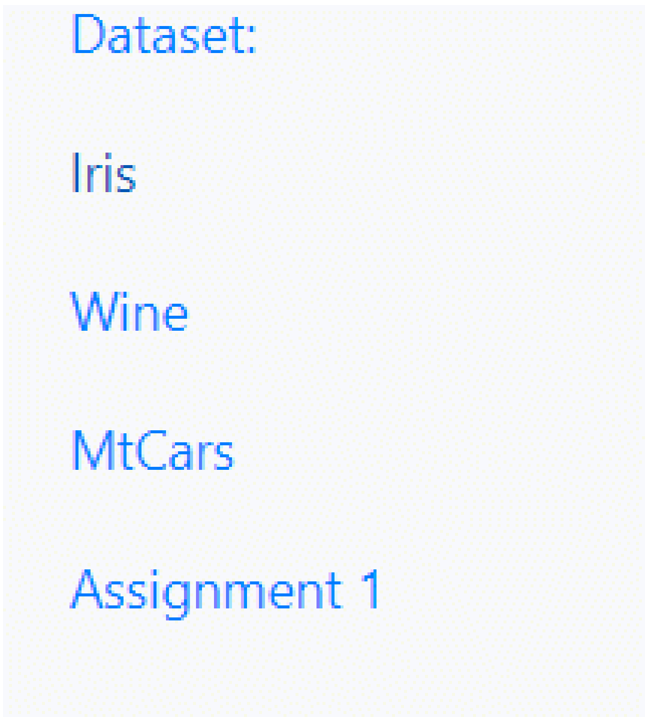
- application.route('/')
- application.route('/read_iris')
- application.route('/read_iris_correlation')
- application.route('/read_iris_clustering')
- application.route('/read_wine')
- application.route('/read_wine_correlation')

- application.route('/read_wine_clustering')
- application.route('/read_car')
- application.route('/read_car_correlation')
- application.route('/read_car_clustering')
- application.route('/read_assignment1')
- application.route('/read_assignment1_correlation')
- application.route('/read_assignment1_clustering')
- application.route('/show_table')

2 *[20 Marks] Add an option on the interface to choose a different dataset (iris or winequality) a. The backend will return the new dataset*

Answer:

Front-end: when you click any option in left panel, the back-end will return corresponding response:



Back-end returning:

- IRIS: http://assignment4-dev.us-east-2.elasticbeanstalk.com/read_iris
- WINE: http://assignment4-dev.us-east-2.elasticbeanstalk.com/read_wine
- MTCARS: http://assignment4-dev.us-east-2.elasticbeanstalk.com/read_car

- ASSIGNMENT1: http://assignment4-dev.us-east-2.elasticbeanstalk.com/read_assignment1

b. You can keep a state on the front-end and send it on every request to identify the current dataset in use.

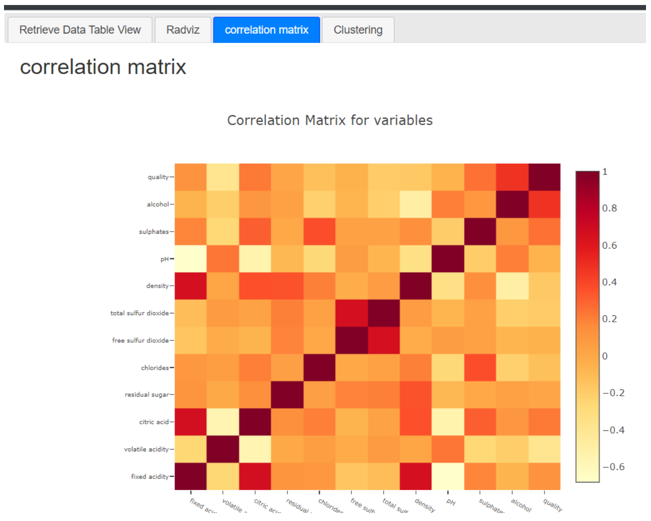
I am using the request parameter $/?data = [dataset\_name]$ to identify current dataset using:

Like $/?data = [iris\|wine\|assignment1\|car]$

3 *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. a. The correlation matrix should be calculated and returned by the back-end.*

Answer:

Front-end:



Back-end:

```
@application.route('/read_wine_correlation')
def read_wine_correlation():
return jsonify(data.read_wine_correlation())
```

So whenever you click "correlation matrix" tab, it will send REST call like(giving an example):

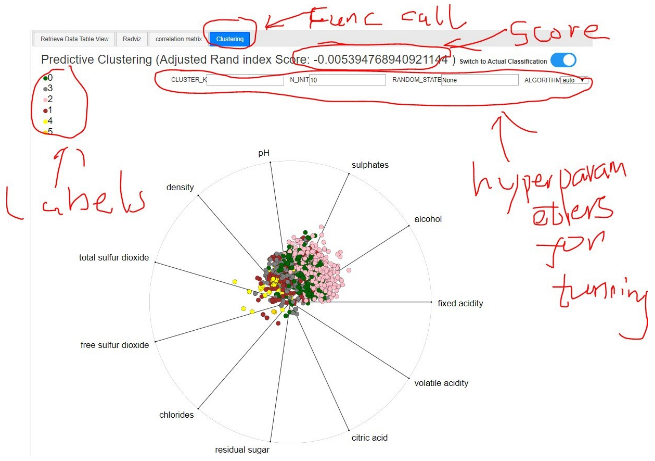
```
def read_wine_correlation()
-> readAssignment4Data:
rawData = getAssignment4Data().wine correlation_result = rawData.corr()
return correlation_result.to_json(orient='records')
```

b. 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):

So I utilize the corr() function of pandas to generate correlation heatmap

4 *[40 Marks] Implement a button that requests the backend to clusterize the data using one of: K-Means or DBScan a. You should color the instances using the clustering information*

Answer:



Retrieve Data Table View | Radviz | correlation matrix | Clustering

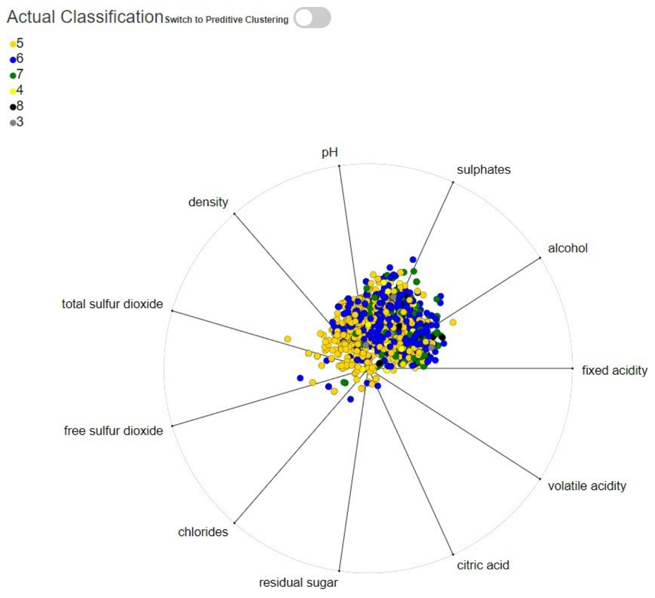
Predictive Clustering (Adjusted Rand index Score: -0.005394768940921144) Switch to Actual Classification

●0
●3
●2
●1
●4
●5

CLUSTER_K [ ] N_INIT [10] RANDOM_STATE [None] ALGORITHM [auto ▾]

b. Add a switch button to choose between the color modes: cluster colors or class-based colors.

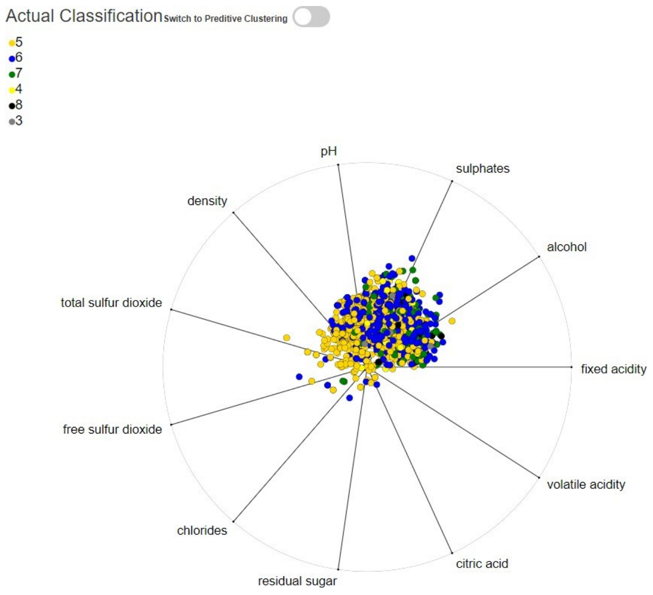
Answer:



Actual Classification Switch to Preditive Clustering

●5
●6
●7
●4
●8
●3

After we switch back to actual classification:



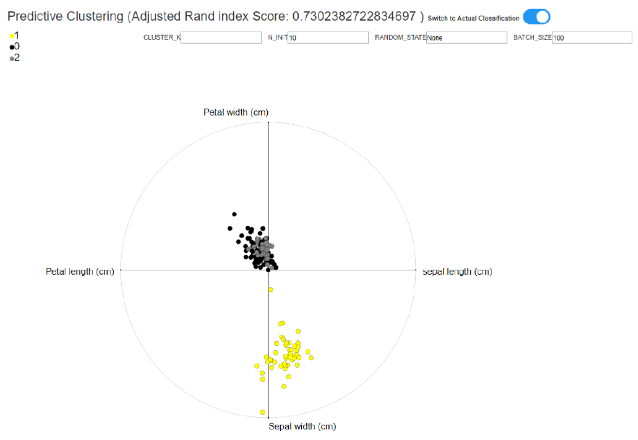
c. The clusterization should be performed on the same dataset currently seem in the visualization.

Answer:

Since /?$data$ = [$dataset\_name$] remembers the current dataset. Clustering tab will go further to call

- @application.route('/read_iris_clustering')
- @application.route('/read_wine_clustering')
- @application.route('/read_car_clustering')
- @application.route('/read_assignment1_clustering')

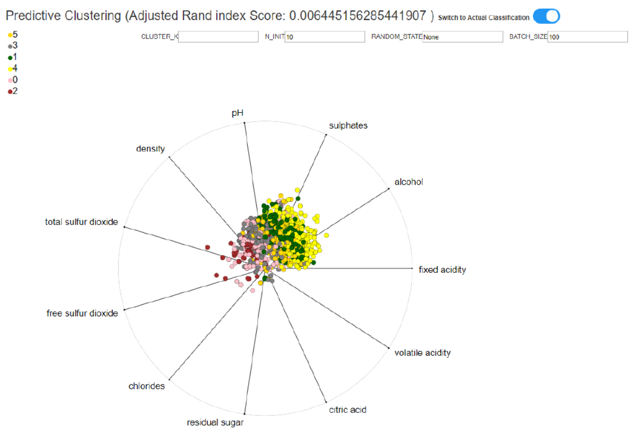
Page will show corresponding result to current dataset:



Predictive Clustering (Adjusted Rand index Score: 0.7302382722834697) Switch to Actual Classification

●1
●0
●2

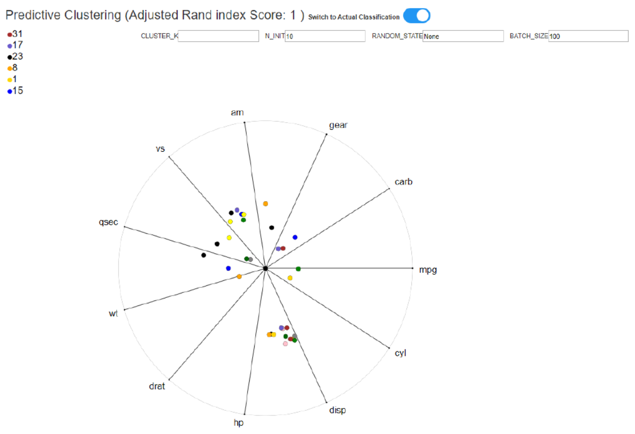
CLUSTER_K N_INIT 10 RANDOM_STATE None BATCH_SIZE 100



Predictive Clustering (Adjusted Rand index Score: 0.006445156285441907) Switch to Actual Classification

●5
●3
●1
●4
●0
●2

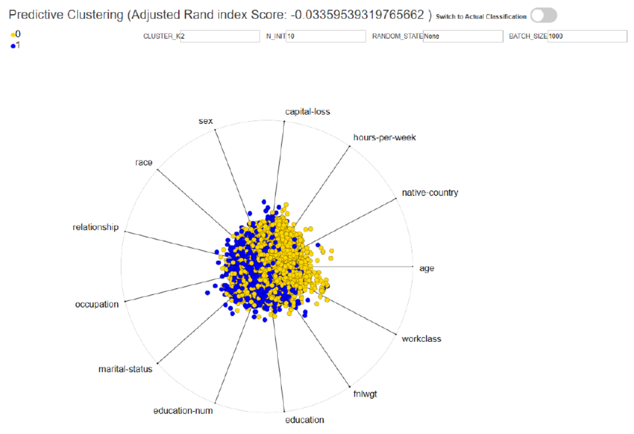
CLUSTER_K N_INIT 10 RANDOM_STATE None BATCH_SIZE 100



Predictive Clustering (Adjusted Rand index Score: 1) Switch to Actual Classification

●31
●17
●23
●8
●1
●15

CLUSTER_K N_INIT 10 RANDOM_STATE None BATCH_SIZE 100


Predictive Clustering (Adjusted Rand index Score: -0.03359539319765662) Switch to Actual Classification

●0
●1

CLUSTER_K 2 N_INIT 10 RANDOM_STATE None BATCH_SIZE 1000

d. You just need to implement for one clusterization algorithm.

Answer:

K-MEAN

e. You may use existing implementations of the clustering algorithms.

Answer:

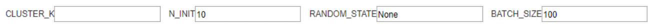
SKLearn.cluster.kmean

5 *[10 Marks] Add one (or more) options to configure the parameters of the clustering algorithm a. Clicking the button should make a new clusterization with the new parameters and update the colors on the visualization.*

Answer:

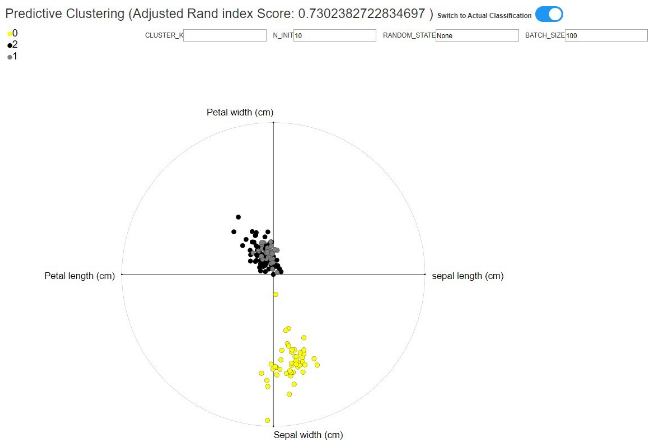
Here I am going to tune up four major different parameters (Quoted from https://scikit- learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html)

- n_clusters : int, optional, default: 8 The number of clusters to form as well as the number of centroids to generate.
- batch_size : int, optional, default: 100 Size of the mini batches.
- random_state : int, RandomState instance or None (default) Determines random number generation for centroid initialization and random reassignment. Use an int to make the randomness deterministic.
- n_init : int, default=3 Number of random initializations that are tried. In contrast to KMeans, the algorithm is only run once, using the best of the n_init initializations as measured by inertia.

The UI toolbar looks like this 
CLUSTER_K N_INIT 10 RANDOM_STATE None BATCH_SIZE 100

It will allow us to adjust four different major parameters of k-mean , let us see the effects:

Originally we have:



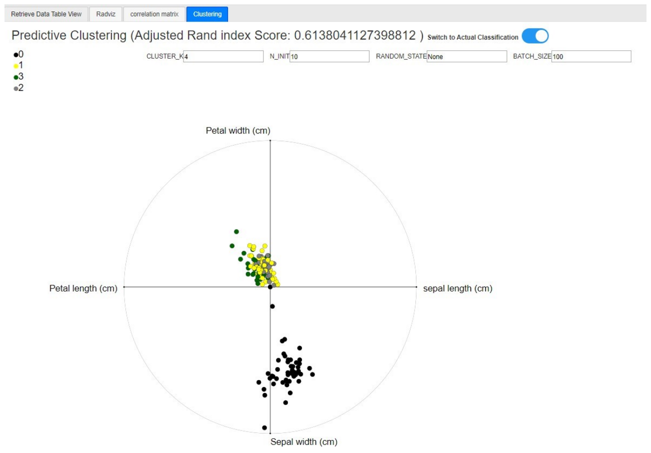
Predictive Clustering (Adjusted Rand index Score: 0.7302382722834697) Switch to Actual Classification

●0
●2
●1

CLUSTER_K N_INIT 10 RANDOM_STATE None BATCH_SIZE 100

let us see how hyperparameters tunning up:

- K-cluster:



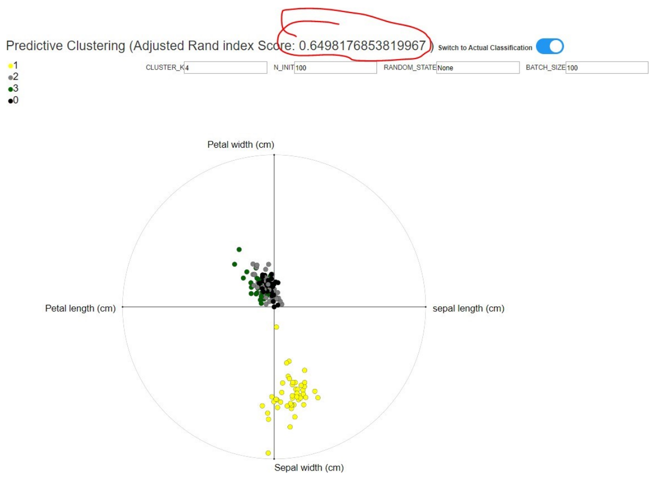
Retrieve Data Table View | Radviz | correlation matrix | Clustering

Predictive Clustering (Adjusted Rand index Score: 0.6138041127398812) Switch to Actual Classification

●0
●1
●3
●2

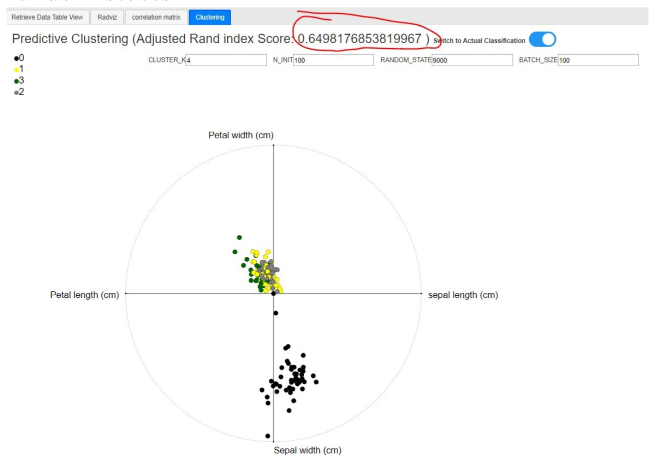
CLUSTER_K 4 N_INIT 10 RANDOM_STATE None BATCH_SIZE 100

- N_init adjustment:


Predictive Clustering (Adjusted Rand index Score: 0.6498176853819967)

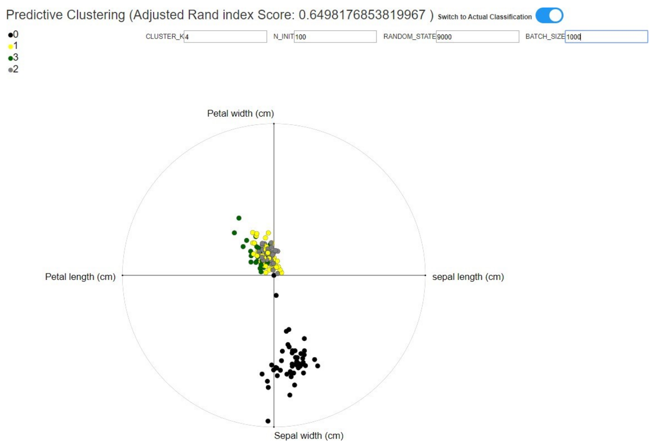
CLUSTER_K 4 N_INIT 100 RANDOM_STATE None BATCH_SIZE 100

- random_states:


Predictive Clustering (Adjusted Rand index Score: 0.6498176853819967)

CLUSTER_K 4 N_INIT 100 RANDOM_STATE 9000 BATCH_SIZE 100

- Batch_size adjustment:


Predictive Clustering (Adjusted Rand index Score: 0.6498176853819967) Switch to Actual Classification

CLUSTER_K 4 N_INIT 100 RANDOM_STATE 9000 BATCH_SIZE 1000

You can see the result will slightly different as we tune the parameters based on the measurement score.

6 *[+30 Bonus Mganr4k1s7] 1A2d6d -anCohpritsiotinanonGthaenginLteirufaceBt0o0c0h4o1o5se61to3see the preprocessed dataset generated by your A1 assignment. a. RadViz/StarCoordinates should only show the numerical columns as anchor points*

Answer:

Using sklearn normalization library to scale and label the columns as digit numbers

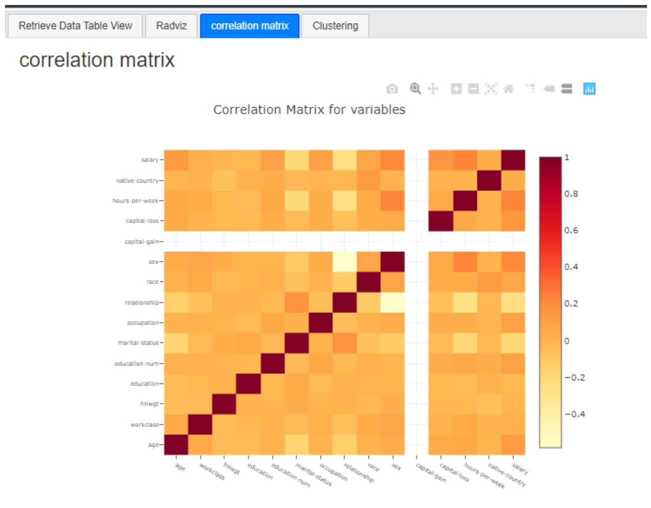
b. The Categorical columns should be shown as the color of the plot. Make an input box selector in the interface to choose the categorical column to be shown as the color.

Answer:

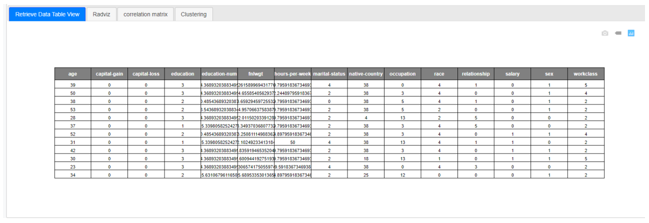
Make the salary as label: ¡50K = 0, ¿50K = 1

Assignment 1 dataset will share same functionalities as three other datasets:

Correlation:

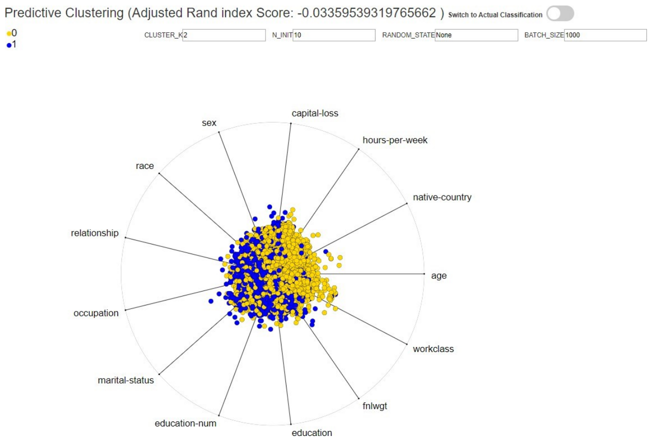


tabular view:



age	capital-gain	capital-loss	education	education-num	fnlwgt	hours-per-week	marital-status	native-country	occupation	race	relationship	salary	sex	workclass
39	0	0	3	4.36893203883495	261589969431776	9.7959183673469	4	38	0	4	1	0	1	5
50	0	0	3	4.36893203883495	4.65585405629372	2.24489795918367	2	38	3	4	0	0	1	4
38	0	0	2	0.48543689320387	3.65929459725532	9.7959183673469	0	38	5	4	1	0	1	2
53	0	0	2	3.54368932038834	4.95706637583879	9.7959183673469	2	38	5	2	0	0	1	2
28	0	0	3	4.36893203883495	2.01150203391289	9.7959183673469	2	4	13	2	5	0	0	2
37	0	0	1	5.33980582524273	3.34937036807732	9.7959183673469	2	38	3	4	5	0	0	2
52	0	0	2	0.48543689320387	3.25081114988362	4.89795918367346	2	38	3	4	0	1	1	4
31	0	0	1	5.33980582524272	2.10249233413184	50	4	38	13	4	1	1	0	2
42	0	0	3	4.36893203883495	835919465352049	9.7959183673469	2	38	3	4	0	1	1	2
30	0	0	3	4.36893203883495	600944192751939	9.7959183673469	2	18	13	1	0	1	1	5
23	0	0	3	4.36893203883495	306574175055974	9.5918367346938	4	38	0	4	3	0	0	2
34	0	0	2	5.63106796116505	5.68953353013654	4.89795918367346	2	25	12	0	0	0	1	2

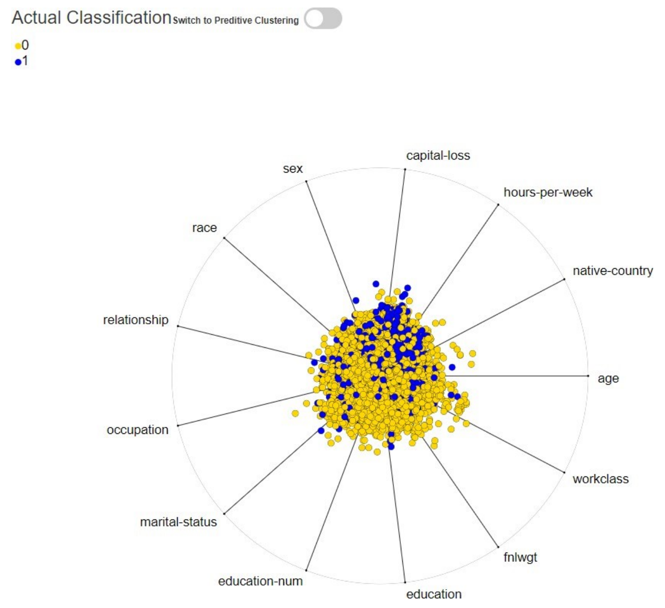
clustering:



Predictive Clustering (Adjusted Rand index Score: -0.03359539319765662)

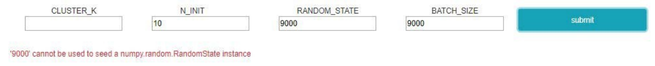
CLUSTER_K 2 N_INIT 10 RANDOM_STATE None BATCH_SIZE 1000

original Redviz:



0.2 Miscellaneous: Error Handling

I added error message box just in the form to show any error messages, whenever clustering has the problem at the back-end. Eg: when user tries to input random_state = 9000, which is not valid for k-means, then front-page will honestly record the error messages returning from back-end, like this:



N_INIT: 10
RANDOM_STATE: 9000
BATCH_SIZE: 9000

'9000' cannot be used to seed a numpy.random.RandomState instance

0.3 Referencing

1 D3: https://d3js.org/

2 Plotly: https://plot.ly/

3 Flask:https://realpython.com/tutorials/flask/

4 Jinja2 Template: https://jinja.palletsprojects.com/en/2.10.x/

5 Jquery: https://api.jquery.com/

6 Bootstrap: https://getbootstrap.com/

7 AWS: https://www.awseducate.com/