Final Project

Definition/Description

In the era of capital-B capital-D Big Data, data analysis and visualization is the best approach for extracting useful information and making decisions.

For your Final Project, you are tasked with creating a web app which manipulates and visualizes a Big Data dataset.

Requirements

General

Your web app shall be written in Python using Flask, pandas, scikit-learn, and matplotlib.

Your web app shall allow clients to choose what subsets of the data they would like to see (via text input, radio buttons, et al.), and the app will serve them a visualization of that data.

Big Data

You may use any dataset of your choice for your Big Data; follow your heart! It must, however, meet the following criteria:

1. It must contain at *least* 1000 datapoints.
2. *At least half* of the data must be numeric.

Your Big Data shall be stored on your server. It may be stored as a csv. For extra credit, your Big Data shall be stored in a database which your web app accesses.

Endpoints

Your web app should implement a number of endpoints.

1. At least 2 endpoints used for GET requests, i.e. directly accessible by browsers (e.g. `'/'`).
    a. At least 1 of these should be a dynamic endpoint created from a client POST request (e.g. for a web app which makes predictions on amounts of insects in a given area, `'/projection/butterflies'`).
2. At least 2 endpoints used for POST requests, i.e. not directly accessible by browsers (e.g. `'/login'` from the Web App Homework).

Scientific Computation

Your web app shall do some sort of computation with the Big Data dataset. This could be as straightforward as aggregating attributes, but *it needs to compute something meaningful*.

Machine Learning

Your web app shall make predictions based on the given Big Data dataset. As long as your web app is using ML to make predictions, you're good to go. These could be:

1. Predictions of how a particular property will change over time.
2. Label classifications of data with unknown labels.

Data Visualization

Your web app shall visualize the Big Data dataset in some meaningful way. At least two types of plots should be accessible:

1. At least 1 plot should visualize the data without any ML processing.
2. At least 1 plot should visualize ML-processed predictions.

These plots do not all have to appear on the same webpage.

Provided Files/Data

Example project

An example of the Final Project from last semester can be found here: http://aldenderfer.pythonanywhere.com/ . This project visualizes COVID-19 confirmed and recovered cases, and also visualizes ML predictions of future confirmed and recovered cases. It uses datasets from The Johns Hopkins University (JHU) Center for Systems Science and Engineering (CSSE), but only uses data up to mid-October.

Reference material

1. Flask documentation: https://flask.palletsprojects.com/en/1.1.x/
2. Jinja documentation: https://jinja.palletsprojects.com/en/2.11.x/
3. matplotlib documentation: https://matplotlib.org/contents.html
4. Scikit-learn documentation: https://scikit-learn.org/stable/modules/classes.html
5. Description of categories of machine learning models and different algorithms: https://towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a
6. Matplotlib with Flask: https://stackoverflow.com/questions/20107414/passing-a-matplotlib-figure-to-html-flask

Big Data sources

You may use any dataset you'd prefer, as long as it meets the criteria for the Final Project. Below is a short, non exhaustive list of Big Data dataset options:

1. Forbes list: https://www.forbes.com/sites/bernardmarr/2018/02/26/big-data-and-ai-30-amazing-and-free-public-data-sources-for-2018/?sh=5f4a369f5f8a
2. Springboard list: https://www.springboard.com/blog/free-public-data-sets-data-science-project/
3. Data from the City of Austin, TX: https://data.austintexas.gov/
4. Kaggle Open Source Data Sets for ML: https://www.kaggle.com/datasets

Deliverables

All Python files, in the same directory, and compressed in a zip file. The zip file should be named:

`ITP_216_FP_YourLastName_YourFirstName.zip`

Grading

Section	Points (Total: 30)
Functionality and User Interface 1. The root endpoint shall display an input section for client-supplied data. 2. Clients shall be able to select a query of existing data, or a prediction based on existing data. 3. Clients shall be able to submit query information, which will generate a POST request. 4. Clients shall be able to see queried information on returned pages.	4 (1 point each)
General Code 1. The code shall contain no global objects other than those provided. 2. View functions shall only contain code related to the view function itself; anything else (e.g. querying the database, constructing a pandas object, et al.) shall be separated and held in its own function.	2 (1 point each)
Web App 1. Flask shall be used to create the web framework routing using endpoints and associated view functions. 2. The web app shall query and manipulate the dataset. 3. The web app shall contain at least 2 GET endpoints. (2 points) a. At least 1 of these shall be a dynamic endpoint created by a client POST request. 4. The web app shall contain at least 2 POST endpoints. (2 points)	6 (1 point each)
Scientific Computation 1. Pandas shall be used for manipulation of the data. 2. The app shall calculate some sort of meaningful aggregation.	2 (1 point each)
Machine Learning 1. Scikit-learn shall be used for the Machine-Learning aspects. 2. A projection shall be created of at least one of the features of the dataset.	2 (1 point each)
Data Visualization 1. Matplotlib shall be used to visualize the data once it has been loaded, prepared, and manipulated. 2. Figures shall have a title. 3. Plot axes shall be labelled. 4. Plots shall contain a legend, and datasets plotted shall be named in the legend. 5. Plot axes shall have values clearly visible (numbers, names, et al.). 6. A distinction shall be made for any plot as to whether it represents existing data or predictive data.	6 (1 point each)
Documentation and Formatting 1. Concise and useful commenting in your codebase is a must. You will need a header with your name, the semester, the section of the course you are in, and the assignment number. 2. You need descriptions of any major sections in your code (functions, classes, methods, et al.).	3 (1 point each)

3. Your code must be generally clear and readable.	
Error Handling 1. The web app shall run with no errors. 2. The web app shall reroute appropriately when given a nonsensical request (e.g. an endpoint that a client isn't meant to request directly, a POST with the wrong data, et al.)	2 (1 point each)
Extra points for free!	3
Extra Credit (database) 1. All data is stored in the database accurately (i.e. appropriate tables, key relationships [if any], attributes, and constraints on attributes). 2. The database is queried correctly given the client input, and returns appropriate data.	5 (2.5 points each)