Lab 04: Data App
CS3300 Data Science

Learning Outcomes

1. Understand the basic process of data science and exploratory data analysis including modes of inquiry (hypothesis driven, data driven, and methods driven).
2. Identify, access, load, and prepare (clean) a data set for a given problem.
3. Communicate findings through generated data visualizations and reports.

Overview

As part of their roles, data scientists may create "data apps" that enable non-technical users like business analysists to analyze data on their own. Data apps use interactive visualizations and simple widgets to support basic queries and filtering. Dashboards are a special type of data app that sources data from a database or other live source enabling real-time monitoring of business data.

Bokeh (Python) is an open-source tool for creating data apps that are commonly used by data scientists. Commercially supported dashboard tools include Tableau and Microsoft Power BI. Bokeh apps often generate single-page web apps saved as a HTML file. The logic for constructing the app is written in Python and converted to HTML and JavaScript by the Bokeh library.

In this lab, you are going to use Bokeh to create a data app to explore the real estate data you worked with previously.

Bokeh's documentation:
https://bokeh.pydata.org/en/latest/index.html

Instructions

You should use the Bokeh library to create a dashboard to visualize and explore the given dataset. The dashboard/visualization should run completely in your Jupyter notebook. For this submission you will need to submit the .ipynb file that has a working version of the visualization and a pdf of the finished notebook. It should be noted that Bokeh supports development of apps/dashboards as Python scripts. With this approach you can use the library to create a single page web app (HTML file). If you wish to explore this functionality in this lab, you have the freedom – but the submission should still be in the form of a Jupyter notebook.

Part 1: Display Real Estate on a Scatter Plot

For this part, you may need first to check the following quick guides:

1. First Steps: 1. Creating a Line Chart
2. First Steps: 2. Adding and customizing renderers
3. Displaying in a jupyter notebook
4. Scatter plots in Bokeh
5. Data Sources: a. Providing data with python lists, b. Providing numpy data and c. Providing data as a ColumnDataSource (check up to the subsection Using a pandas dataframe, you don't need to check after this subsection)

Define a method (make_plot) that accepts a ColumnDataSource (bokeh.models) as an argument. This method should return a bokeh figure object (bokeh.plotting).

- Data Frames can be used in the creation of ColumnDataSource objects (Using a pandas dataframe).
- Use the latitudes and longitudes to create a scatter plot. Label the axes.
- Color the points by residence type. Here you have two options: 1. you may add a "color" column to your dataframe that specifies a unique hex color that matches the residence type (you can check bokeh.palletes to choose the hex color: example) and then assign the name of this column to the parameter color of the circle() method (example). OR 2. Use factor_cmap() (example).
- When the user puts their mouse over a point, display the address, price, square footage, number of beds, and number of baths as a tooltip. (example: scroll down to see a complete code example)

At the end of experiment 1, you should have a make_plot method and display the whole dataset using show(figure).

Part 2: Refine ColumnDataSource Object based on Search Criteria

Now that you have a method definition that creates a scatterplot from a ColumnDataSource, you will want to be able to create new plots depending on search criteria. Create a new method definition (make_dataset) that accepts a dataframe and separate lists that describe ranges of values for each of the features you are interested in. This method should return a new ColumnDataSource object.

- Your method should accept lists describing the following inclusion criteria for the ColumnDataSource that you will create.
    - Residential type
    - Price range
    - Baths range
    - Beds range
    - Square Foot range

- Your method should “filter” your given dataframe to only include entries that meet the range/search criteria.

At the end of experiment 2, you should have a make_dataset method and display a plot of all the residential properties valued between $50,000 and $75,000 with 1-2 bathrooms, between 1000 to 2000 square feet, and 1-2 bedrooms.

Part 3: Add Widgets and Create and Interactive Visualization

With ability to create a new ColumnDataSource given ranges/filter parameters, you can now incorporate controls for your end user to specify desired ranges. Create widgets to add to your visualization/data app to set the desired filter criteria.

- CheckboxGroup – For residential type
- RangeSlider – For Price, Square Footage, Number of Beds, and Number of Baths.

To help ease event handling in python/bokeh, we have provided a partially complete method. You will need to fill in the method. When run, it should display your working figure. (Note: you will want to design your widgets, columns, and controls to adhere to the variable names/definitions provided in the method).

- Check the code provided in the next page (you don’t need to add anything to the update(attr, old, new) method).
- After defining the widgets, you will have to initialize a **`source`** (using make_dataset) and **`figure_object`** (using make_plot).
- Bonus Material – It is often helpful to display tabular data based on the returned observations. You may choose to implement a table widget that is refined based on search criteria. Try to display the address and associated price.

This experiment should conclude with the interactive visualization that includes working sliders for Price, Square Footage, Number of Beds, and Number of Baths.

For more information, you can check the method on_change() and the following tutorials:

Part I - https://towardsdatascience.com/data-visualization-with-bokeh-in-python-part-one-getting-started-a11655a467d4

Part II - https://towardsdatascience.com/data-visualization-with-bokeh-in-python-part-ii-interactions-a4cf994e2512

Part III - https://towardsdatascience.com/data-visualization-with-bokeh-in-python-part-iii-a-complete-dashboard-dc6a86aa6e23

```
from bokeh.layouts import column, row
# you may need to import more modules

def modify_doc(doc):

    # Instantiate here the CheckboxGroup and RangeSlider objects
    # Check the update method below to make sure you choose the same identifiers for the objects




    # create the data source by calling the method make_dataset
    source =

    # call the method make_plot
    figure_object =



    # Update function takes three default parameters
    def update(attr, old, new):
        # Get the list of selected types
        selected_types = [housing_checkbox_group.labels[i] for i in housing_checkbox_group.active]

        # Make a new column source according to the selected properties
        source2 = make_dataset(data_frame_sacramento, property_type=selected_types,
                               price_range=[slider_price.value[0], slider_price.value[1]],
                               baths_range=[slider_baths.value[0], slider_baths.value[1]],
                               beds_range=[slider_beds.value[0], slider_beds.value[1]],
                               sqf_range=[slider_sqft.value[0],slider_sqft.value[1]]
                              )

        # Update the data of the main source
        source.data.update(source2.data)


    housing_checkbox_group.on_change('active', update)
    slider_price.on_change('value', update)
    slider_beds.on_change('value', update)
    slider_baths.on_change('value', update)
    slider_sqft.on_change('value', update)

    controls = #column() #pass in to the column the slider objects and the checkbox_group object

    doc.add_root(row(figure_object, controls))

show(modify_doc)
```

Submission Instructions

You will submit two separate files to Canvas. The .ipynb file containing your experiments and working visualization app. A pdf copy of your notebook.

Rubric

Followed submission instructions	5%
Formatting: Report is polished and clean. No unnecessary code. Section headers are used. The report contains an introduction and conclusion.	5%
Part 1: Display Real Estate on a Scatter Plot	
Scatter plot of longitude and latitude	10%
Tool tip functions appropriately	10%
Use of color palettes to distinguish residential type	5%
Function (make_plot) meets specification	5%
Part 2: Refine ColumnDataSource Object based on Search Criteria	
Function narrows/filters dataset based on given parameters	15%
Function (make_dataset) meets specification	5%
Expected plot (shows filtered observations)	5%
Part 3: Add Widgets and Create and Interactive Visualization	
Widget Creation	15%
Functional Data App	10%
Technical/Organizational/Presentation Proficiency	10%