Ploty's Dash Framework

Introduction:

Dash is a productive Python framework for building web applications.

Written on top of Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly custom user interfaces in pure Python. It's particularly suited for anyone who works with data in Python.

Through a couple of simple patterns, Dash abstracts away all of the technologies and that are required to build an interactive web-based application. Dash is simple enough that you bind a user interface around your Python code in an afternoon.

Dash apps are rendered in the web browser. You can deploy your apps to servers and then share Them through URLs. Since Dash apps are viewed in the web browser, Dash is inherently Cross-platform and mobile-ready.

Dash is a user interface library for creating analytical web applications. Those who use Python for data analysis, data exploration, visualization, modeling, instrument control, and reporting will find immediate use for ash.

While Dash apps are viewed in the web browser, you don't have to write any Javascript or HTML. Dash provides a Python interface to a rich set of interactive web-based components.

Dash provides a simple reactive decorator for binding your custom data analysis code to your Dash user interface.

```
@dash_app.callback(Output('graph-id', 'figure'),Input('slider-id', 'value')])
def your_data_analysis_function(new_slider_value):
  new_figure = your_compute_figure_function(new_slider_value)
  return new_figure
```

When an input element changes (e.g. when you select an item in the dropdown or drag a slider), Dash's decorator provides your Python code with the new value of the input.

Your Python function can do anything that it wants with this input new value: It could filter a Pandas DataFrame, make a SQL query, run a simulation, perform a calculation, or start an experiment. Dash expects that your function will return a new property of some element in the UI, whether that's a new graph, a new table, or a new text element.

Architecture:

Dash applications are web servers running Flask and communicating JSON packets over HTTP requests. Dash's frontend renders components using React.js, the Javascript user-interface library written and maintained by Facebook.

Flask is great. It's widely adopted by the Python community and deployed in production environments everywhere. The underlying instance of Flask and all of its configurable properties is accessible to Dash app developers. For advanced developers, Dash apps can be extended through the rich set of Flask Plugins as well.
React is fantastic too. At Plotly, we've rewritten our entire web-platform and our online chart editor with React. One of the incredible things about React is how prolific and talented the community is. The open-source. React community has published thousands of high-quality interactive components, from Dropdowns to Sliders to Calendar Pickers to Interactive Tables.
Dash leverages the power of Flask and React, putting them to work for Python data scientists who may not Expert Web programmers.

The full set of HTML tags, like , <img/>, <table/> are also rendered dynamically with React and their Python classes are available through the dash_html_component library. A core set of interactive components like Dropdown, Graph, Slider will be maintained by the Dash core team through the dash_core_components library. Both of these libraries use the standard open-source React-to-Dash toolchain that you could use if you were to write your own component library.

Data Visualization:

Dash ships with a Graph component that renders charts with plotly.js. Plotly.js is a great fit for Dash: it's declarative, open-source, fast, and supports a complete range of scientific, financial, and business charts. Plotly.js is built on top of D3.js (for publication-quality, vectorized image export) and WebGL (for high-performance visualization).

Dash's Graph element shares the same syntax as the open-source plotly.py library, so you can easily switch between the two. Dash's Graph component hooks into the plotly.js event system, allowing Dash app authors to write applications that respond to hovering, clicking, or selecting points on a Plotly graph.

Dash Installation:

One can install all the Dash Libraire using: **pip install dash==1.12.0**

***Note**: starting with dash 0.37.0, dash automatically installs dash-renderer, dash-core-components, dash-html-components, and dash-table, using known-compatible versions of each. You need not and should not install these separately any longer, only dash itself.*

Basic code layout:

```
import dash
import dash_core_components as dcc
import dash_html_components as html

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
```

```
app.layout = html.Div(children=[
    html.H1(children='Hello Dash'),

    html.Div(children='''
        Dash: A web application framework for Python.
    '''),

    dcc.Graph(
        id='example-graph',
        figure={
            'data': [
                {'x': [1, 2, 3], 'y': [4, 1, 2], 'type': 'bar', 'name': 'SF'},
                {'x': [1, 2, 3], 'y': [2, 4, 5], 'type': 'bar', 'name': u'Montréal'},
            ],
            'layout': {
                'title': 'Dash Data Visualization'
            }
        }
    )
])

if __name__ == '__main__':
    app.run_server(debug=True)
```

Note: Copy the above code in a file and run it in the cmd using the following command

```
:$ python dash_app.py
```