UNIVERSITY 4 INDUSTRY

Einführung Plotly Dash

Robert Heise und Florian Edenhofner – Education4Industry GmbH

26.06.2023

University4Industry

Agenda

Kurzeinführung App-Entwicklung mit Ploty Dash

- Was ist Plotly Dash?
- Prinzipieller Grundaufbau des Quellcodes
- Beschreibung des Layouts
- Beschreibung der Funktionalität (Callbacks)
- Erweiterte Beschreibung des Layouts mit Bootstrap-Komponenten

Was ist Plotly Dash?

Plotly Dash ist ein Framework speziell für die Entwicklung analytischer Dashboards.

- flexibles Framework für die Erstellung von Dashboards.
- Point&Click-Interface
- einfaches low-code-Development direkt in Python
- Kenntnisse in Html, CSS und JavaScript sind nicht notwendig
- basierend auf Flask, Plotly.js, React.js
- Verfügbar für Python, R, Julia, F#

`https://dash.plotly.com`

Tutorials mit zahlreichen Beispielen

Grundlagen Dash-App

Typische Schritte: Initialisierung, Beschreibung des Layouts, Beschreibung der Funktionen, Starten der App.

```
from dash import Dash, html

# Inizialisierung der Dash-App
app = Dash()

# Beschreibung des Layouts - Wie solle die App aussehen?
app.layout = html.Div(...)

# Beschreibung der Funktionalitäten (callbacks)
@app.callback(Output(...),Input(...))
def graph_update(...):
  ...
  return output_values

# Starten der Dash-App
if __name__ == '__main__':
    app.run_server(debug=True) # Startet Server im Debug-Modus
```

Dash Layout: Html-Komponenten

HTML-Komponenten bilden die Basis das Layout einer Dash-App. Sie beschrieben typische Html-Komponenten, wie Überschriften oder Buttons.

```
from dash import Dash, html
```

Verwendung einiger Html-Komponenten zur Beschreibung des Layouts

```
# Beipiel für das Erstellen eines Layouts der App ...
# ... durch das Zusammenfügen verschiedener HTML-Komponenten
app.layout = html.Div(
    children=[
        html.H1(children='Eine Beispiel-App',
                style={'textAlign': 'center', 'marginTop': 40, 'marginBottom': 40}),
        html.H2(children='Wissenswertes'),
        html.Div(children='Beipieltext. Dash ist ein tolles Ding!'),
    ]
)
```

siehe auch `dash_example_1_html_components.py`

Dash Layout: weitere Html-Komponenten

Einige Beispiele für Html-Komponenten

```
1   html.Div()          # Division - Generische Html-Container
2   html.Button()       # Druckschaltfläche
3   html.H1()           # Überschrift Größen 1-3
4   html.H2()
5   html.H3()
6   html.Img()          # Bild
```

Weiter Information über HTML-Komponenten unter

`https://dash.plotly.com/dash-html-components`

Informationen zu einzelnen Html-Komponenten unter

`https://developer.mozilla.org/en-US/docs/Web/HTML`

Dash Layout: Core components

Die Kern-Komponenten (core components) erweitern die Html-Komponenten durch speziell für Dash entwickelte Objekte.

```
from dash import dcc # Import der Dash Core Components (dcc)
```

Beispiel: Layout mit dcc.Graph und plotly.express

```
import plotly.express as px
df = px.data.stocks()
fig = px.line(df, x=df['date'], y=df['GOOG'])

app.layout = html.Div(
    children=[
        html.H1(children='Zeitreihe Aktie Google'),
        dcc.Graph(figure=fig) # Graph-Komponente
    ]
)
```

siehe auch `dash_example_2_dcc_plotly.py`

Dash Layout: weitere Core components

Einige Beispiele für weitere Komponenten

```
dcc.Graph()
dcc.Input()
dcc.Dropdown()
dcc.Slider()
dcc.Textarea()
dcc.Checklist()
dcc.RadioItems()
dcc.DatePickerSingle()
```

Weiter Information über HTML-Komponenten unter

`https://dash.plotly.com/dash-core-components`

Dash Callbacks

Um die Funktionalität der App zu erklären werden Callbacks verwendet. Eine Funktionalität entspricht einer Aktion, welche ausgelöst wird und eine zugeordneten Änderung als Effekt hat. Solche Aktionen können z.B an einen Element ausgelöst werden. Einzelne des Layout müssen mittels des Argumentes id identifiziert werden.

Beispiel: Layout mit dcc.Dropdown und id's für die Verwendung mit Callbacks

```
#Layoutbeschreibung mit Id's und Dropout-Element
app.layout = html.Div(
    children=[
        html.H3(id='title',children='Aktienkurse'),              # erhält id='title'
        dcc.Dropdown(id='dropdown',
                     options=[
                         {'label': 'Google', 'value': 'GOOG'},
                         {'label': 'Apple', 'value': 'AAPL'},
                         {'label': 'Amazon', 'value': 'AMZN'},
                     ],
                     value='GOOG'),
        dcc.Graph(id='line_plot'),                             # erhält id='line_plot'
    ]
)
```

siehe auch `dash_example_3_dcc_with_callbacks.py`

Dash Callbacks

Für die Erklärung der Callbacks in Dash werden Dekoratoren verwendet. Sie werden nach dem Layout erklärt und verküpfen verschiede Elemente.

```
from dash.dependencies import Input, Output
```

Beispiel für Callback

```
@app.callback(Output(component_id='line_plot', component_property= 'figure'),
              Input(component_id='dropdown', component_property= 'value'))
def graph_update(value_of_input_component):
    print(value_of_input_component)
    fig = px.line(df, x = df['date'], y = df[value_of_input_component])
    return fig
```

- Als Dekorator wird eine Function Factory verwendet
- Output- und Inputelement und deren Property wird definiert
- Übergabe der Funktionsargument erfolgt in der Reihenfolge der Definition der Inputs
- Der Rückgabewert der Funktion entspricht der neuen Property des Outputs

Dash Callbacks - vollständige App: dash_example_3_dcc_with_callbacks.py

```
1  from dash import Dash,dcc,html
2  from dash.dependencies import Input, Output
3  import plotly.express as px
4  df = px.data.stocks()
5  app = dash.Dash()
6
7  #Layoutbeschreibung mit Id's und Dropout-Element
8  app.layout = html.Div(children=[html.H3(id='H2',children='Aktienkurse'),
9                                  dcc.Dropdown(id='dropdown',
10                                     options=[{'label': 'Google', 'value': 'GOOG'},
11                                              {'label': 'Apple', 'value': 'AAPL'},
12                                             {'label': 'Amazon', 'value': 'AMZN'},],
13                                     value='GOOG'),
14                                 dcc.Graph(id='line_plot')])
15
16 # Callback erklärt Funktionalität des Dropdown-Element
17 @app.callback(Output(component_id='line_plot', component_property='figure'),
18               Input(component_id='dropdown', component_property='value'))
19 def graph_update(value_of_input_component):
20     print(value_of_input_component)
21     fig = px.line(df, x=df['date'], y=df[value_of_input_component])
22     return fig
23
24 if __name__ == '__main__':
25     app.run_server(debug=True)
```

Bootstrap-Komponeten

Durch Dash-Bootstrap-Komponenten können Dash-Apps um Bootstrap-CSS erweitert werden. Dadurch können u.a. z.B. Zeilen und Spalten erzeugt werden. Es stehen weitere Bootstrap-Element zur Verfügung (z.B. Cards).

```
1  import dash_bootstrap_components as dbc
```

Benutzen eines externen Stylesheets ändert Erscheinung der App

```
1  app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])
```

Beispiel für Layout mit Bootstrap-Elementen Row und Col

```
1  app.layout = html.Div(
2    children=[
3      dbc.Row([                                  # Erstellung einer Zeile
4        dbc.Col([html.H1('HTML Component')]),    # Erstellung einer Spalte in dieser Zeile
5        dbc.Col([html.H1('HTML Component')]),
6        ], align='center'),
7    ]
8  )
```

siehe auch `dash_example_4_dbc_with_callbacks.py`

Zusammenfassung Komponeten

Verschiedene Komponententypen

- Html components (HTML-Elemente)
- Core components (dash-eigene Bedienelemente)
- Bootstrap components (Bootstrapelement)
- Dash DAQ (weitere weitere dash-eigegne Bedienelemente)

$\rightarrow$ Es existieren weitere Pakete mit speziellen Elementen.

Ändern von IP und Port

```
1    if __name__ == '__main__':
2        app.run_server(host = '0.0.0.0', port=8080, debug=False)
```

Eine auf dem Raspberry Pi laufende Dash-Applikation kann durch Angabe der IP-Adresse des Raspberry's im lokalen W-Lan freigegeben werden.

Das Auto kann auf diese Weise z.B. durch einen Klick auf einene Button gestartet oder gestopp werden.

Tipps zum Einarbeiten in Dash

- Verstehen Sie den Grundaufbau
  - Layout + Callbacks (`https://dash.plotly.com/`)
- Einzelne Element haben eine Vielzahl von Stylingoptionen
- Verschaffen Sie sich einen Überblick über die verschiedenen Elemente
  - html components (`https://dash.plotly.com/dash-html-components`)
  - dash core components (`https://dash.plotly.com/dash-core-components`)
  - bootstrap components (`https://dash-bootstrap-components.opensource.faculty.ai/`)
- Erkunden Sie die Stylingoptionen nur für ausgewählte Elemente