Coach Market Woche 5

Robert Heise und Florian Edenhofner – Education4Industry GmbH

03.12.2021

University4Industry

Agenda

1. Doing Data Science with Python
    - Exploring and Processing Data - Part 1
2. Building Data Visualisations Using Plotly
    - Building Basic Charts with Plotly

Doing Data Science with Python

Kurze Zusammenfassung wichtiger Themen

- Erfassen grundlegender Strukturen des Datasets
- Zusammenfassende Statistik für numerische und nicht-numerische Daten

Heutiges Ziel

- Umsetzung der obigen Themen in Python/Pandas

Grundlegende Strukturen eines Datasets

Verschaffen Sie sich einen Überlick über das Dataset (technische Aspekte).

- Anzahl der Spalten und deren Bedeutung
- Anzahl der Zeilen
- Datentypen
- Gibt es fehlende Werte?
- Beurteilung der Datenqualität

Zusammenfassende Statistik für numerische Werte

Erlaubt ersten Überblick über die Verteilung der Daten

Centrality Measures (Zentralitätsmaße/Lageparameter):

- Mittelwert (arithmetische Mittel) `mean`
  kann durch extreme Werte verzerrt werden
- Median (Zentralwert) `median`
  ist weniger anfällig für extreme Werte

Spread Measures (Dispersionsmaße/Streuungsparameter)

- Minimum `min`
- Maximum `max`
- Spannweite `range` (`max-min`)
- Quartilen `quantile`
- Varianz `var`
- Standardabweichung `std`

Varianz

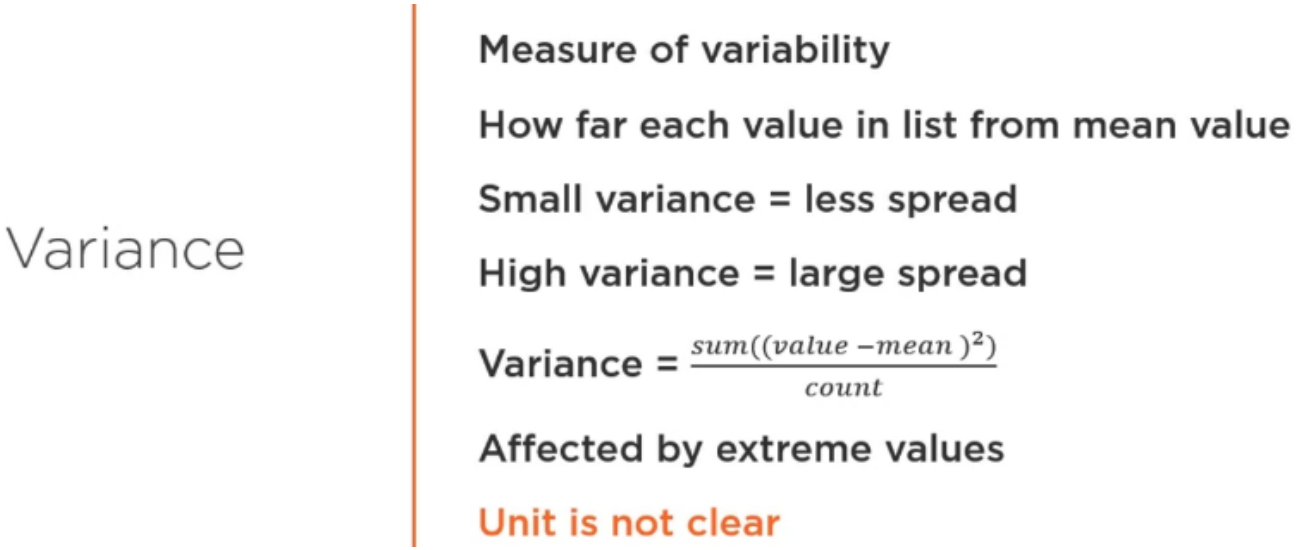

Figure 1: Eigenschaften der Varianz

Standardabweichungen

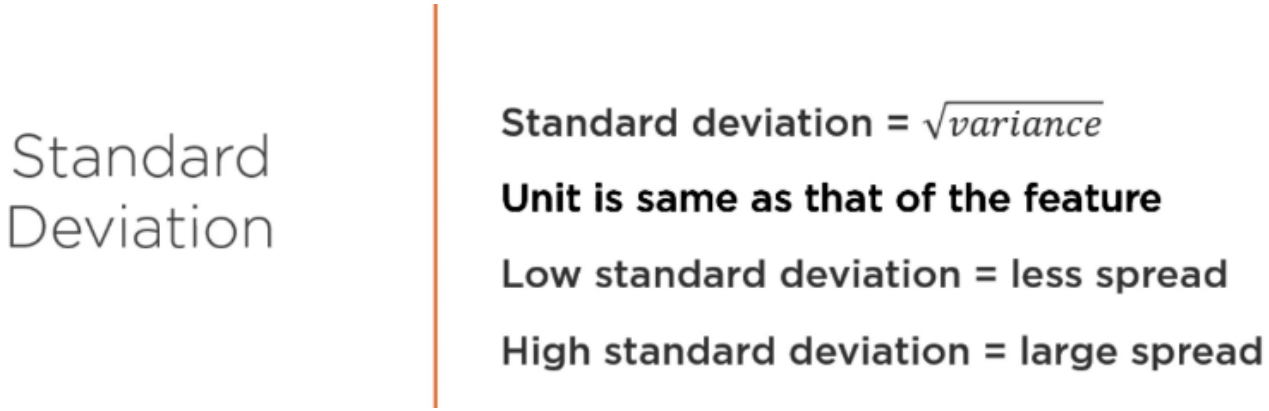

Figure 2: Eigenschaften der Standardabweichung

Deskriptive Statistik für nicht-numerische Datentypen

Was sind nicht-numerische Datentypen?

Die Daten beschreiben damit die Zugehörigkeit zu einer Klasse oder Kategorie (z.B. Geschlecht, Apfelsorten, Berufsgruppen). Diese Kategorien besitzen keine Ordnung! (nominaler Datentyp, engl. categorical datatype)

Absolute und relative Häufigkeit des einzelnen Kategorien

- Pandas Methoden: `unique`, `nunique`, `value_counts`
- Histogramme geben diese Häufigkeiten grafisch wieder

Modus/Modalwert

- Häufigster Wert (Wert, welcher “am ehsten” zu beobachten wäre)

Building Basic Charts with Plotly

Was ist Plotly?

Plotly ist ein Unternehmen mit Sitz in Montreal, Kanada.

- Entwicklungen in Data Analytics und Data Visualisation (Dash, Chart Studio, Plotly.js)

Plotly.js ist eine Open-Source-Bibliothek zur interaktiven Datenvisualisierung

- Geschrieben in JavaScript

Plotly.py ist eine Python-API für Plotly.js.

- Unterteilung in mehrere Module z.B. **plotly.graph_objects**
- Zahlreiche Beispiele und Dokumentation unter `https://plotly.com/python/`

Heutiges Ziel

- Erstellen einfacher Grafiken (Scatter Plots, Histogramme, Box Plots)
- Anwenden der Visualisierungen auf ein konkreten Dataset

Plotly.js: Anwendung

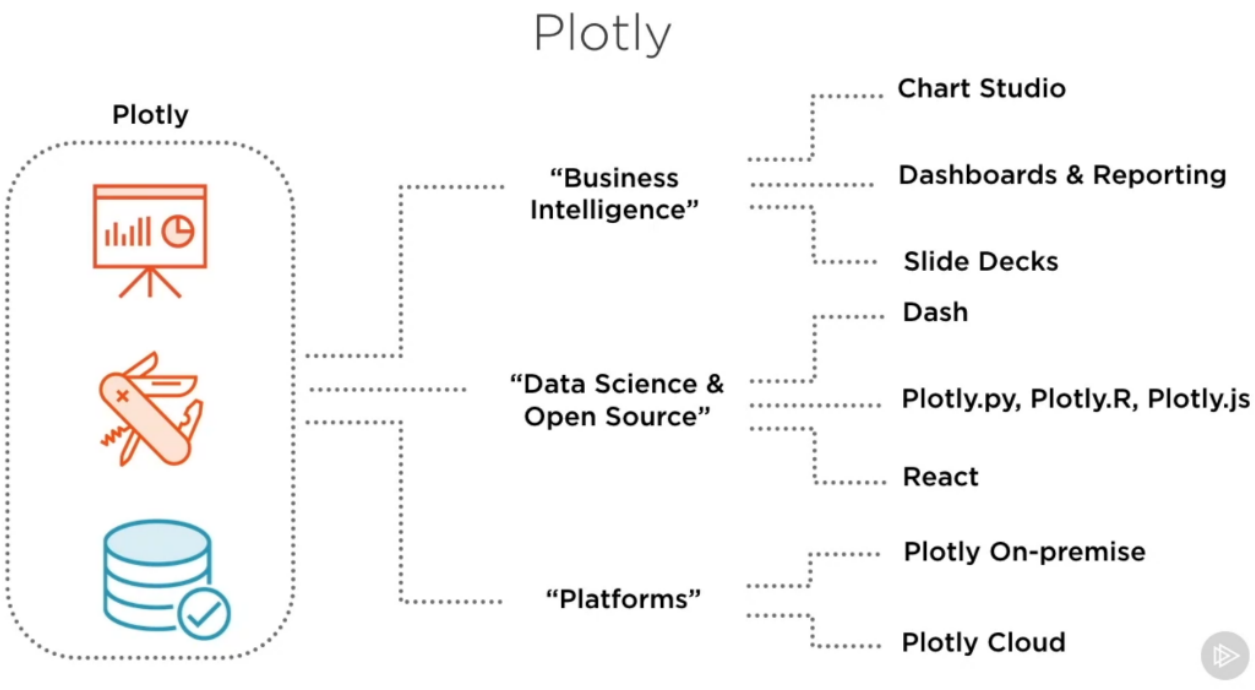

Figure 3: Plotly ist mehr als nur ein Python Package

Plotly.py: Basiselemente

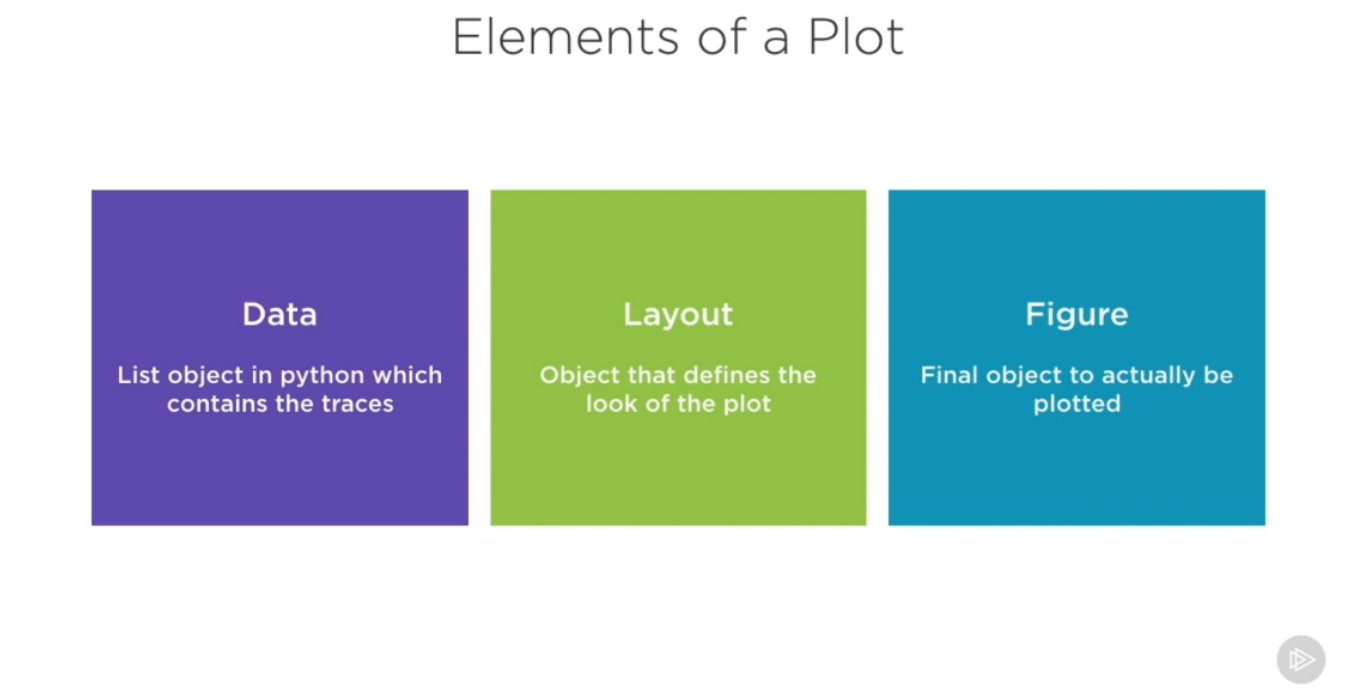

Figure 4: Die drei Komponenten jeder Plotly.js Visualisierung

Plotly.py: Basiselemente

Das Module **plotly.graph_objects** erlaubt die Erstellung von Grafiken (Figure-Objekten).

Einfacher Bar Chart mit plotly.graph_objects

```
1  import plotly.graph_objects as go
2
3  fig = go.Figure(
4
5      data   = [ go.Bar( x = [1, 2, 3], y = [1, 3, 2] ) ],
6
7      layout = go.Layout( title="Ein einfaches Balkendiagramm" )
8  )
9
10 fig.show()
```

Komplexerer Scatter Plot mit plotly.graph_objects

```
1  import plotly.graph_objects as go
2  import numpy as np
3  x=np.linspace(0,1,10)
4  y1=2*x+np.random.rand(10)
5  y2=2*x+np.random.rand(10)
6
7  trace1 = go.Scatter(x=x, y=y1,
8                  name='Linie 1', mode='markers+lines',width = 2),
9                  line = dict(color=('rgb(250,0,2)'))
10
11 trace2 = go.Scatter(x=x, y=y2,
12                 name='Linie 2', mode='lines', width = 2
13                 line = dict(color=('rgb(150,150,150)'))
14
15 data = [trace2, trace1]
16
17 layout = go.Layout(title="Zufallszahlen + linearer Trend",
18                 xaxis=dict(title='x'),
19                 yaxis=dict(title='y = F(x)'))
20
21 fig = go.Figure(data = data, layout = layout)
22
23 fig.show()
```

Plotly.py: plotly.express

Das Module **Plotly.express** erlaubt als High-Level-API eine vereinfachte Erstellung von Figure-Objekten.

plotly.express

```
1  import plotly.express as px
2
3  fig = px.bar(x = [1, 2, 3], y = [1, 3, 2], title = "Ein einfaches Balkendiagramm")
4  fig.show()
```

plotly.graph_objects

```
1  import plotly.graph_objects as go
2
3  fig = go.Figure(
4      data   = [ go.Bar( x = [1, 2, 3], y = [1, 3, 2] ) ],
5      layout = go.Layout( title = "Ein einfaches Balkendiagramm" )
6  )
7  fig.show()
```

Plotly.py: Dataframes in plotly.express

Plotly.express erlaubt einen vereinfachten Zugriff auf Daten in einem Pandas Dataframe

Histogramm

```
1  import plotly.express as px
2  import numpy as np
3  x1 = [np.random.normal(0,1) for i in range(10000) ]
4  x2 = [np.random.normal(0,2) for i in range(10000) ]
5  dataframe = pd.DataFrame({'col1':x1,'col2':x2})
6
7  fig = px.histogram(dataframe, x="col1", nbins=100, title = 'Histogramm')
8  fig.show()
```