

# Data Science: Predictive Analytics, Übung 1

HENRY HAUSTEIN

## Business Understanding

- (a) viele Fahrräder vermieten
- (b) Fahrräder müssen gekauft und repariert werden, und zwar möglichst dann, wenn sie in absehbarer Zeit gebraucht werden bzw. gerade nicht gebraucht werden.
- (c) Vorhersage der vermieteten Fahrräder
- (d) ?
- (e) Die Daten sind kompletter Schrott: nur 2 von 10 (sinnvoll numerischen) Spalten sind numerisch; unnötige Informationen wie die Spalte *Seasons* (wenn ich das Datum kenne, weiß ich auch welche Jahreszeit gerade herrscht); doppelte Informationen in Spalte *Hotness* und *Temperature*; und was soll die Spalte *Functioning Day*?

## Data Understanding

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import missingno as msno
5
6 def fixCols(cols):
7     liste = []
8     for col in cols:
9         newName = col.split("(")[0].strip().replace(" ", "_")
10        liste.append(newName)
11    return liste
12
13 # 7.
14 data = pd.read_csv("dataset.csv", delimiter = ";", header = 0,
15                    index_col = 0)
16
17 # 8.
18 print(data.head(7))
19 print(data.tail(7))
20
21 # 9.
22 print(len(data.index))
```

```

22
23 # 10.
24 print(data.columns)
25 print(len(data.columns))
26
27 # 11.
28 data.columns = fixCols(data.columns)
29
30 # 12.
31 print(data.describe())
32
33 # 13.
34 data.info()
35
36 # 14.
37 print(data.isna().sum())
38 msno.matrix(data)
39 plt.show()
40
41 # 15.
42 print(data["Functioning_Day"].unique())
43 print(data["Seasons"].unique())
44
45 # 16.
46 print(len(data["Solar_Radiation"].unique()))
47
48 # 17.
49 data.hist(column = "Rented_Bike_Count", bins = 20)
50 plt.title("Rented Bike Count")
51 plt.xlabel("# Bikes")
52 plt.ylabel("Häufigkeit")
53 plt.show()
54
55 # 18.
56 means = data.groupby("Hour").mean()
57 plt.plot(means["Rented_Bike_Count"])
58 plt.title("ausgeliehene Räder über den Tag")
59 plt.xlabel("Uhrzeit")
60 plt.ylabel("durchschnittlich ausgeliehene Räder")
61 plt.show()
62
63 # 19.
64 print(data.corr())
65
66 # 20.
67 corr = data.corr()
68 sns.heatmap(corr, xticklabels = corr.columns, yticklabels = corr.
              columns)
69 plt.show()

```