**Definitionen:**

Objects: Eine Datenstruktur, die Attribute (Daten) und Methoden (Funktionen) enthält, wie z. B. Datenrahmen (dataframes), Arrays, Listen und dictionaries.

Beispiele:

Liste der Zahlen: [1, 3, 7, 10, 15]

Dataframe: Eine Datenstruktur, die zur Speicherung und Bearbeitung von Daten verwendet wird, ähnlich wie Excel-Tabellen.

**Object Funktionen:**

round(Zahl, Dezimalzahlen\_gewünscht) : Eine Funktion zum Runden einer Zahl auf eine bestimmte Dezimalzahl

len(): Eine Funktion, mit der man feststellen kann, wie viele Elemente in einem object enthalten sind, z. B. lists, arrays, dataframes.

**Pandas Funktionen:**

pandas.read\_excel():  Eine Funktion zum Laden von excel-Dateien als dataframe (read\_csv, read\_json, etc).

dataframe.head(): Eine Funktion zum Anzeigen des dataframe.

dataframe.info(): Eine Funktion zur Anzeige von Informationen über den dataframe (data types, null values).

dataframe.nunique(): Eine Funktion, um herauszufinden, wie viele eindeutige/verschiedene Einträge es in einer bestimmten Spalte gibt.

dataframe.fillna(Wert, inplace=True): Eine funktion, to replace a ‘not a number’ with a specific value.

dataframe.rename(columns={'name\_to\_change': 'new\_name'}, inplace=True): Ändert den Namen einer Spalte.

dataframe.pivot(index='varible\_1',columns='variable\_2')

groupby(): Eine Funktion zum Gruppieren von Zeilen eines dataframe basierend auf einem bestimmten Merkmal.

value\_counts(): Eine Funktion, die eine **\*\*Anzahl der eindeutigen Werte\*\*** in einer Spalte zurückgibt.

drop(): Eine Funktion zum Entfernen einer Zeile oder Spalte aus dem dataframe.

dataframe\_1.merge(dataframe\_2, how=, on=):  Eine Funktion zum Zusammenführen zweier dataframe auf der Grundlage einer Bedingung.

sort\_values(by='parameter\_by\_which\_to\_sort'):  Eine Funktion zum Sortieren von Elementen in einem object.

loc[]: Eine Funktion zur Auswahl von Zeilen auf der Grundlage von Bedingungen

idxmax(): Gibt den Index des Maximalwerts zurück.

idxmin(): Gibt den Index des Mindestwerts zurück.

agg(): Eine Methode zur Anwendung von 1 oder mehreren Funktionen auf einen dataframe. Beispiele: 'mean', 'count', 'sum', 'max'.

series.to\_frame(): Konvertiert eine series in einen dataframe.

**Seaborn Funktionen:**

**Numerische Features**

hist()

Beschreibung: Erstellt ein Histogramm, um die Verteilung eines numerischen Features zu visualisieren.

Beispiel: dataframe[„muster“].hist(bins=15)

boxplot()

Beschreibung: Zeigt Median, Quartile und Ausreißer eines numerischen Features. Kann auch für kategoriale vs numerische Features genutzt werden.

Beispiel: seaborn.boxplot(x= unabhängige\_Eigenschaft, y= abhängige\_Eigenschaft, data=dataframe)

scatterplot()

Beschreibung: Visualisiert die Beziehung zwischen zwei numerischen Features, z. B. zur Zielvariable.

Beispiel: seaborn.scatterplot(x='BMI', y='price', data=df)

regplot()

Beschreibung: Erstellt einen Scatterplot mit Regressionslinie, um den linearen Zusammenhang zwischen zwei numerischen Variablen zu visualisieren.

Nutzen: Zeigt, ob und wie stark ein Trend (z. B. steigend oder fallend) zwischen Variablen besteht.

Beispiel: seaborn.regplot(x='BMI', y='price', data=df)

**Kategoriale Features**

countplot()

Beschreibung: Erstellt ein Balkendiagramm, um die Anzahl der Beobachtungen in jeder Kategorie anzuzeigen.

Beispiel:

seaborn.countplot(data=None,     # DataFrame containing the data

                  x=,            # Variable for x-axis (categorical)

                  y=,            # Variable for y-axis (categorical)

                  hue=,          # Grouping variable for color encoding

                  palette=None,  # Color palette to use

)

boxplot() (Kategorial vs Numerisch)

Beschreibung: Zeigt, wie sich ein numerisches Feature über verschiedene Kategorien verteilt, inklusive Median, Quartile und Ausreißer.

Beispiel: seaborn.boxplot(x=unabhängige\_Eigenschaft, y=abhängige\_Eigenschaft, data=dataframe)

**Zusammenhänge zwischen Features**

heatmap()

Beschreibung: Zeigt eine Farbmatrix der Korrelationen zwischen numerischen Features. Hilfreich, um starke Zusammenhänge zu erkennen.

Beispiel: seaborn.heatmap(dataframe.corr(), annot=True, cmap='coolwarm')