

# Einfaches Beispiel für Support Vector Regression (SVR)

## Vorhersage einer linearen Regression (mit Daten aus Kapitel 5).

Wir nutzen die Datenpunkte eines linearen Beispiel einer Regression aus dem Kapitel "Regression". Dies erfolgt in mehreren Schritten:

1. Datenerstellung: wir nehmen die Datenpunkte aus Aufgabe (5.6)-Teil2
2. SVR-Modell: Ein linearer-Kernel wird verwendet, da es sich um eine lineare Regression handelt. Die Parameter C und gamma steuern die Anpassung des Modells.
3. Vergleich der Regressionsgeraden aus Aufgabe (5.6)-Teil2 mit der Geraden die durch SVR approximiert wurde

Autor: Dr. H. Völlinger, Dezember 2024

In [1]:

```
1  # Import and check needed libraries
2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from sklearn.svm import SVR
6
7  # to check the time of execution, import function time
8  import time
9
10 # check versions of libraries
11 print('numpy version is: {}'.format(np.__version__))
```

numpy version is: 1.18.1

## Step1: Daten bereistellen und ausplotten

In diesem zweiten Schritt nehmen wir die Daten aus Übungsaufgabe (5.6)-Teil2. Diese Daten sind gegeben als 10 Datenpunkte in dem Bereich vom 3 nach 10 in der x-Achse und (19, 48) in der y-Achse. Wir plotten diese Daten aus.

```

In [2]: 1 # Daten erstellen
2 x = np.array([ 5, 4, 5, 3, 9, 8, 10, 5, 3, 3]).reshape((-1, 1))
3 y = np.array([41,27,35,26,48,45,46, 27,29,19])
4
5 # Daten ausdrucken
6 print ("This is how x and y look now:")
7 print("x=",x)
8 print("y=",y)
9
10 # Datenpunkte plotten
11 plt.scatter(x, y, color='orange', label='Datenpunkte') # Streudiagramm
12 plt.xlabel('x')
13 plt.ylabel('y')
14 plt.title('Datenpunkte')
15 plt.legend()
16 plt.show()

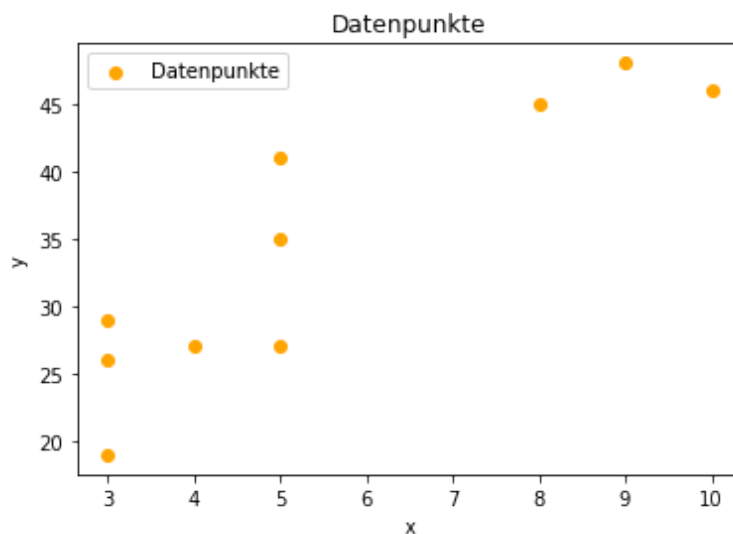
```

This is how x and y look now:

```

x= [[ 5]
 [ 4]
 [ 5]
 [ 3]
 [ 9]
 [ 8]
 [10]
 [ 5]
 [ 3]
 [ 3]]
y= [41 27 35 26 48 45 46 27 29 19]

```



## Step2: SVR Modell definieren (Typ =linear) und anwenden

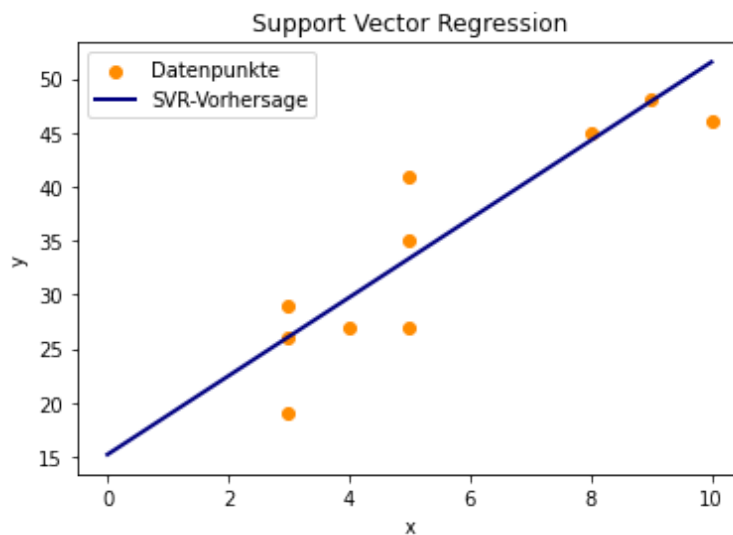
Setzen der 2 linearen SVR-Modell-Parameter:  $C = 100$ ,  $\gamma = \text{auto}$

```
In [3]: 1 # SVR-Modell trainieren
2 svr_lin = SVR(kernel="linear", C=100, gamma="auto")
3 svr_lin.fit(x, y)
4
5 # Vorhersagen treffen
6 X_test = np.linspace(0, 10, 50).reshape(-1, 1) # Werte zum Testen
7 y_pred = svr_lin.predict(X_test)
```

## Ergebnisse visualisieren

Anzeige der SVR-Vorhersage und der Datenpunkte

```
In [4]: 1 # Ergebnisse visualisieren
2 plt.scatter(x, y, color='darkorange', label='Datenpunkte')
3 plt.plot(X_test, y_pred, color='navy', lw=2, label='SVR-Vorhersage')
4 plt.xlabel('x')
5 plt.ylabel('y')
6 plt.title('Support Vector Regression')
7 plt.legend()
8 plt.show()
```



## Support-Vektoren anzeigen und zählen

Alle Datenpunkte sind Support Vektoren

```

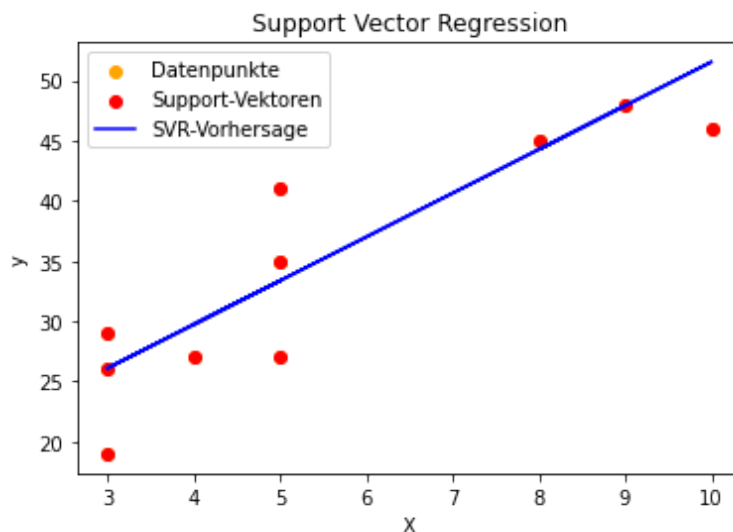
In [5]: 1 # Anzahl der Support-Vektoren
2 print(f"Anzahl der Support-Vektoren: {len(svr_lin.support_)}")
3
4 # Indizes der Support-Vektoren
5 print("Indizes der Support-Vektoren:")
6 print(svr_lin.support_)
7
8 # print("Support-Vektoren:")
9 # print(svr_lin.support_)
10
11 # Plot mit Support-Vektoren
12 plt.scatter(x, y, color='orange', label='Datenpunkte') # Alle Datenpunkte
13 plt.scatter(x[svr_lin.support_], y[svr_lin.support_], color='red', label='Support-Vektoren')
14 plt.plot(x, svr_lin.predict(x), color='blue', label='SVR-Vorhersage') #
15 plt.xlabel('X')
16 plt.ylabel('y')
17 plt.title('Support Vector Regression')
18 plt.legend()
19 plt.show()

```

Anzahl der Support-Vektoren: 10

Indizes der Support-Vektoren:

[0 1 2 3 4 5 6 7 8 9]



### Step3: Vergleich mit Ergebnissen der Übungsaufgabe (5.6)-Teil2

```

In [6]: 1 print('Lösungen bei Aufgabe (5.6)-Teil2 waren Steigung ~ 3,5 und Achsenabschnitt ~ 15')
2 print('Lösungen durch Abmessen der Datenpunkte: Steigung ~ 3,4 und Achsenabschnitt ~ 15')
3 print('Lösungen sind also bei beiden Verfahren durchaus vergleichbar!')

```

Lösungen bei Aufgabe (5.6)-Teil2 waren Steigung ~ 3,5 und Achsenabschnitt ~ 15

Lösungen durch Abmessen der Datenpunkte: Steigung ~ 3,4 und Achsenabschnitt ~ 15

Lösungen sind also bei beiden Verfahren durchaus vergleichbar!

In [7]:

```
1 # Print current date and time
2 print('***** Aktuelles Datum und Zeit*****')
3 print("Date & Time:",time.strftime("%d.%m.%Y %H:%M:%S"))
4 # end of import test
5 print ("Ende Python-Programm ***SVR_LinRegression ***")
```

\*\*\*\*\* Aktuelles Datum und Zeit\*\*\*\*\*

Date & Time: 10.12.2024 20:00:57

Ende Python-Programm \*\*\*SVR\_LinRegression \*\*\*