Einfaches Beispiel für Support Vector Regression (SVR)

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

Wir nutzen die Datenpunkte eines simplen Beispiel (2-dimensional) einer Regression aus dem Kapitel "Regression". Dies erfogt in mehreren Schritten:

1. Datenerstellung: wir nehem 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]:
```
# Import and check needed libraries

import numpy as np
import matplotlib.pyplot as plt
from sklearn.svm import SVR

# to check the time of execution, import function time
import time

# check versions of libraries
print('numpy version is: {}'.format(np.__version__))
```

```
numpy version is: 1.18.1
```

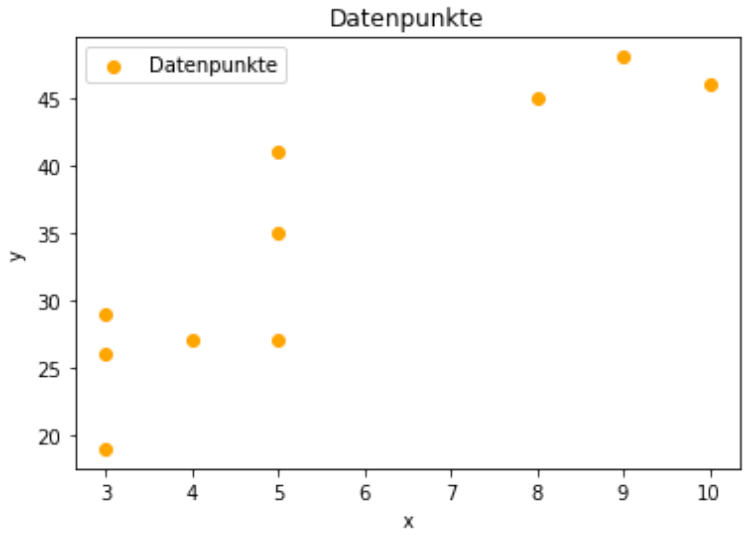
Step1: Daten bereistellen und ausplotten

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]:

```
# Daten erstellen
x = np.array([ 5, 4, 5, 3, 9, 8, 10, 5, 3, 3]).reshape((-1, 1))
y = np.array([41,27,35,26,48,45,46, 27,29,19])

# Daten ausdrucken
print ("This is how x and y look now:")
print("x=",x)
print("y=",y)

# Datenpunkte plotten
plt.scatter(x, y, color='orange', label='Datenpunkte')  # Streudiagramm
plt.xlabel('x')
plt.ylabel('y')
plt.title('Datenpunkte')
plt.legend()
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: 3 SVR Modelle definieren (Typ =linearm RBF und Poly(Grad=3)) und anwenden

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

In [3]:

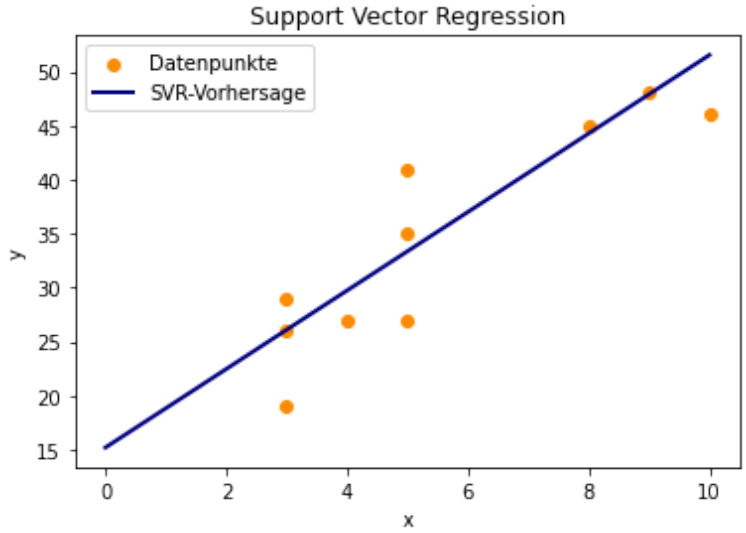
```python
# SVR-Modell trainieren
svr_lin = SVR(kernel="linear", C=100, gamma="auto")
svr_lin.fit(x, y)

# Vorhersagen treffen
X_test = np.linspace(0, 10, 50).reshape(-1, 1)  # Werte zum Testen
y_pred = svr_lin.predict(X_test)
```

Ergebnisse visualisieren

Anzeige der SVR-Vorhersage und der Datenpunkte

In [4]:

```python
# Ergebnisse visualisieren
plt.scatter(x, y, color='darkorange', label='Datenpunkte')
plt.plot(X_test, y_pred, color='navy', lw=2, label='SVR-Vorhersage')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Support Vector Regression')
plt.legend()
plt.show()
```



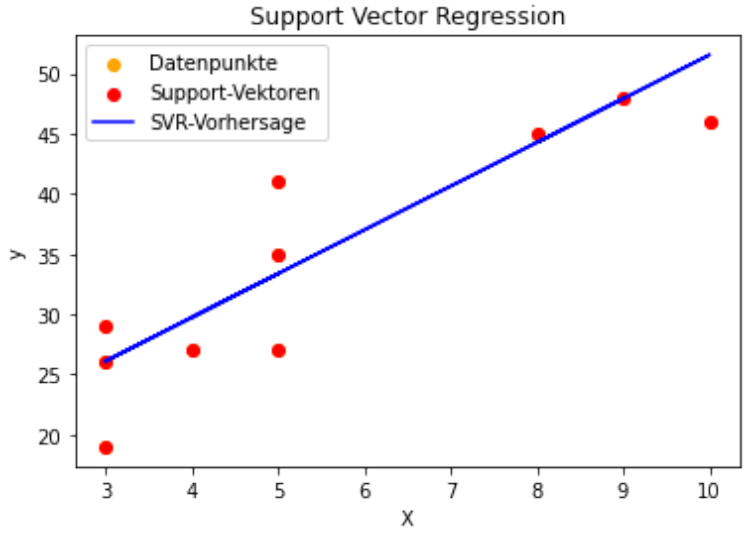
Suport-Vektoren anzeigen und zählen

Alle Datenpunkte sind Support Vektoren

In [5]:

```
# Anzahl der Support-Vektoren
print(f"Anzahl der Support-Vektoren: {len(svr_lin.support_)}")

# Indizes der Support-Vektoren
print("Indizes der Support-Vektoren:")
print(svr_lin.support_)

# print("Support-Vektoren:")
# print(svr_lin.support_)

# Plot mit Support-Vektoren
plt.scatter(x, y, color='orange', label='Datenpunkte')  # Alle Datenpunk
plt.scatter(x[svr_lin.support_], y[svr_lin.support_], color='red', label
plt.plot(x, svr_lin.predict(x), color='blue', label='SVR-Vorhersage')  #
plt.xlabel('X')
plt.ylabel('y')
plt.title('Support Vector Regression')
plt.legend()
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]:

```
print('Lösungen bei Aufgabe (5.6)-Teil2 waren Steigung ~ 3,5 und Achsena
print('Lösungen durch Abmessen der Datenpunkte: Steigung ~ 3,4 und Achse
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
~ 15,2
Lösungen sind also bei beiden Verfahren durchaus vergleichbar!
```

In [7]:

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

```
******* Aktuelles Datum und Zeit*********
Date & Time: 14.12.2024  14:02:19
Ende Python-Programm ***SVR_LinRegression ***
```