mein_zweites_notebook

February 12, 2024

1 Lernen und Lehren mit Jupyter Notebooks

Jupyter Notebooks bieten eine interaktive Umgebung für experimentelles Lernen und die Vermittlung komplexer Konzepte in der Informatik, insbesondere im Bereich Data Science und maschinelles Lernen. Sie ermöglichen die Kombination von erklärungstext, Code, Visualisierungen und weiteren Medien in einem einzigen Dokument.

1.1 Beispiel: Lineare Regression

In diesem Beispiel demonstrieren wir, wie eine einfache lineare Regression mit Python durchgeführt wird. Dieses Konzept ist fundamental in der Statistik und maschinellem Lernen.

[ ]:
```
# Importieren notwendiger Bibliotheken
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

1.1.1 Datensatz generieren

Wir generieren einen einfachen Datensatz, der eine lineare Beziehung zwischen zwei Variablen darstellt.

[ ]:
```
# Generieren von Beispieldaten
x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
y = np.array([5, 20, 14, 32, 22, 38])
```

1.1.2 Modelltraining

Nun trainieren wir ein lineares Regressionsmodell mit den generierten Daten.

[ ]:
```
# Modell initialisieren und trainieren
# Das LinearRegression-Modell aus sklearn wird mit unseren Daten (x, y)␣
 ↪trainiert,
# um die Parameter für die bestmögliche lineare Anpassung zu finden.
model = LinearRegression()
model.fit(x, y)
```

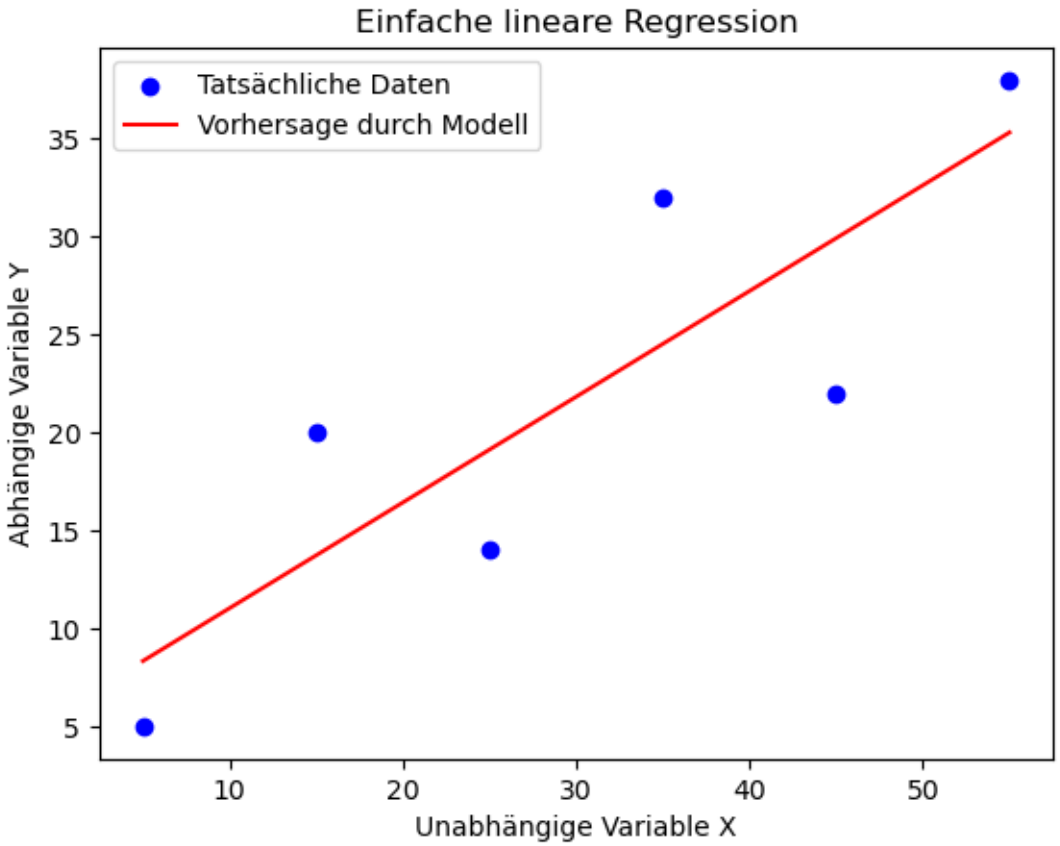
[ ]:
```
LinearRegression()
```

1.1.3 Vorhersage und Visualisierung

Mit dem trainierten Modell können wir Vorhersagen treffen und die Ergebnisse visualisieren.

[ ]:
```
# Vorhersage
y_pred = model.predict(x)

# Visualisierung
plt.scatter(x, y, color='blue', label='Tatsächliche Daten')
plt.plot(x, y_pred, color='red', label='Vorhersage durch Modell')
plt.title('Einfache lineare Regression')
plt.xlabel('Unabhängige Variable X')
plt.ylabel('Abhängige Variable Y')
plt.legend()
plt.show()
```



Dieses einfache Beispiel zeigt, wie Jupyter Notebooks als interaktives Lehrmittel verwendet werden können, um komplexe Themen wie lineare Regression zu vermitteln. Durch die direkte Integration von Code, Ausführungsergebnissen und erklärungstext wird ein umfassendes Lernumfeld geschaffen, das sowohl für Lehrkräfte als auch Studierende von Vorteil ist.

[ ]:
```
from ipywidgets import interact
import numpy as np

def plot_linear_regression(a=20, b=5):
    x = np.linspace(0, 10, 100)
    y = a * x + b
    plt.plot(x, y, '-r', label='y=ax+b')
    plt.title('Interaktive lineare Regression')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.legend(loc='best')
    plt.show()

interact(plot_linear_regression, a=(0,40), b=(0,10))
```

```
interactive(children=(IntSlider(value=20, description='a', max=40),␣
↪IntSlider(value=5, description='b', max=10…
```

[ ]:
```
<function __main__.plot_linear_regression(a=20, b=5)>
```

1.2 Weiterführende Ressourcen

- Linear Regression in Python
- Scikit-Learn Dokumentation
- Deep Learning Book