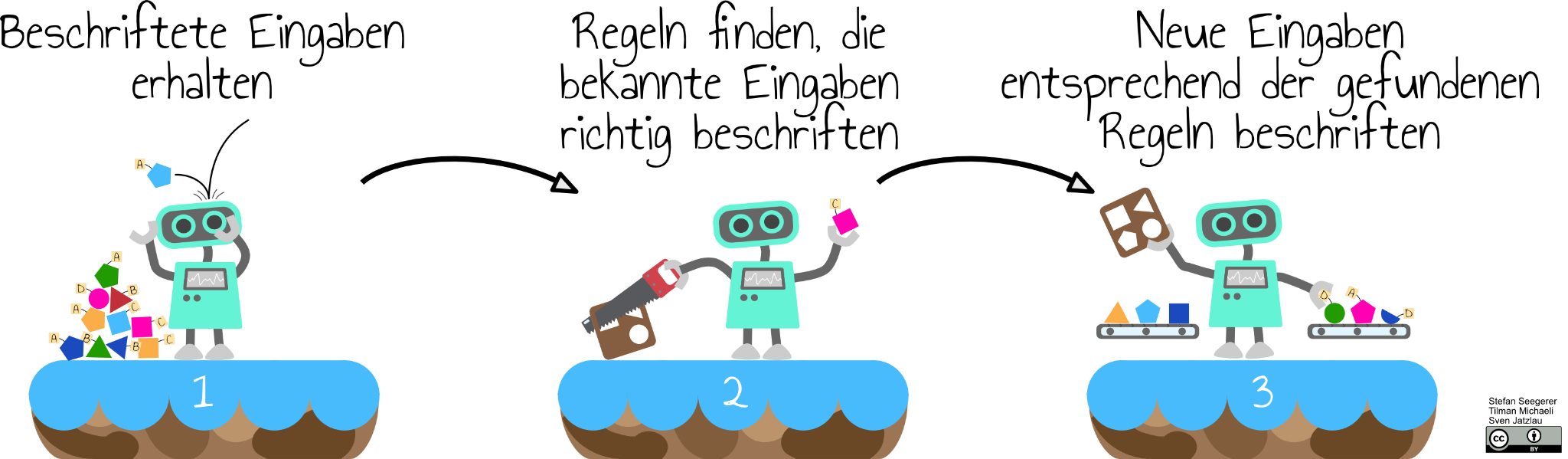
Überwachtes Lernen

Überwachtes Lernen kann verwendet werden, um eine Beschriftung aus einer Reihe von vorgegebenen Beschriftungen wie **beißt** und **beißt nicht** vorherzusagen, es kann aber auch dazu verwendet werden, einen Zahlenwert vorherzusagen, etwa die voraussichtlich anfallenden Schadenssummen bei einer Versicherung oder die Entwicklung von Aktienkursen und Hauspreisen.

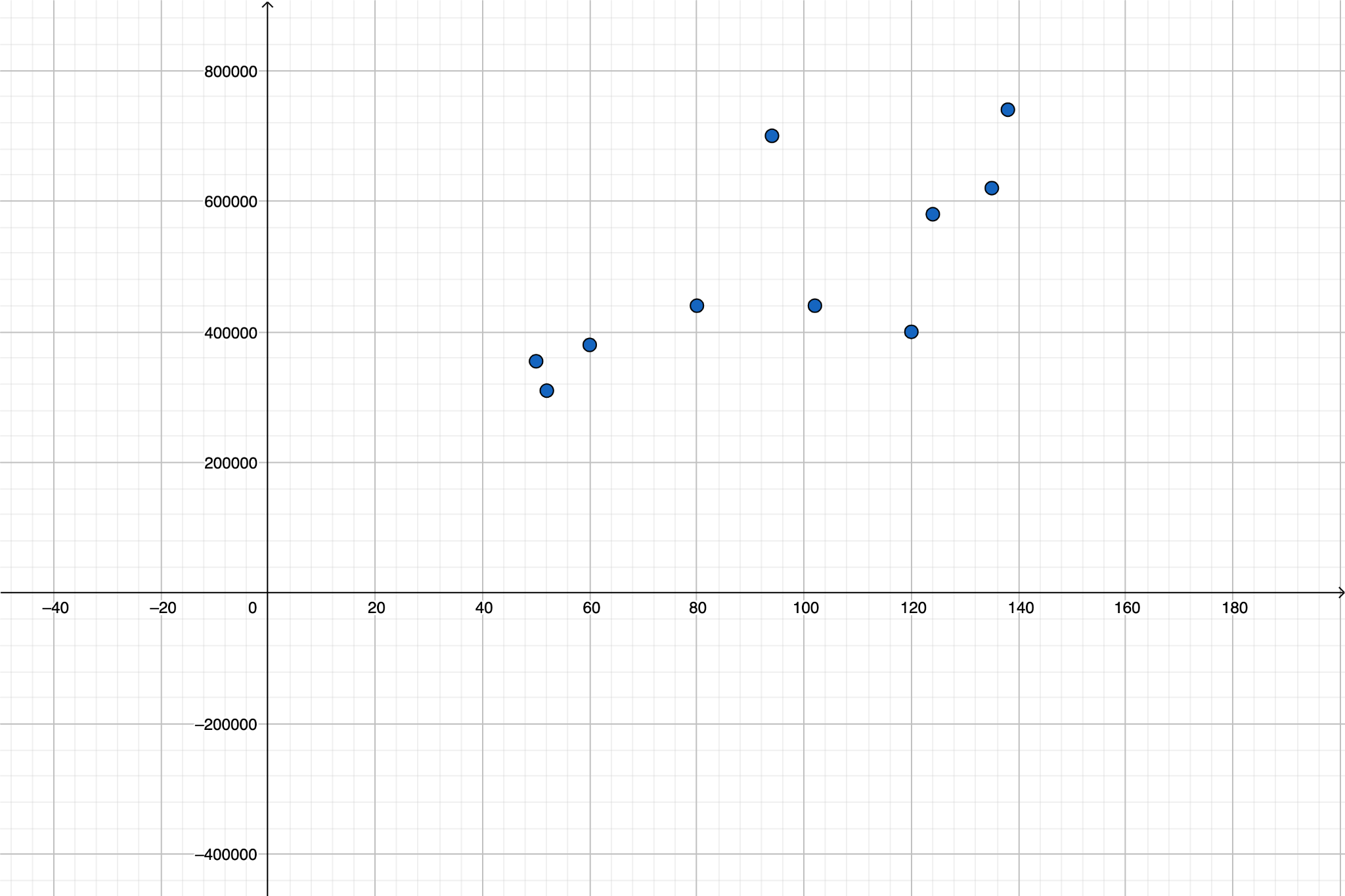
In diesem Arbeitsmaterial setzt du mit linearer Regression ein überwachtes Lernverfahren in Snap*!* um, um Hauspreise abhängig von ihrer Fläche in Quadratmetern vorherzusagen.

## Aufgabe: Lineare Regression

a) In folgender Abbildung sind die Marktpreise und zugehörige Fläche in Quadratmetern für verschiedene Häuser angezeichnet. Zeichne eine Gerade ein, die sich den gegebenen Punkten bestmöglich annähert.

Marktpreis

Fläche in qm



b) Triff nun mithilfe deiner Gerade eine Vorhersage wie hoch der Preis für ein Haus mit 180m² Fläche ist: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| Überwachtes Lernen wird neben Klassifikationsproblemen auch bei **Regressions- problemen** eingesetzt, in denen Eingaben keine Klasse (wie „beißt“ oder „beißt nicht“) sondern ein numerischer Wert (wie etwa ein Hauspreis) zugeordnet werden soll. Der Computer versucht dazu einen Zusammenhang zwischen Eingabedaten (Fläche) und deren Beschriftung (Hauspreis) zu finden. Das Finden einer Gerade bezeichnet man als **lineare Regression**.  Mathematisch wird eine Gerade durch die **Steigung** undden y-Achsenabschnittwie folgt beschrieben:  Aufgabe des Computers ist es also, geeignete Werte für a und b zu finden. Die Parameter a und b stellen für den Computer also das Modell dar, mit deren Hilfe dann auch Vorhersagen für weitere Häuser mit ihrer gegeben Fläche getroffen werden können. |

## Aufgabe: Vorhersage in Snap!

Zunächst benötigen wir einen Block, der für ein bereits existierendes Modell Vorhersagen trifft.

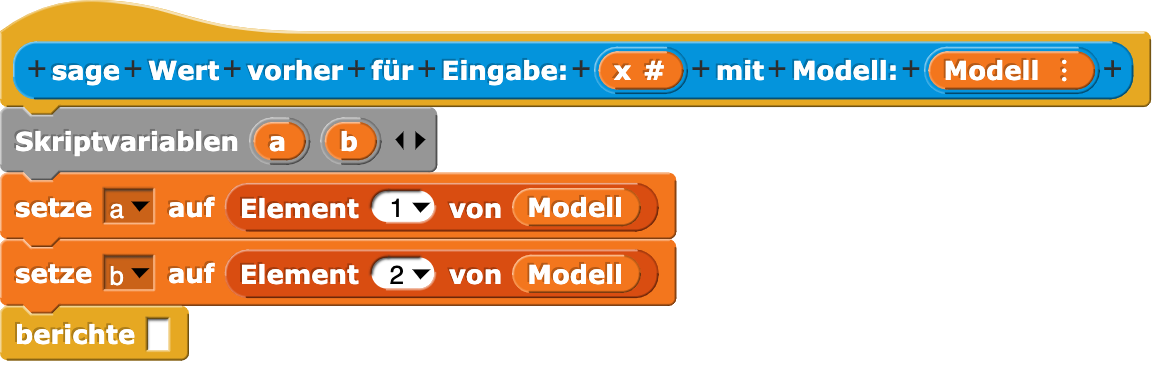
**a)** Öffne die Vorlage des Projekts “Hauspreise”: <https://bit.ly/A1-reg>

Verwende anschließend den , um die Trainingsdaten auf der Bühne anzeigen zu lassen.



**b)** Implementiere den Block,



indem du via Rechtsklick>Bearbeiten… in den Bearbeitungsmodus wechselt und den Wert berechnest und über den berichte Block zurückgibst.

Zur Erinnerung: mit Fläche

Die Vorlage speichert die Parameter a und b in den beiden gleichnamigen Skriptvariablen

**c)** Rufe den gerade implementierten Block mit den folgenden Eingaben auf und überprüfe, ob du jeweils das angegebene Ergebnis erhältst.



## Aufgabe: Lernen

Noch fehlt uns allerdings der wichtigste Schritt eines maschinellen Lernverfahrens: das Lernen des Modells und damit das Finden geeigneter Parameter und . Der Computer löst das aber nicht wie wir vorhin mit “draufgucken”, sondern nimmt eine Gerade und passt sie schrittweise an.

Implementiere nun den Block , indem du via Rechtsklick>Bearbeiten… in den Bearbeitungsmodus wechselst.



Einige vorbereitende Schritte sind bereits erledigt: Die Variable enthält alle Eingabedaten, die Variable die zugehörigen Beschriftungen.



Hinweis: Da wir hier immer dieselben Operationen für alle Eingabedaten und erwarteten Ausgaben durchführen, können wir den Blöcken als Eingaben statt Einzelwerten auch Listen übergeben.

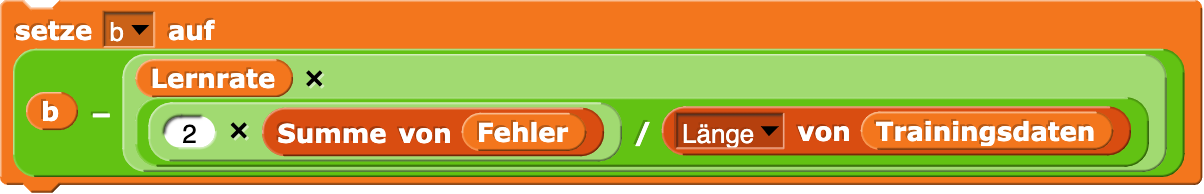
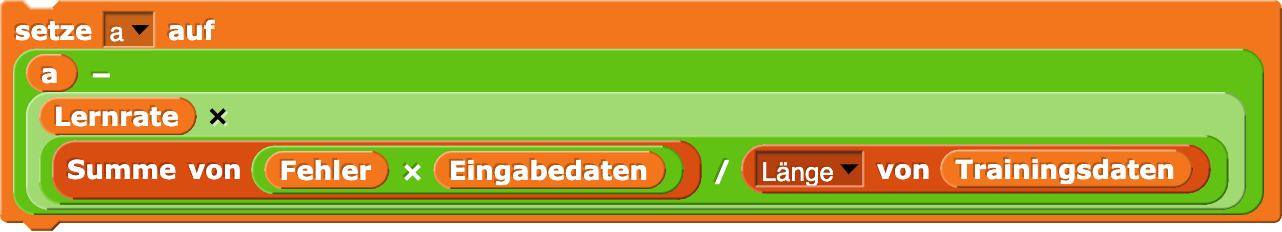
**a)** Rufe zunächst den Block mit und dem aktuellen Modell auf. Speichere das Ergebnis in der Variablen .



**b)** Diese Vorhersagen sollen nun mit den erwarteten Ausgaben verglichen werden. Bestimme dazu den Fehler, indem du die Differenz aus und bestimmst.



**c)** Um diesen Fehler zu reduzieren, muss das Modell und damit die Variablen und entsprechend angepasst werden. Passe dazu und gemäß folgender Formel an:



**d)** Überprüfe die gelernte Gerade visuell, indem du den folgenden Block ausführst:



## Aufgabe: Mehr als nur Quadratmeter

In der Realität hängt der Hauspreis aber natürlich nicht nur von der Fläche, sondern auch von vielen weiteren Faktoren ab. Sammle weitere für den Hauspreis relevante Merkmale:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_