KI für Alle 2: Verstehen, Bewerten, Reflektieren

Themenblock Datenbeschaffung und -aufbereitung:
04_03Aufbereitung_Dimensionsreduktion

Techniken zur Dimensionsreduktion

Erarbeitet von
Dr. Ann-Kathrin Selker

Lernziele

- Du kannst erklären, wieso Dimensionsreduktion bei Daten benötigt wird
- Du kannst verschiedene Verfahren zur Dimensionsreduktion nennen und mit Anleitung durchführen
- Du kannst die Funktionsweise von PCA erklären

Projekt ai4all „KI für Alle"
HeiCAD –Heine Center for Artificial Intelligence and Data Science

Inhalt

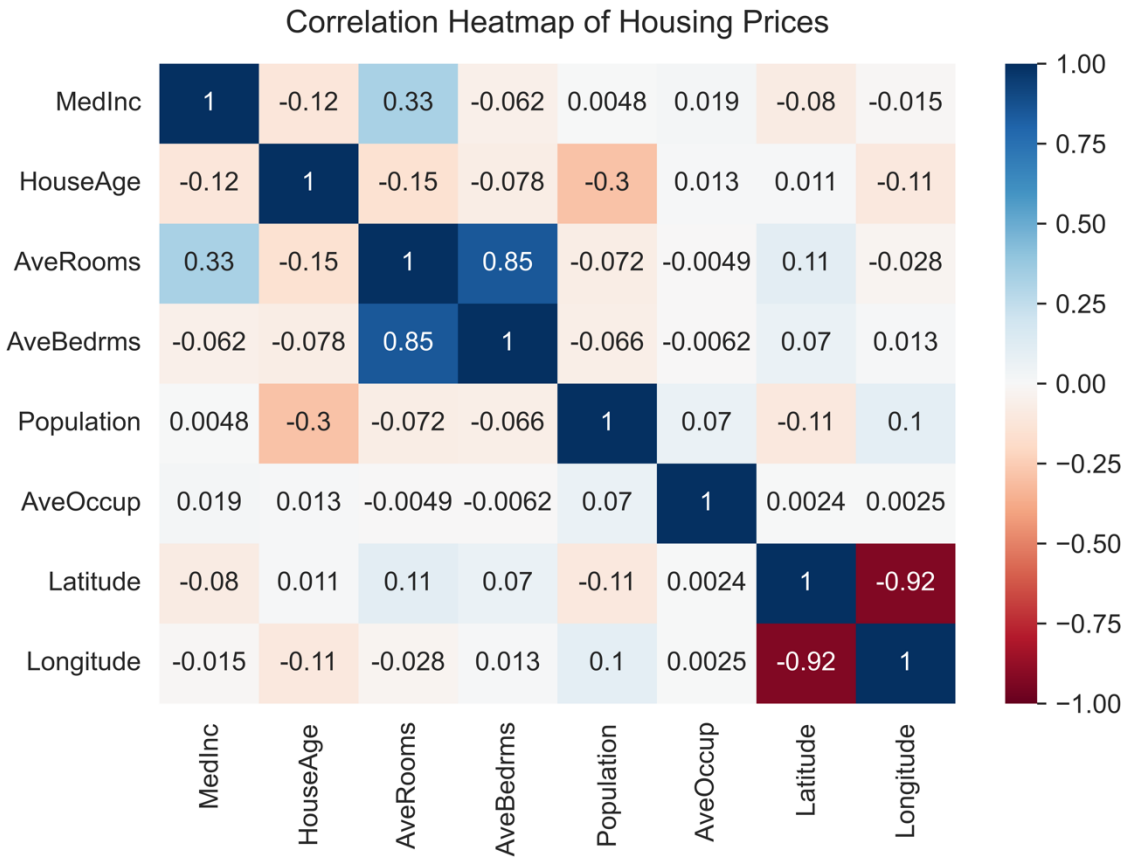
Einstieg

In vielen Aspekten des Machine Learning gilt: Je mehr Daten, desto besser. Je mehr Informationen zur Verfügung stehen, desto besser kann der Lernprozess stattfinden. Aber wann trifft diese Philosophie auf ihre Grenzen, und wie gehen wir damit um?

Zur Erinnerung: Die Anzahl der Features ist gleichzeitig auch die sogenannte Dimension eines Datenpunktes. Der Fluch der Dimensionalität, so ursprünglich benannt von Richard E. Bellman, besagt im Kontext des maschinellen Lernens, dass bei steigender Feature-Anzahl die Anzahl der benötigten Trainingsdaten exponentiell wächst. Um dies zu umgehen, sollten wir also im Rahmen des Feature Engineerings die Dimension unserer Daten verringern. Dimensionsreduktion führt zu einem teilweise erheblichen Geschwindigkeitsgewinn beim Training eines Machine-Learning-Modells. Dies gilt insbesondere für tiefe neuronale Netze. Außerdem können viele Machine Learning Verfahren nicht gut mit irrelevanten Features umgehen. Daher verbessert es in diesen Fällen die Performance, diese Features komplett zu entfernen. Dimensionsreduktion führt zwar manchmal zu einem Verlust der Accuracy, dies wird aber für die höhere Trainingsgeschwindigkeit und Vereinfachung der Daten in Kauf genommen. In diesem Video schauen wir uns einmal verschiedene Techniken zur Dimensionsreduktion an.

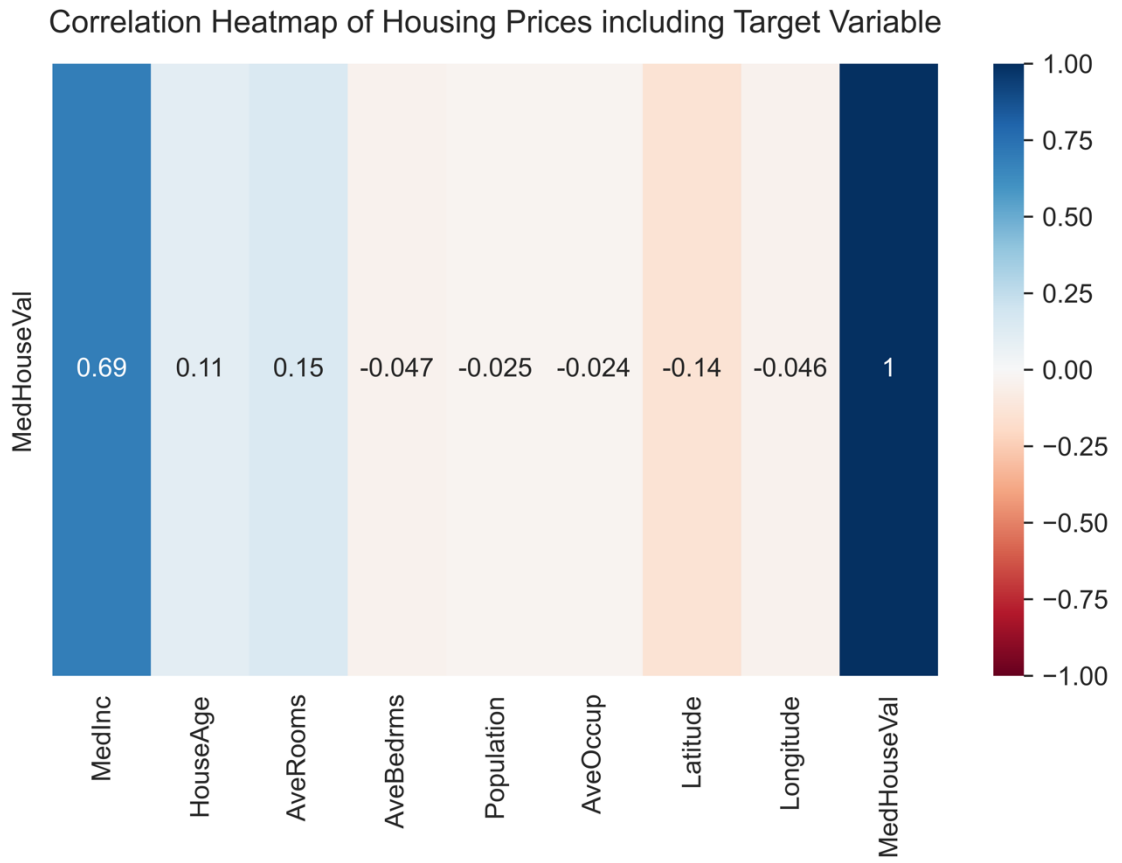
Korrelationsanalyse

Eine Korrelationsanalyse kann helfen, korrelierte Features zu erkennen. Betrachten wir noch einmal die Korrelationsmatrix des California Housing Datensatzes, erstellt mithilfe einer Pearson-Korrelationsanalyse.



Die Features Längengrad (Longitude) und Breitengrad (Latitude) sind stark korreliert. Das lässt darauf schließen, dass sie beide ähnliche Informationen enthalten und eines der beiden Features dadurch redundant ist. Das Entfernen eines der beiden Features führt also zu einer Verringerung der Information, ohne dass viele Informationen verloren gehen. Am besten entfernen wir das Feature, was in der folgenden Analyse schlechter abschneidet.

Als nächstes betrachten wir den Zusammenhang zwischen der Zielvariable und unseren Features und finden so heraus, welche Features überhaupt relevante Informationen für die Zielvariable beinhalten. In unserem Beispiel sehen die Korrelationskoeffizienten so aus:



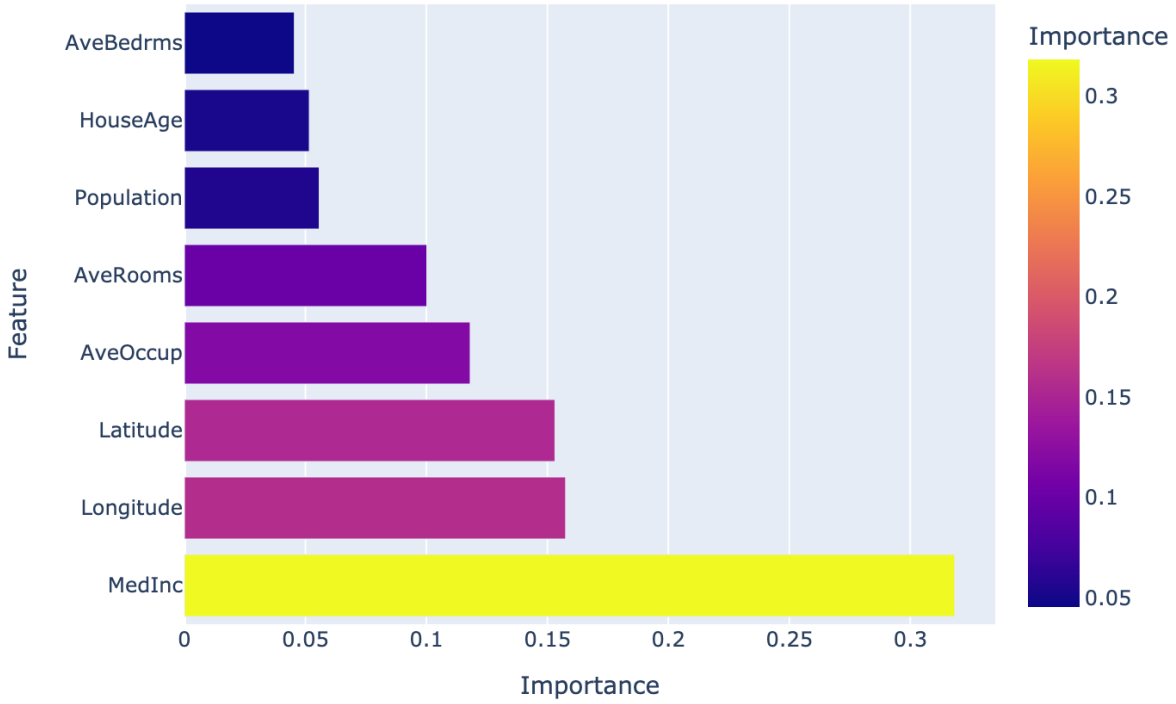
Es fällt auf, dass das Medianeinkommen der Gegend mit dem Median-Hauswert korreliert und es sich daher wohl um ein wichtiges Feature handelt. Dahingegen ist der Korrelationskoeffizient bei Features wie Einwohnerzahl der Gegend und Längengrad des Hauses im Grunde bei 0 und daher nicht zusammenhängend mit der Zielvariablen. Hier kann es sich also um gute Kandidaten zum Entfernen handeln.

Achtung: In unserem Beispiel haben wir aus Gründen der Einfachheit und Übersichtlichkeit natürlich nur auf lineare Zusammenhänge und nur auf Zusammenhänge zwischen zwei Variablen geachtet. Daher kann es sein, dass wir bedeutende Zusammenhänge übersehen haben. Außerdem gilt wie immer: Die gefundenen Korrelationen müssen keine Kausalität implizieren!

Entscheidungsbäume

Ein ähnliches Verfahren kann bei Klassifikationsproblemen helfen: Wir trainieren auf unseren Daten einen Entscheidungsbaum. Die Features, anhand derer die Klassenzugehörigkeit eines Datenpunktes zu einer Klasse entschieden wird, sind anscheinend die Features, die relevant sind. Alle anderen Features liefern nur wenige Informationen und können entfernt werden. Das Python-Modul scikit-learn speichert die Wichtigkeiten der Features in einer Variable namens feature_importances_, sodass wir uns diese ausgeben können.

Zuerst diskretisieren wir die Zielvariable „Hauswert" des California Housing Datensatzes, um mit ihm Klassifikationsalgorithmen durchführen zu können. Als nächstes trainieren wir auf dem modifizierten Datensatz einen Entscheidungsbaum.



Wie im Bild zu sehen ist, wird auch hier das Medianeinkommen der Gegend als Feature mit dem meisten Informationsgehalt identifiziert, wohingegen z. B. das Alter der Häuser erstaunlich wenig Einfluss auf den Hauswert hat und in diesem Fall ein guter Kandidat für das Entfernen von Features ist. Enthält der Datensatz deutlich mehr als die acht Features aus California Housing, wird es auch viele Features geben, die nicht im Baum vorhanden sind. In dem Fall wären dies die Features, die entfernt werden können.

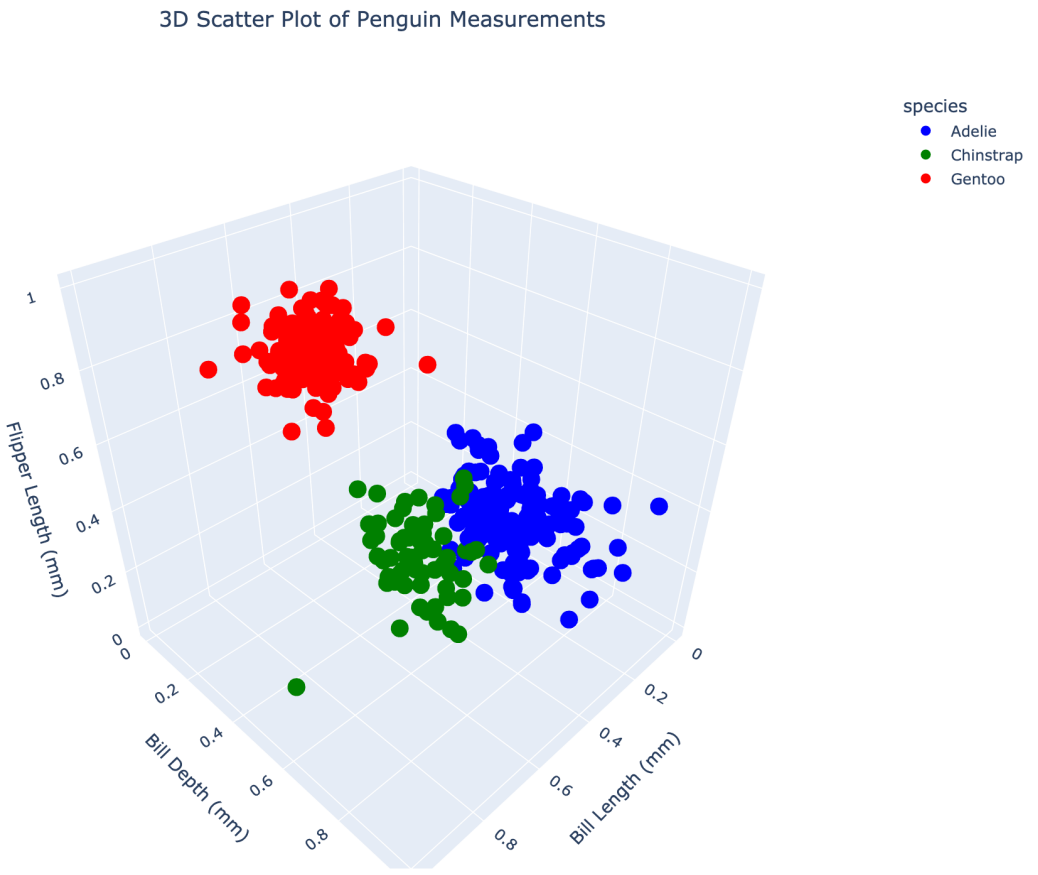
Principal Component Analysis (PCA)

Bisher haben wir uns nur angesehen, wie wir komplette Features entfernen können. Aber es ist auch möglich, durch Kombinieren von Features die Komplexität zu reduzieren. Ein Beispiel dafür ist die sogenannte Principal Component Analysis, kurz PCA. PCA ist ein Verfahren zur Dimensionsreduktion, das stark auf mathematischen Konzepten beruht, die normalerweise nur aus einem Studium der Mathematik oder vergleichbaren Studiengängen bekannt sind. Daher gehe ich hier nicht genau darauf ein, wie und wieso das Verfahren im Detail funktioniert.

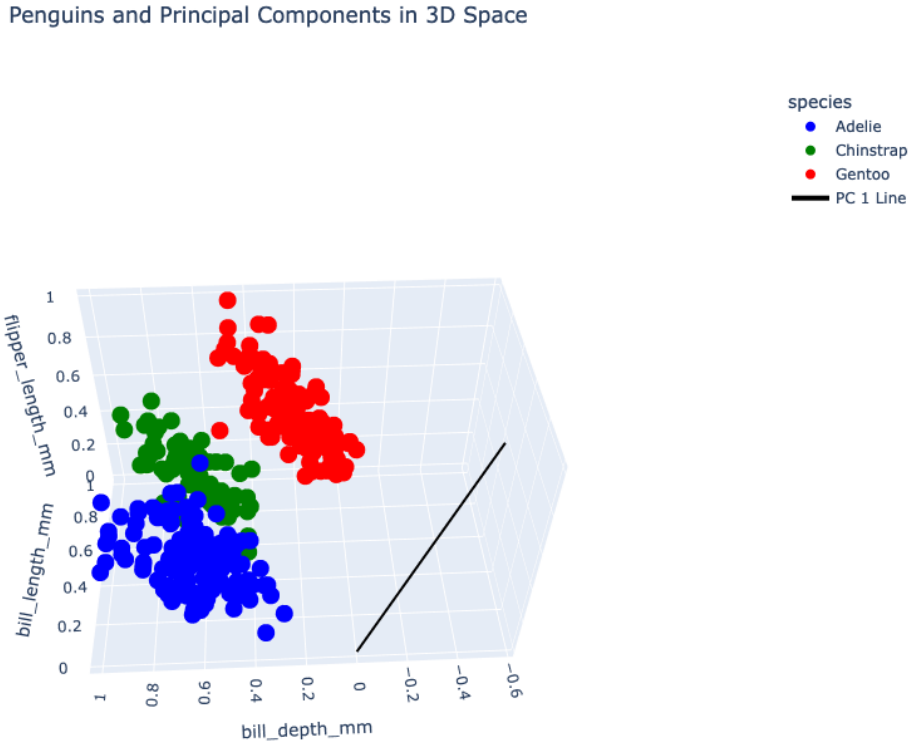
Bei PCA handelt es sich selber bereits um einen Machine Learning Algorithmus. Es funktioniert dabei nach dem Prinzip des unüberwachten Lernens, d. h. es erhält ungelabelte

Daten und findet Muster in den Daten. PCA nimmt die vorhandenen Features und erstellt daraus neue Features. Dabei werden die „Hauptbestandteile" der alten Features extrahiert und in neuen Features zusammengefasst. Daher kommt auch der englische Name Principal Components Analysis. Die Anzahl der Features wird bei diesem Vorgang nicht nur drastisch verringert, es geht auch kaum Information verloren. Die entstehenden neuen Features sind dabei nach Wichtigkeit sortiert. Das neue erste Feature enthält also die meisten Informationen über unsere Daten, das neue zweite die zweitmeisten usw.

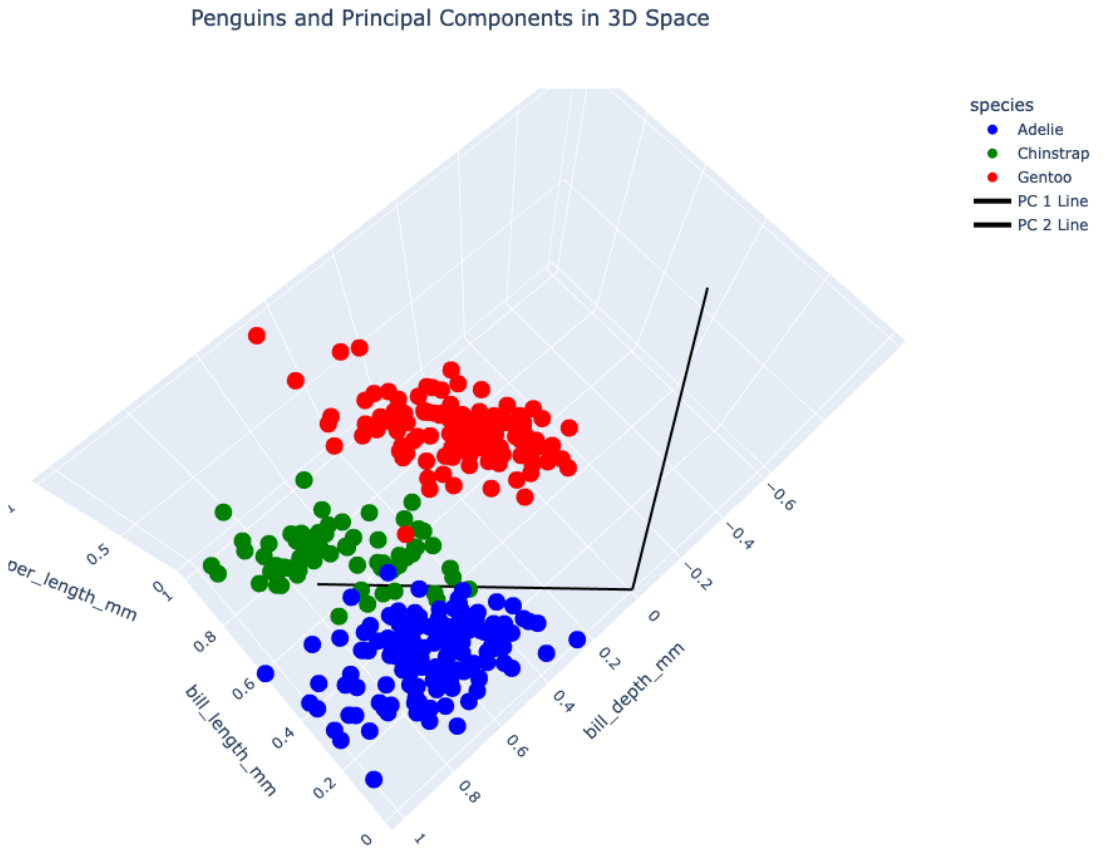
Genauer gesagt findet PCA Vektoren mit gewissen Eigenschaften, zu denen unter anderem der Abstand zu den Datenpunkten zählt. Features, die für eine geringe Streuung der Ergebnisse sorgen, sind weniger wichtig als Features, die für eine große Streuung der Ergebnisse sorgen, und tragen daher auch weniger zu den Principal Components bei. Daher ist es auch so wichtig, vor dem Benutzen von PCA deine Daten zu normalisieren, da ansonsten die Streuung nicht zu vergleichen ist. Betrachte als Beispiel den Datensatz Palmer Penguins. Hierbei handelt es sich bereits um einen Datensatz mit extrem wenigen Features, wir verwenden ihn aber trotzdem, um uns die Visualisierung zu erleichtern. Genauer gesagt schauen wir uns nur die Features Schnabellänge, Schnabeltiefe und Flossenlänge an. Die Farben entsprechen den jeweiligen Pinguinarten.



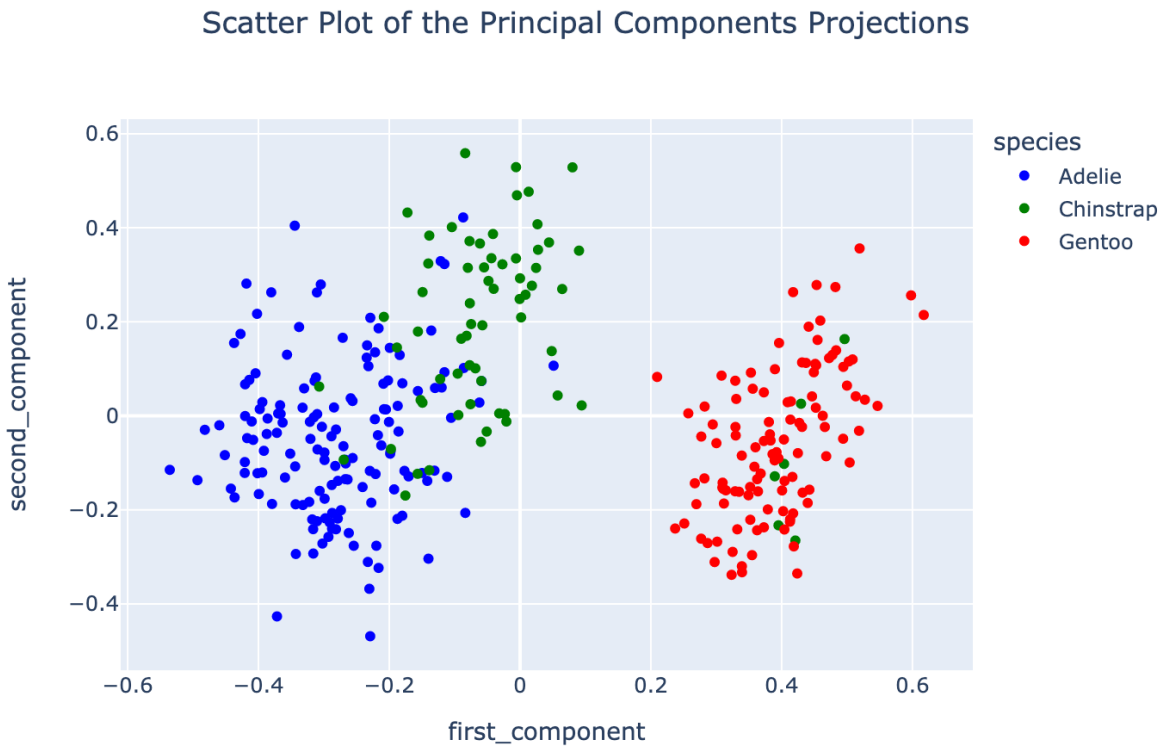
Der erste Vektor, den PCA findet, ist diese Linie. Wir sehen, dass es in dieser Richtung tatsächlich eine große Streuung der Ergebnisse gibt.



Jetzt brauchen wir uns nur noch um die Streuung in den anderen Richtungen zu kümmern. Der zweite gefundene Vektor sieht so aus.



Mithilfe dieser zwei Vektoren ist es jetzt möglich, die Datenpunkte mathematisch so zu transformieren, dass sie zweidimensional werden und damit ihre Dimension um eins verringert wird. Wie gesagt, du musst nicht wissen, wie diese Transformation genau abläuft. Das Wichtige: Unsere Daten sind jetzt zweidimensional, Mission erfolgreich!



PCA funktioniert am besten, wenn einige Bedingungen erfüllt sind. Dazu gehört neben normalisierten Daten auch das vorherige Behandeln von Ausreißern, da PCA sehr anfällig für Ausreißer ist. Weiterhin müssen Korrelationen in den Daten vorhanden sein, da PCA unter anderem die Korrelationsmatrix der Daten benutzt. In Python ist PCA zum Beispiel bereits im Modul scikit-learn implementiert und kann von dort aus aufgerufen werden.

Abschluss

Ein guter Nebeneffekt der Dimensionsreduktion ist es auch, dass Visualisierungen einfacher und vor allem leichter verständlich werden. Da z. B. PCA die Eigenschaft hat, dass die entstehenden Features nach Wichtigkeit sortiert sind, ist es auch gut möglich, nur die ersten zwei oder drei Features zu plotten und trotzdem einen guten Überblick über die Daten zu erhalten, selbst wenn die Daten mehr als diese Hauptkomponenten haben.

Dieses Video hat dir einen kleinen Ausschnitt an Dimensionsreduktionstechniken präsentiert.

Weiterführendes Material

https://www.geeksforgeeks.org/principal-component-analysis-pca/

Udacity: Curse of Dimensionality – Georgia Tech – Machine Learning
https://www.youtube.com/watch?v=QZ0DtNFdDko
https://www.youtube.com/watch?v=OyPcbeiwps8

Disclaimer