Machine Learning 2
Refugeeks / intoCODE
2022

Torben Laue, M.Sc.
Florian Kluger, M.Sc.

Anwendungsbeispiel Kundendaten

1 Aufgabe

Die Kundensegmentierung ist für Unternehmen wichtig, um ihr Zielpublikum zu verstehen. Je nach demografischem Profil, Interessen und Wohlstandsniveau können verschiedene Werbemittel zusammengestellt und an unterschiedliche Zielgruppen gesendet werden.

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

Abbildung 1: Kundendaten

Es stehen die Kundendaten zur Verfügung, wie sie in der Abbildung 1 aufgelistet sind. Benutzen Sie für diese Aufgabe die Datei *customer_clustering.py*.

1.1 Skalierung

(a) Berechnen Sie für alle Features den Mittelwert und die Standardabweichung. Geben Sie außerdem jeweils den kleinsten und größten Wert aus.

(b) Berechnen Sie mithilfe von numpy die Kovarianzmatrix der Features und interpretieren Sie diese.

(c) Führen Sie eine Standardskalierung auf den Daten durch. Warum ist dies sinnvoll?

(d) Berechnen Sie erneut die Kovarianzmatrix.

1.2 Clustering: K-Means

(a) Importieren Sie die KMeans-Klasse und erstellen Sie ein Objekt davon.

(b) Erstellen Sie einen Ellbogen-Plot. Berechnen Sie für jede Clusteranzahl auch den Silhouette-Score.

(c) Entscheiden Sie sich für eine Anzahl an Cluster. Erstellen Sie einen 3D *scatter*-Plot um das Clusteringergebnis grafisch darzustellen.

(d) Berechnen Sie den Mittelwert und die Standardabweichung je Cluster.

(e) Welche Gemeinsamkeiten haben die Kunden innerhalb der jeweiligen Cluster?

1.3 Clustering: DBSCAN

(a) Importieren Sie die DBSCAN-Klasse und erstellen Sie ein Objekt davon.

(b) Bestimmen Sie mithilfe des Silhouette-Score die optimalen Werte für *eps* und *min_samples*. Nehmen Sie dabei an, dass maximal 30% der Daten Ausreißer sind.

(c) Welcher Algorithmus eignet sich besser, um den vorliegenden Datensatz zu Clustern: K-Means oder DBSCAN?