

1 Begriffsdefinitionen

1.1 Klassifikation - Clustering

Klassifikation ist eine Methode um verschiedene Messungen einer jeweiligen Klasse zuzuordnen. Das ist eine Aufgabe für supervised Learning, d.h. wir haben Testbeispiele mit Daten und Klassenzuordnung die wir trainieren und daraus abstrakte Aussagen über Klassenzugehörigkeit treffen wollen, also Algorithmen trainieren die uns für beliebige Daten möglichst die korrekte Klasse liefern.

Clustering ist eine Methode um aus einer Menge aus Daten ähnliche Gruppen von Punkten zu extrahieren. Das ist eine Aufgabe für unsupervised Learning, d.h. wir haben nur die Daten ohne irgendeine Zuordnung. Das ist der eine große Unterschied zwischen Clustering und Klassifikation, dass wir Punkte nicht in Klassen einordnen sondern nur aufgrund von Strukturen in den Daten Cluster bilden. Außerdem ist es beim Clustering kein Problem einem Punkt mehreren Clustern zuzuordnen, so z.B. beim hierarchischem Clustering. Bei der Klassifikation wollen wir immer nur eine Klasse herausfinden für einen Datenpunkt.

1.2 Kernobjekt - Dichteerreichbarkeit - Dichteverbunden

1.2.1 Kernobjekt

Bei DBSCAN hat man für den Clustering-Algorithmus zwei Parameter - den Radius, welcher definiert was eine Nachbarschaft eines Punktes ist, sowie eine Anzahl an Nachbarn, also eine Anzahl von Punkten die mindestens in der Nachbarschaft eines Punktes sind damit dieser als dicht gilt. Hat ein Punkt mindestens die Anzahl der Nachbarn in seiner Nachbarschaft, so gilt er als dicht und somit als Kernobjekt.

1.2.2 Dichteerreichbar

Wie wir vorher festgestellt haben gibt es bei DBSCAN Kernobjekte. Ein Punkt gilt als dichteerreichbar falls er in der Nachbarschaft eines Kernobjekts liegt, aber selber kein Kernobjekt ist, also zuwenig Nachbarn hat.

1.2.3 Dichteverbunden

Zwei Punkte A und B sind dichteverbunden, wenn es einen Pfad bestehend aus Kernobjekten zwischen ihnen gibt. Das heißt dass Punkte A von einem Kernobjekt dichteerreichbar sein muss, welches wiederum mit anderen Kernobjekten verbunden ist, bis wir beim Punkt B sind. Alle Punkte die untereinander dichteverbunden sind bilden einen Cluster.

1.3 Lazy learning

Lazy learning bedeutet dass wir kein Modell unserer Trainingsdaten erstellen und dieses dann auf Testdaten anwenden, sondern erst ein Modell erstellen sobald wir eine Testaufgabe bekommen. Ein Beispiel welches wir schon hatten war k-Means, wo die Centroide und Cluster erst angepasst wurden als wir einen

Punkt einem Cluster zuordnen mussten, und nicht Cluster mit allen Punkten erstellt wurden und diese dann angepasst. Lazy learning beschreibt ein Paradigma in Machine Learning, hat also nicht unbedingt mit Clustering und Klassifikation zu tun.

2 DBSCAN

2.1 k-Distanzdiagramm

Für die initiale Berechnung von k und $MinPts$ brauchen wir als Parameter die Dimensionalität unserer Daten. In unserem Fall geht es um Punkte auf der zweidimensionalen Ebene (Punkte auf der Karte), darum ist unsere Dimensionalität 2, bzw. $d = 2$. Die Gleichungen für die Anfangswerte sind dann:

$$k = (2 * d) - 1 = 4 - 1 = 3$$
$$MinPts = k + 1 = 3 + 1 = 4$$

2.1.1 Graphen

Im folgenden die Graphen für die gegebene Anzahl von k :

Figure 1: $k = 1$

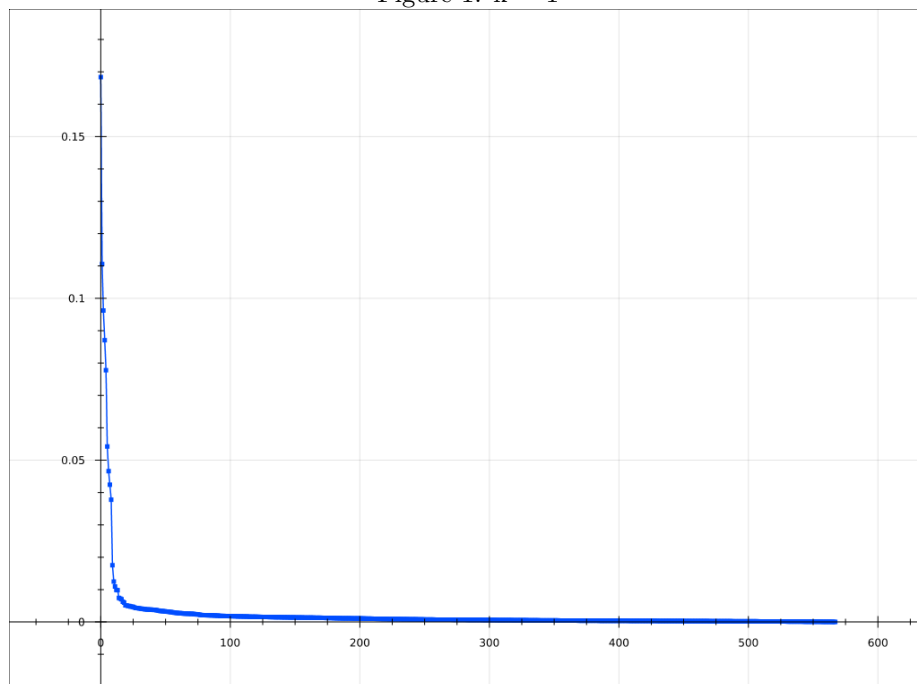


Figure 2: $k = 2$

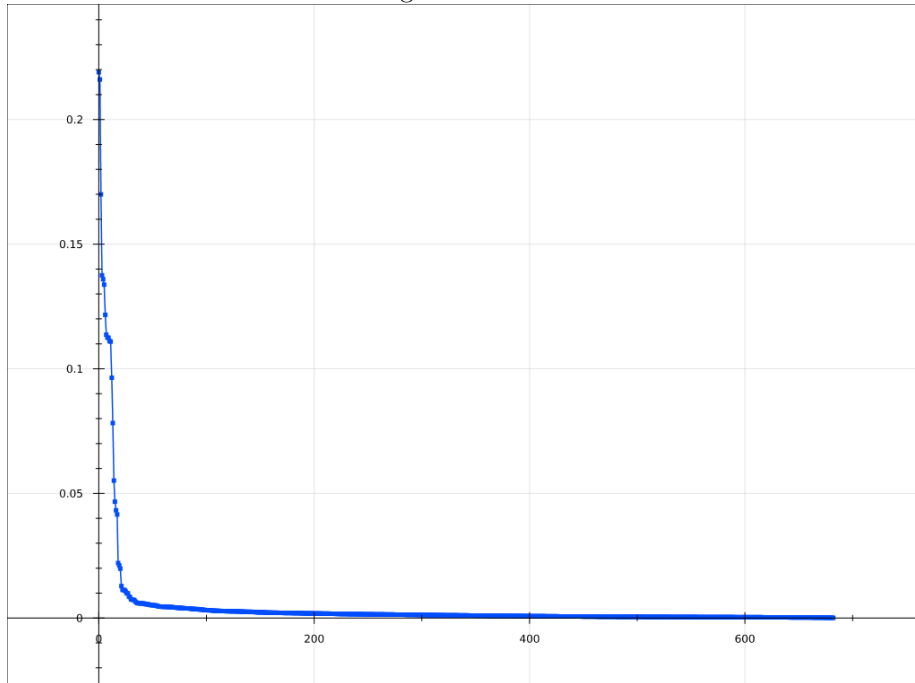


Figure 3: $k = 3$

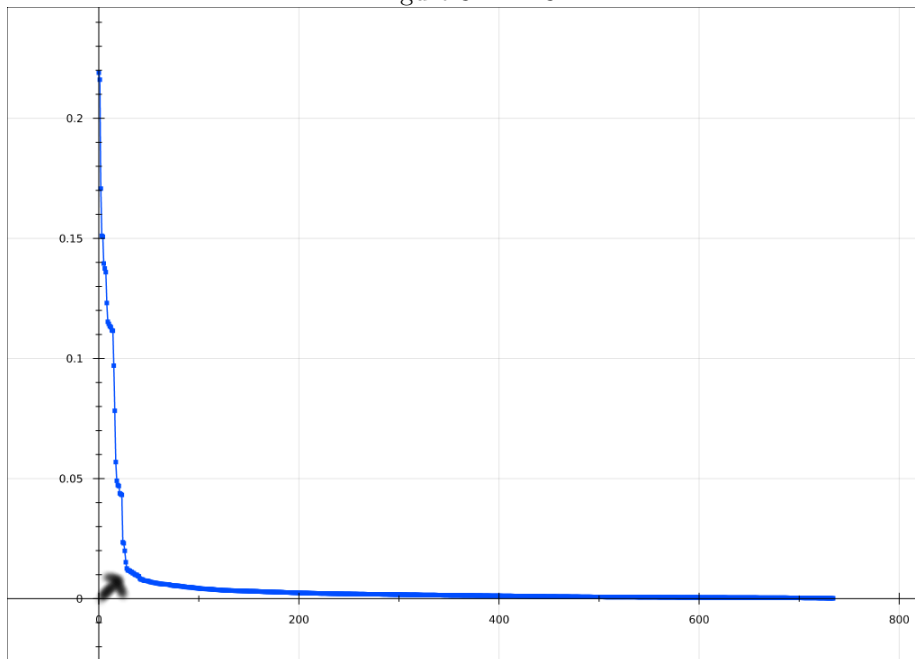


Figure 4: $k = 4$

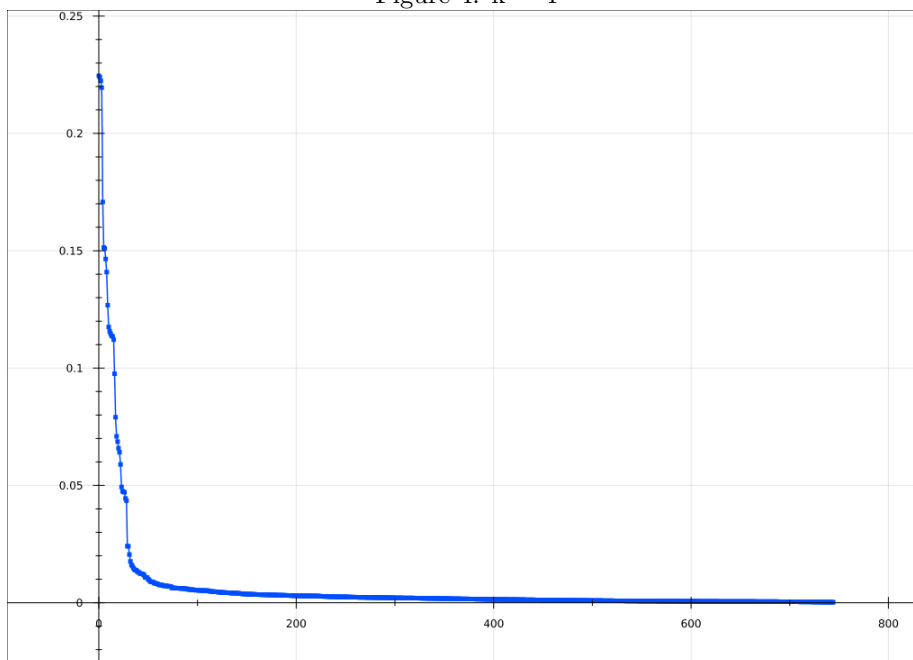


Figure 5: $k = 5$

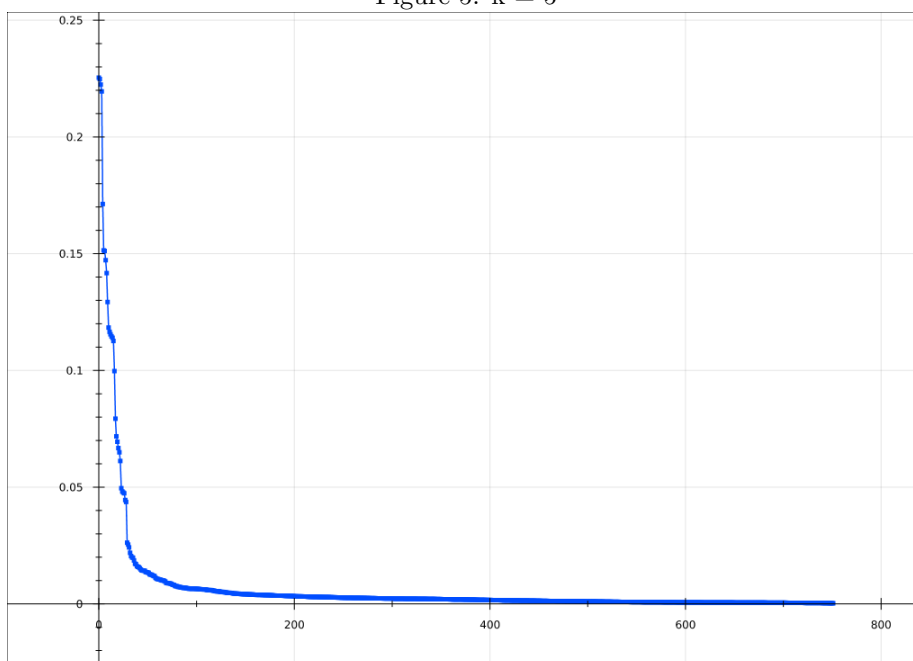
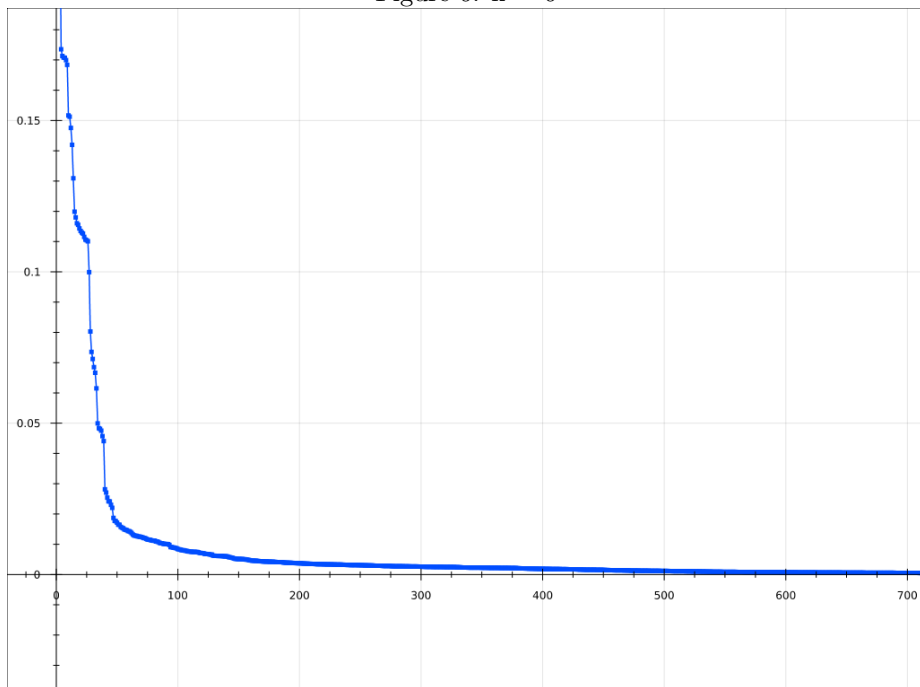


Figure 6: $k = 6$



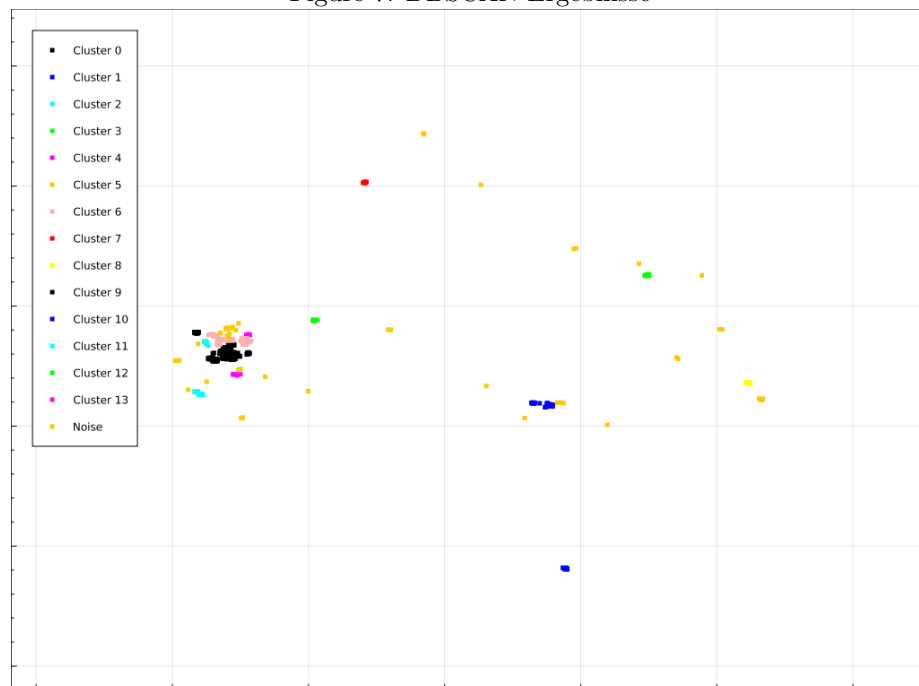
2.1.2 Auswahl Parameter

Wie im Graphen ?? markiert, wählen wir das Grenzobjekt mit den Werten $\text{MinPts} = 4$ und $\text{Epsilon} = 0.012563822348311689$ aus, wobei wir Epsilon der Einfachheit halber auf 0.012 abrunden.

2.2 DBSCAN

Mit den vorhin bestimmten Parametern wurde der DBSCAN-Algorithmus auf unsere Daten angewandt. Dabei kam folgendes Clustering zustande:

Figure 7: DBSCAN-Ergebnisse



Die verlangten Kennziffern sind im folgenden aufgelistet:

KENNZAHLEN

Anzahl Cluster: 14

Kosten fuer el Jefe: 50.861321

Anzahl der Noise-Punkte: 43

Silhouettenkoeffizient: 0.702970

2.3 Vergleich DBSCAN und kMeans

Um die beiden Algorithmen vergleichen zu können, im folgenden noch einmal die Ergebnisse der beiden Verfahren als Plot:

2.3.1 kMeans Clustering

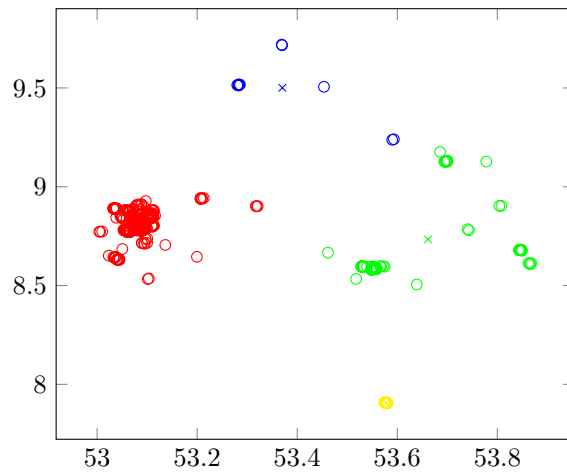
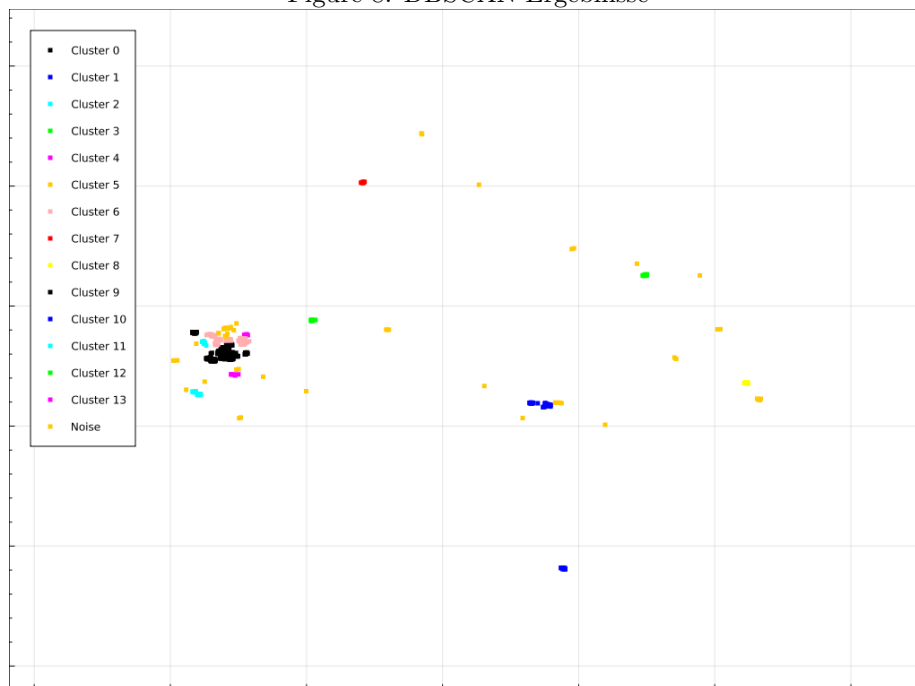


Figure 8: DBSCAN-Ergebnisse



Zum Vergleich: Die Kosten in Gummimünzen sind beim Dichtecustering viel höher, und es werden mehr Fahrer benötigt. Das ist natürlich für den Chef ein Nachteil, weswegen dieses Clustering wahrscheinlich nicht gewählt werden wird. Für die Kunden ist außerdem von Nachteil, dass beim DBSCAN Clustering manche von ihnen nicht mehr angefahren werden, da manche Punkte Noise sind und damit keinem Cluster zufallen. Dafür haben sie allerdings den Vorteil, schneller zu beliefert werden, da ein Fahrer weniger von ihnen betreut.