kNN Cheatsheet

Onur Yilmaz

June 1, 2023

Instanzbasierte Klassifikation

Instanzbasierte Klassifikation ist eine Methode, bei der die Klassenzugehörigkeit eines Testdatums basierend auf den Klassenzugehörigkeiten der Trainingsdaten bestimmt wird.

K-Nearest-Neighbor (KNN) Algorithmus

Der K-Nearest-Neighbor-Algorithmus ist ein instanzbasierter Klassifikationsalgorithmus, bei dem die Klassenzugehörigkeit eines Testdatums anhand der Klassenzugehörigkeiten der K nächstgelegenen Trainingsdaten bestimmt wird.

K festlegen

Um K festzulegen, müssen wir die optimale Anzahl von Nachbarn auswählen. Dies kann durch verschiedene Methoden wie Kreuzvalidierung oder Grid Search erfolgen.

Beispiele in Python

Hier sind Beispiele für die Implementierung des KNN-Algorithmus in Python:

```
from sklearn.neighbors import KNeighborsClassifier

X_train, y_train = load_train_data()
X_test, y_test = load_test_data()


knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)


accuracy = knn.score(X_test, y_test)
```

Standardisierung und Distanzmaße

Standardisierung ist ein Verfahren, bei dem die Merkmale auf eine gemeinsame Skala gebracht werden, um eine bessere Vergleichbarkeit zu gewährleisten. Beim KNN-Algorithmus können verschiedene Distanzmaße verwendet werden, um die Ähnlichkeit zwischen Datenpunkten zu berechnen.

Euklidische Abstand

Der euklidische Abstand ist die gebräuchlichste Distanzmetrik und wird verwendet, um die Entfernung zwischen zwei Punkten im Raum zu messen.

```
1 import numpy as np
2
3 def euclidean_distance(x1, x2):
4         return np.sqrt(np.sum((x1 - x2)**2))
```

Manhattan-Distanz

Die Manhattan-Distanz, auch bekannt als Taxicab-Distanz oder L1-Distanz, misst die Summe der absoluten Differenzen zwischen den Koordinaten der Punkte.

```
1 import numpy as np
2
3 def manhattan_distance(x1, x2):
4         return np.sum(np.abs(x1 - x2))
```