Methods & Algorithms FS 2023

Aufgabenblatt 2: *k*-Nächste-Nachbarn und *k*-Mitten

Die Bearbeitung der Aufgaben ist freiwillig; es erfolgt keine Bewertung.

Aufgabe 1

a) Gegeben ist folgende Tabelle mit $n = 12$ Werten der tatsächlichen Klassifikation y und der durch k-Nächste-Nachbarn vorhergesagten Klassifikation $\hat{y}$ eines Datensatzes (Klassen 0 und 1). Bestimmen Sie die Treffsicherheit des Algorithmus.

i	1	2	3	4	5	6	7	8	9	10	11	12
y_i	1	1	1	1	1	1	0	1	0	1	1	0
$\hat{y}_i$	1	1	1	1	0	1	1	0	0	1	0	0

b) Wie gross ist demnach die Fehlinterpretationsrate (Hamming-Verlust)?

Aufgabe 2

Die Datei „`wine.dat`“ in Moodle enthält drei Spalten des zugehörigen Datensatzes aus dem Paket `sklearn.datasets`. Die ersten beiden Spalten entsprechen 2D-Koordinaten (x und y), die dritte Spalte entspricht der tatsächlichen Klassifikation der Punkte.

a) Visualisieren Sie die Datei als Streudiagramm (Scatter Plot).

b) Ermitteln Sie anhand der Ellenbogen-Methode den optimalen k-Wert für eine Clusterbildung gemäss k-Mitten-Algorithmus und zeichnen Sie die zugehörige (Trägheits)Kurve.

c) Bestimmen Sie für den aus b) ermittelten optimalen k-Wert eine Vorhersage (`fit_predict`) der Klassenzugehörigkeit aller Punkte der Datei „`wine.dat`“ und visualisieren Sie das Ergebnis als Streudiagramm. Wie gut stimmt die vorhergesagte Klassifikation – rein optisch betrachtet – mit dem tatsächlichen Ergebnis aus a) überein?

Aufgabe 3 (für Experimentierfreudige)

In der Vorlesung wurde für die Datei „`smp_data.dat`“ (x-y-Punktkoordinaten) mittels linearer Regression die zugehörige Regressionsgerade bestimmt. Das Paket `sklearn.neighbors` bietet hier die Möglichkeit, mittels `KNeighborsRegressor` eine Regression basierend auf dem bekannten k-Nächste-Nachbarn-Algorithmus durchzuführen. Dabei wird das zugehörige

Modell mit den *x*-Punktkoordinaten als Daten und den *y*-Punktkoordinaten als Klassifikation trainiert.

```
data = np.loadtxt("smp_data.dat", delimiter=",")
x_data = data[:,0].reshape((-1,1))
y_data = data[:,1]
from sklearn.neighbors import KNeighborsRegressor as knr
model = knr()
model.fit(x_data, y_data)
```

a) Bestimmen Sie den optimalen *k*-Wert der Regression gemäss **`KNeighborsRegressor`** für die Daten aus der Datei „**`smp_data.dat`**“. Beachten Sie, dass es sich um eine Regression und nicht um eine Klassifizierung handelt, d.h. die Qualität der Vorhersage durch **`KNeighborsRegressor.predict()`** muss mittels R^2-Wert bestimmt werden.

b) Erzeugen Sie sich eine Sequenz aus 50 *x*-Punktkoordinaten im Bereich [1, 11] mittels

```
x = np.linspace(1,11,50)
```

Aufgabe b.) nicht relevant

und nutzen Sie diese Sequenz für die Vorhersage durch **`KNeighborsRegressor`** unter Verwendung des in Aufgabe a) ermittelten, optimalen Wertes für *k*.

```
model = knr( --Ihr-k-Wert-- )
model.fit(x_data, y_data)
y = model.predict(x.reshape((-1,1)))
```

Die vorhergesagten Werte entsprechen den zugehörigen *y*-Punktkoordinaten der Regressionskurve. Zeichnen Sie die Originaldaten inklusive Regressionskurve in einem Diagramm.