Data Science

Wintersemester 2020/2021

Übungsblatt 9

Aufgabe 1 (E-Commerce Kunden)

Über Tracking kann man eine Menge Informationen über User bekommen, die noch nicht auf einer Web-Seite gekauft haben. Anhand der Informationen der bereits bekannten Kunden, also User, für die es eine Kauf-Historie gibt, kann man so probieren, das Potential von unbekannten Benutzern vorherzusagen.
Im Verzeichnis `Vorlesung/data` befindet sich die Datei `customer-profiles.csv`, die die aggregierten Informationen über bekannte Kunden enthält, sowie einige Eigenschaften, die aus dem Tracking dieser Kunden gewonnen wurde.

1. Laden Sie den Datensatz in einen DataFrame. Welche Attribute enthält der Datensatz?

2. Welche Attribute könnten ihrer Meinung nach aus den Tracking Daten eines Users abgeleitet werden? Welche Attribute stehen für unbekannte User wohl nicht zur Verfügung?

3. Die Spalte `sales` gibt den 2-Jahresumsatz eines Kunden an. Die Spalte `category` ist eine Aufteilung der Kunden in die Klassen `LOW`, `MID` und `HIGH` anhand der Spalte `sales`.

   Erzeugen Sie einen DataFrame `X` mit den `interest`- und `search` Spalten, sowie der Spalte `age`. Die Spalte `category` soll als Zielvariable `y` benutzt werden.

4. Teilen Sie `X` und `y` mit der Funktion `train_test_split` in Trainings- und Testdaten auf und trainieren Sie einen Entscheidungsbaum auf den Daten um das Attribute `category` vorherzusagen.

   Zur Erinnerung: Einen Entscheidungsbaum erzeugen Sie mit:

   ```
   from sklearn.tree import DecisionTreeClassifier()

   m = DecisionTreeClassifier()
   m.fit(X, y)  # Model trainieren
   ```

   Bestimmen Sie den Trainings- und Testfehler. Was fällt Ihnen dabei auf?

5. Trainieren Sie ihr Entscheidungsbaum-Modell mit unterschiedlichen Werten für den Parameter `max_depth`. Nehmen Sie dafür Werte aus dem Bereich $[1, \ldots, 20]$ und berechnen Sie jeweils den Trainings- und den Test-Fehler.

   Welchen Wert für `max_depth` würden Sie nehmen?

Aufgabe 2 (Klassifikation)

Im Verzeichnis `Vorlesung/data` finden Sie die Datei `fruits_simple.csv`. Diese Datei enthält einen kleinen Datensatz mit dem Gewicht, den Maßen und einem Farb-Wert für eine Reihe von Obstsorten.

1. Laden Sie den Datensatz in einen DataFrame.
2. Definieren Sie die Variable `X` als DataFrame mit den Spalten für Gewicht, Maße und Farbe, sowie die Variable `y` für die Spalte `fruit_name`.
3. Benutzen Sie die Funktion `train_test_split` aus SkLearn um einen Trainings- und Test-Datensatz zu erstellen.
4. Trainieren Sie ein lineares SVM Modell auf den Trainingsdaten und bestimmen Sie mit der Funktion `accuracy_score` aus dem Paket `sklearn.metrics` den Test-fehler.

   Hinweis: Ein lineares SVM Modell erzeugen Sie durch den Aufruf:

   ```
   from sklearn import svm

   m = svm.SVC(kernel="linear")
   ```

5. Denken Sie an den Parameter `C` für die Gewichtung des Trainingsfehlers. Für welchen Parameter `C` liefert ihr Modell den geringsten Test-Fehler?
6. Wie verbessert sich ihre Vorhersage, wenn Sie die Daten zuvor normalisieren?
7. Denken Sie sich ein Beispiel für ein Obst aus (Gewicht, Maße und Farbwert) und erzeugen Sie einen DataFrame für ihr ausgedachtes Beispiel. Welche Vorhersage macht das Modell auf ihrem Beispiel?