Hochschule Bochum
Bochum University
of Applied Sciences

Fachbereich Wirtschaft
Prof. Dr. Christian Bockermann

Data Science

Wintersemester 2020/2021

Übungsblatt 8

Aufgabe 1 (Hyperparameter-Suche)

Im Verzeichnis `Kurse/DataScience1/data/` finden Sie den Iris Datensatz `iris.csv`. Untersuchen Sie, welche Hyperparameter den kleinsten Trainingsfehler ergeben.

1. Laden Sie den Datensatz in einen DataFrame und erstellen Sie einen DataFrame `X` mit allen Spalten ausser `species`, sowie eine Series `y`, die die Spalte `species` enthält.

2. Benutzen Sie die Funktion `train_test_split` aus dem Modul `sklearn.model_selection` um `X_train`, `X_test`, `y_train` und `y_test` zu erzeugen.

3. Trainieren Sie ein lineares SVM Modell mit unterschiedlichen Werten für den Parameter `C` auf `X_train` und `y_train`. Wählen Sie dabei für `C` Werte von 100 bis 2500 in 100er Schritten.

   Für welches `C` liefert das Modell die besten Werte auf den Testdaten? Bestimmen Sie dazu für jeden Wert den `accuracy_score`.

   Das Ergebnis kann z.B. eine Liste der folgenden Art sein:

   ```
   [(c1, train-err1, test-err1), (c2, train-err2, test-err2), ...]
   ```

Aufgabe 2 (Generalisierungsfehler)

Wir hatten im Tutorial mal die Aufgabe zur Logistik. Die Daten hatte das Format wie in folgender Tabelle angegeben:

Länge (cm)	**Breite** (cm)	**Höhe** (cm)	**Entfernung** (km)	**Dienstleister**
100	100	40	20	WPS
110	100	100	60	UPS
100	210	100	70	WPS
50	100	70	89	DHL

Es soll dabei die Spalte `Dienstleister` vorhergesagt werden.
In der Datei `Kurse/DataScience1/data/versand-data.csv` finden Sie einen großen Datensatz mit Versandangaben und dem jeweils zugehörigen Dienstleister.

1. Laden Sie den Datensatz in einen DataFrame und normalisieren Sie die numerischen Spalten mit der z-Normalisierung.

   Hinweis: Für die Normalisierung können Sie natürlich gerne Ihre eigenen Funktionen vom Übungsblatt 6 benutzen. Alternativ steht eine Normalisierungsfunktion im Modul `datascience` zur Verfügung:

   ```
   import datascience
   df_norm = datascience.z_norm_df(df)
   ```

2. Trainieren Sie mit *SciKit-Learn* einen SVM-Klassifizierer (SVC) mit `rbf` Kern-Funktion und geben Sie den Trainings- und den Test-Fehler an.

   Nutzen Sie dazu wieder die Funktion `train_test_split` um 20% der Daten für das Testen zu verwenden.

3. Probieren Sie verschiedene Kombinationen der Kernfunktionen `linear` und `rbf` mit Werten für den Parameter `C` von 1 bis 1001 in 100er Schritten.

4. Erstellen Sie einen DataFrame mit den Spalten `KernFunktion`, `C`, dem Trainings- und dem Test-Fehler.

5. Berechnen Sie die Spalte `Volumen` als zusätzliches Merkmal und wiederholen Sie das Training, sowie die Evaluation. Wie hat sich ihre Vorhersage verbessert?