Lehrstuhl für Software Engineering

Übungsblatt: Lineare Verfahren

In dieser Übung werden Sie selbst das Verfahren der *linearen Regression* implementieren. Für diesen Zweck wird Ihnen eine Python-Datei mit vorgefertigten Hilfsmethoden und Methodenköpfen, die Sie implementieren müssen, zur Verfügung gestellt.

In dieser Übung werden Sie mit einem Datenset über Hauspreise in Boston arbeiten. Dieses liegt im Ordner *data* als csv-Datei und wird mittels *Pandas*[1] importiert. Das Datenset enthält 13 Features sowie die zugehörigen Hauspreise (der Name dieser Spalte ist *target*).

Mittels der Methode *get_linear_regression_training_set_from_df* können Sie sich ein Trainingsset für beliebige Features ausgeben lassen.

Aufgabe 1

In dieser Aufgabe sollen Sie sich zunächst die Datenpunkte für verschiedene Features in Relation zu den Hauspreisen ansehen. Zu diesem Zweck können sie die Methode *plot_features_from_df* nutzen.

Schauen Sie sich die Features 2, 3, 4, 5 und 9 an.
Welches dieser Features lässt sich am besten mit einem linearen Modell darstellen?
Geben Sie Ihre Antwort als String in der Methode *question_1* zurück. Ihre Antwort muss der Name eines Features in derselben Schreibweise wie im Datenset sein.

Welches dieser Features lässt sich am schlechtesten mit einem linearen Modell darstellen?
Geben Sie Ihre Antwort als String in der Methode *question_2* zurück. Ihre Antwort muss der Name eines Features in derselben Schreibweise wie im Datenset sein.

Aufgabe 2

Implementieren Sie zu erst die Methoden *mse* und *gradient_descent*.

In der Methode *mse* sollen Sie die Kostenfunktion M*ean-Squared-Error* implementieren. Mittels dieser Kostenfunktion kann das Model evaluiert werden.

In der Methode *gradient_descent* sollen die optimalen Model-Parameter für das übergebene Datenset ermittelt werden. Nutzen Sie dazu die aus der Vorlesung bekannte Formel und Vorgehensweise, um die Modell-Parameter direkt iterativ zu berechnen, dabei wird die von Ihnen implementierte Methode *mse* als Kostenfunktion verwendet. Außerdem steht Ihnen als Unterstützung schon einige Code-Zeilen zur Verfügung, sodass Sie nur an den mit TODO markierten Stellen etwas einfügen müssen. Gerne dürfen Sie aber auch die Methode auf Ihre eigene Weise implementieren, solange die Methodensignatur und Rückgabewerte gleich erhalten bleiben.
Hinweis: Zusätzlich zur Vorlesung verwenden wir hier einen *stopping_threshold:* sobald dieser unterschritten wird, kann es z.B. aus numerischer Sicht sinnvoll sein, die Iteration zu beenden.

---

[1] https://pandas.pydata.org

Aufgabe 3

Alternativ zur Annäherung durch das Gradientenverfahren können (z.B. bei wenigen Samples) die optimalen Werte auch direkt berechnet werden.
Die Formel dafür lautet: $\theta = (X^T X)^{-1} X^T y$.
Implementieren Sie diese Formel in der Methode *closed_form_solution*. Mit Hilfe dieser Formel können die optimalen Parameter errechnet werden.
Die Herleitung der Formel finden Sie im Anhang der Vorlesungsfolien.

Interlude
Die Methoden *closed_form_solution* und *mse* werden genutzt, um die in Aufgabe 1 betrachteten Features einzeln zu evaluieren. Entspricht das Verhältnis der *MSEs* zueinander dem was Sie in Aufgabe 1 erwartet haben?

Aufgabe 4

Berechnen Sie die optimalen Model Parameter für die Features 2, 3, 4, 5 und 9 zusammengenommen.
Welches dieser Features hat den größten Einfluss auf die Vorhersage?
Geben Sie Ihre Antwort als String in der Methode *question_3* zurück. Ihre Antwort muss der Name eines Features in derselben Schreibweise wie im Datenset sein.

Welches dieser Features hat den geringsten Einfluss auf die Vorhersage?
Geben Sie Ihre Antwort als String in der Methode *question_4* zurück. Ihre Antwort muss der Name eines Features in derselben Schreibweise wie im Datenset sein.

Optional
Nutzen Sie die Methoden, um verschiedene Features und Kombinationen von Features zu evaluieren.