Technische Universität Chemnitz
Fakultät für Mathematik

Prof. Dr. O. Ernst
J. Blechschmidt

14. Oktober 2018
Abgabe Hausaufgabe bis 28. Oktober 2018

Einführung in Data Science

Übung 2: Datenimport und Lineare Regression

Aufgabe 1: Einführung – Data Import

Lade dir von der Homepage zur Lehrveranstaltung die Dateien `Advertising.csv` sowie `importAdvertising.py` herunter. Betrachte die `csv`-Datei in einem Tabellenkalkulationsprogramm, z. B. `LibreOffice`. Die Datei enthält Informationen darüber, inwieweit die Ausgaben für Fernseh-, Radio- und Zeitungswerbung mit den Verkaufszahlen zusammenhängen.

Führe die Aufgaben in der Datei `importAdvertising.py` aus, d.h. importiere die `csv`-Datei mit der Funktion `np.genfromtxt` und stelle die Daten geeignet grafisch dar.

Aufgabe 2: Erstellung einer Funktion in Python

Der Mean Squared Error, kurz MSE, ist eine der wichtigsten Kenngrößen für die Güte eines Fits.

Ziel ist die Erstellung der Funktion `computeMSE` mit folgendem Input:

- die zu den Messungen $x_i \in X$, $i = 1, \ldots, N$ gehörigen Beobachtungen $y_i \in Y$, $i = 1, \ldots, N$
- die Vorhersage von $f(x_i)$: $\hat{f}(x_i), i = 1, \ldots, N$

und zugehörigem Output:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{f}(x_i))^2$$

Lade dir die Datei `mse.py` von der Homepage der Lehrveranstaltung und führe die darin enthaltenen Aufgaben aus.

Aufgabe 3: "Nichtlineare" lineare Regression

Lade dir die Datei `higherOrderLinReg.py` von der Homepage zur Lehrveranstaltung. Das Ziel der Aufgabe ist es, gegebene Datenpunkte (x_i, y_i) für $i = 1, \ldots, n$ durch Polynome vom Grad p zu approximieren, indem wir ein lineares Regressionsproblem lösen:

$$y_i \approx \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_p x_i^p$$

Anhand der Aufteilung der Daten in einen Trainingsdatensatz und einen Testdatensatz wollen wir grafisch das Problem des Overfittings illustrieren.

Hierfür kannst du diesselben Funktionen wie in Aufgabe 2 verwenden. Insbesondere solltest du die Funktion `computeMSE` in die aktuelle Datei kopieren, oder mittels

```
from mse import computeMSE
```

in das aktuelle Skript einbinden.

Führe die Aufgaben aus, die in der Datei `higherOrderLinReg.py` notiert sind. Spiele mit den Parametern `eps`, `n` und `ntrain`, was beobachtest du?

Hinweis: Beim Einbinden mittels `from mse import computeMSE` werden beim Ausführen des aktuellen Skripts `higherOrderLinReg.py` auch alle anderen Kommandos in dem Skript `mse.py` ausgeführt. Um das zu verhindern, solltest du die Struktur des Skripts `mse.py` ändern:

```
import numpy as np
import matplotlib.pyplot as plt

def computeMSE(...):
    # Put here the definition of the function
    return ...

if __name__ == "__main__":
    # Put here all the rest
```

Hausaufgabe 1: Einführung – Pandas

In Aufgabe 1 hast du eine Möglichkeit kennengelernt, `csv`-Dateien mit der Funktion `np.genfromtxt` zu importieren. Das Problem hierbei ist, dass `numpy`-Arrays grundsätzlich auf einen Datentyp festgelegt sind. Haben wir unterschiedliche Arten von Daten, z. B. Booleans, floats, integers, strings in einer csv Tabelle, kommen wir damit nicht weiter.

Eine Möglichkeit besteht in der Nutzung des Pakets `csv`. Da hierbei jede Zeile einzeln eingelesen wird, ermöglicht es dies, jeden Eintrag einzeln zu behandeln, und auch Spezialfälle direkt abzufangen.

Eine andere Möglichkeit, die in der Komplexität zwischen diesen beiden Methoden steht, bietet das Paket `pandas`, welches mit

```
import pandas as pd
```

geladen werden kann, mit der Funktion `pandas.read_csv`.

Erstelle das Skript `importAuto.py`, in welchem folgende Aufgaben abgearbeitet werden.

(a) Mache dich mit `pandas` durch das Tutorial https://pandas.pydata.org/pandas-docs/stable/10min.html#min vertraut.

(b) Lade dir von der Homepage zur Lehrveranstaltung die Datei `Auto.csv` herunter. Importiere die `csv`-Datei mit der Funktion `pandas.read_csv` unter dem Namen `Auto`. Achte darauf, dass die Datei fehlende Daten enthalten kann, die beim Import korrekt beachtet werden sollten. Nutze den optionalen Parameter `na_values` der Funktion `pandas.read_csv`. In dieser Aufgabe wollen wir Datensätze, die einen fehlenden Wert enthalten, löschen. Wir können dies mit der Methode `dropna(axis=0, inplace=True)` erreichen. Danach sollte der `DataFrame` 392 Datensätze mit 9 Features enthalten.

(c) Erstelle mit der Methode `describe` eine kurze Übersicht der wichtigsten statistischen Kenngrößen. .

(d) Erstelle mit der Methode `hist` ein grafische Übersicht über die Verteilung der Inputvariablen. Versuche, mit Hilfe der Funktion

```
pd.scatter_matrix(Auto, marker='o', alpha=.7)
```

Rückschlüsse auf mögliche Zusammenhänge zwischen den Daten zu ziehen.

(e) Erstelle zwei Plots, die jeweils die Variable "horsepower" in Beziehung setzen zu "mpg" bzw. "weight". Nutze dafür die Möglichkeiten, die durch `pandas` bereitgestellt werden.

(f) Untersuche, inwieweit ein linearer bzw. quadratischer Zusammenhang zwischen "horsepower" und "mpg" sowie zwischen "horsepower" und "weight" zu beobachten ist. Nutze dafür auch den MSE (mean squared error). Was stellst du fest?