OST
Ostschweizer
Fachhochschule

Python for Machine Learning Übungen

Part 3

OST
Ostschweizer Fachhochschule

Oberstrasse 10
8640 Rapperswil
Switzerland

T +41 79 350 24 36
michael_inden@hotmail.com
ost.ch

Michael Inden

7 Aufgaben zum Datei-Handling

Aufgabe 1: Verzeichnis und Dateien anlegen und Inhalt auflisten

Legen Sie mindestens zwei Verzeichnisse, etwa `new_and_empty` und `another-dir`, an. Erstellen Sie zwei bis drei Dateien auf gleicher Ebene, etwa `image-info.txt` und `response-data.json` und `mypic.png`.

Aufgabe 2: Texte in Datei schreiben und wieder lesen

Schreiben Sie ein paar Zeilen in die Datei, etwa die gerade angelegte Datei `image-info.txt`, und lesen Sie diese danach wieder ein. Fügen Sie ein paar Zeilen an.

Aufgabe 3: JSON in Datei schreiben und wieder lesen

Befüllen Sie ein Dictionary mit einigen Werten und schreiben Sie die dort gespeicherten Informationen in eine Datei, etwa `response-data.json`. Lesen Sie diese Daten danach wieder ein.

Aufgabe 4: Verzeichnisinhalt auflisten

Schreiben Sie ein Python-Programm, um den Inhalt eines Verzeichnisses auszugeben, ohne dies für alle möglicherweise enthalten Unterverzeichnisse zu wiederholen. Dabei sollen Verzeichnisse und Dateien unterschiedlich markiert werden. Geben Sie auch die Dateigröße an, für Verzeichnisse einfach -/-. Das kann in etwa so aussehen:

```
TEST -/- [DIR]
ex01_write.py 277
ex04_fileinfo.py 322
ex05_print_dir.py 326
ex03_existence.py 99
TESTDIR -/- [DIR]
ex02_filesize.py 281
MIKE-DIR -/- [DIR]
data.txt 83
personenListe.txt 67
```

Aufgabe 5: CSV-Highscore-Liste einlesen

In dieser Aufgabe geht es um die Verarbeitung von kommaseparierten Daten, auch CSV (Comma Separated Values) genannt. Statt trockener Anwendungsdaten nutzen wir als Eingabe eine Liste von Spielständen. Stellen wir uns eine x-beliebige Spieleapplikation vor, die es einem Spieler erlaubt, entsprechende Punktzahl vorausgesetzt, sich in einer Highscore-Liste zu verewigen.

Die Highscores sollen mithilfe eines `namedtuple` modelliert werden. Zudem sind die Spielstände wären kommasepariert etwa wie folgt in Form in einer Datei `Highscores.csv` gespeichert. In diesem Beispiel sind bewusst auch fehlerhafte Einträge dargestellt, die dazu dienen, die die Implementierung einer robusten Fehlerbehandlung erforderlich machen. Natürlich muss das Einlesen auch Leerzeilen und Zeilen mit Kommentaren (startend mit #) passend verarbeiten, also ignorieren ☺

```
# Name, Punkte, Level
Matze, 1000, 7
Peter, 985, 6
ÄÖÜßöäü, 777, 5

# Fehlender Level
Peter, 985,

# Falsches Format des Levels
Peter, 985, A6
```

Als Ausgangspunkt ist noch die `main()`-Funktion gegeben:

```
def main():
    highscores_from_csv = read_highscores_from_csv("Highscores.csv")
    print("Highscores:")
    for highscore in highscores_from_csv:
        print(highscore)


if __name__ == "__main__":
    main()
```

8 Aufgaben Datumsverarbeitung

Aufgabe 1: Wochentage

Welcher Wochentag war der Heiligabend 2019 (24. Dezember 2019)? Welche Wochentage waren der erste und der letzte Tag im Dezember 2019?

Eingabe	Resultat
24. Dezember 2019	Dienstag
01. Dezember 2019	Sonntag
31. Dezember 2019	Dienstag

Aufgabe 2: Freitag, der 13.

Berechnen Sie alle Vorkommen von Freitag, dem 13. für einen Bereich, definiert durch zwei Datumsangaben. Schreiben Sie eine Funktion `all_friday13th(startIncl, endExcl)`, wobei das Startdatum inklusive und das Enddatum exklusive anzugeben ist.

Zeitspanne	Resultat
2013 – 2015	[2013-09-13, 2013-12-13, 2014-06-13, 2015-02-13, 2015-03-13, 2015-11-13]

Aufgabe 3: Mehrmals Freitag, der 13.

In welchen Jahren gab es mehrfach Freitag, den 13.? Um diese Frage etwa für den Zeit- raum von 2013 bis einschließlich 2015 zu beantworten, berechnen Sie ein Dictionary, in dem zu jedem Jahr die entsprechenden Freitage assoziiert sind. Schreiben Sie dazu die Funktion `friday13th_grouped(startIncl, endExcl)`.

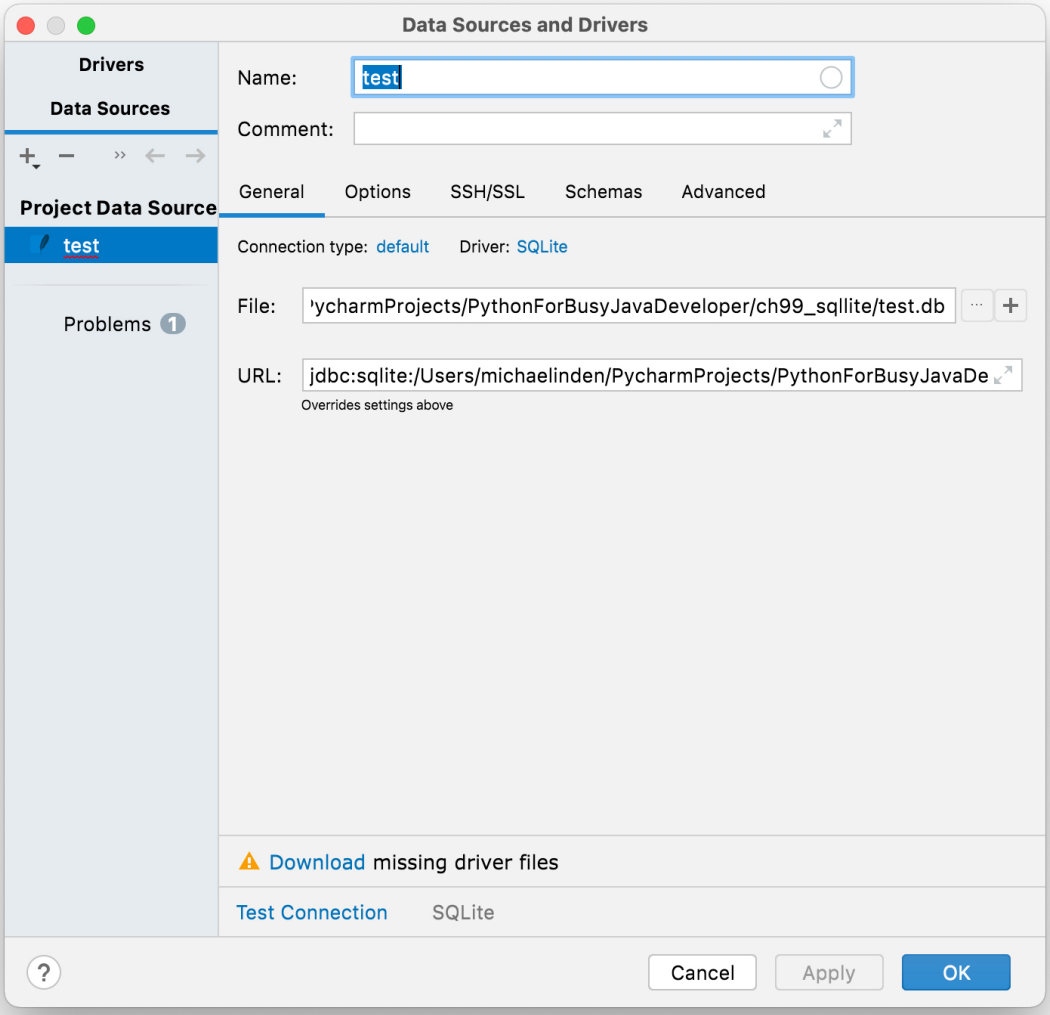
Jahr	Resultat
2013	[2013-09-13, 2013-12-13]
2014	[2014-06-13]
2015	[2015-02-13, 2015-03-13, 2015-11-13]

Aufgabe 4: Schaltjahre

In dieser Aufgabe soll die Anzahl der Schaltjahre in einem Bereich, gegeben durch zwei Jahreszahlen. Dazu implementieren wir eine Funktion `count_leap_years(start, end)`, wobei das Startjahr inklusive und das Endjahr exklusive anzugeben ist.

Zeitraum	Resultat
2010 – 2019	2
2000 – 2019	5

Database



Praxisübungen: Tic Tac Toe und Worträtsel

- Analysieren / Erstellen / Komplettieren Sie ein einfaches Tic Tac Toe-Spiel
- Analysieren / Erstellen / Komplettieren Sie ein einfaches Worträtsel
    - File IO
    - Klassen
    - HTML
    - Webbrowser