INTERNATIONALE HOCHSCHULE

HAUSARBEIT

Aufgabenstellung zum Kurs:

DLMDWPMP01 – Programmieren mit Python

INHALTSVERZEICHNIS

1. AUFGABENSTELLUNG

Für die Hausarbeit steht folgende Aufgabenstellung zur Verfügung.

Als Ausgangsbasis für die Hausarbeit dient zunächst das Studienskript, dessen Inhalte als Basiswissen die Voraussetzung für die vertiefende Betrachtung der nachfolgenden Fragestellung darstellen. Es wird erwartet, dass in der Hausarbeit weitere Literaturquellen zu dieser Fragestellung recherchiert und verarbeitet werden.

1.1. Die Aufgabe

Du erhältst:

A) 4 Training-Datensätze

B) einen Test-Datensatz

C) einen Datensatz, der 50 ideale Funktionen beschreibt

Alle Daten bestehen aus x-y-Paaren. Die Struktur in den CSV-Files ist wie folgt:

x	y
X1	Y1
…	…
Xn	Yn

Deine Aufgabe ist, ein Python Programm zu schreiben, welches mittels der vier Trainingsdatensätze (A) die vier besten Passungen / Fits aus dem Datensatz von 50 idealen Funktionen (C) findet. Die folgenden Kriterien sollen beachtet werden:

1. Das Kriterium zur Selektion idealer Funktionen für den Training-Datensatz ist die Minimierung der Summe aller quadratischen y-Abweichungen (Least-Square).
2. Dein Programm muss den Test-Datensatz B zur Validierung der Selektion benutzen. Hierbei soll für jedes x-y-Paar im Test-Datensatz überprüft werden, ob die Werte zu den vier idealen Funktionen passen.
    a. Benutze ein Kriterium, welches sicherstellt, dass die maximale Abweichung zwischen der vorher ermittelten idealen Funktion und den Testwerten nicht die maximale Abweichung zwischen den Trainingsdaten (A) und den vier idealen Funktionen aus (C) um mehr als den Faktor *Wurzel aus zwei* (sqrt(2)) übersteigt.
    b. Sollten die Testdaten an die von Dir gefundenen vier Funktionen anpassbar sein, speichere für jeden Testdatensatz die entsprechenden Abweichungen ab.
3. Alle Daten sollten logisch visualisiert werden.
4. Schreibe Unit-Tests, wo immer möglich.

Um Deine im Kurs erlernten Fähigkeiten unter Beweis zu stellen, musst Du die im folgenden Kapitel (Details) dargestellten Kriterien erfüllen.

1.2. Details

Datenbank und Tabellen

- Du erhältst vier Trainingsdatensätze in Form von CSV-Dateien. Dein Python-Programm muss in der Lage sein, eine SQLite-Datenbank (Datei) idealerweise über sqlalchemy unabhängig zu kompilieren und die Trainingsdaten in eine einzelne, fünfspaltige Tabelle zu laden. Die erste Spalte zeigt die x-Werte aller

Funktionen. Tabelle 1 am Ende dieses Unterabschnitts zeigt Dir, welche Struktur Deine Tabelle voraussichtlich haben wird.

- Die fünfzig idealen Funktionen, die auch über eine CSV-Datei bereitgestellt werden, müssen in eine andere Tabelle geladen werden. Ebenso zeigt die erste Spalte die x-Werte, was bedeutet, dass insgesamt 51 Spalten vorhanden sind. Tabelle 2 am Ende dieses Unterabschnitts beschreibt schematisch, welche Struktur erwartet wird.
- Nachdem die Trainingsdaten und die idealen Funktionen in die Datenbank geladen wurden, müssen die Testdaten (B) Zeile für Zeile aus einer anderen CSV-Datei geladen und - wenn sie das Kriterium im Unterabschnitt 2 erfüllt - mit einer der vier abgeglichen Funktionen abgespeichert werden.
- Anschließend müssen die Ergebnisse in einer anderen vierspaltigen Tabelle in der SQLite-Datenbank gespeichert werden. Gemäß Tabelle 3 am Ende dieses Unterabschnitts enthält diese Tabelle vier Spalten mit x- und y-Werten sowie die entsprechend gewählte ideale Funktion und die damit verbundene Abweichung.
- Schließlich werden die Trainingsdaten, die Testdaten, die gewählten Idealfunktionen sowie die entsprechenden / zugewiesenen Datensätze unter einer entsprechend gewählten Darstellung der Abweichung visualisiert.

Struktur des Python Programms

- Das Programm soll soweit wie möglich Objekt-orientiert sein.
- Es soll mindestens eine Vererbungshierarchie (inheritance) haben.
- Benutze sowohl Standard als auch user-definiertes Exception Handling.
- Für die Programmlogik solltest Du Pandas benutzen, aber auch Visualisierung mittels Bokeh, matlibplot etc.
- Schreibe Unit-Tests, wo immer es sich anbietet.
- Dokumentiere Dein Programm vollständig und mache von docstrings Gebrauch.

Verwendung von Git

- Bitte verwende Git zur Versionskontrolle Deines Codes.

Tabelle 1: Training Daten Datenbank Tabelle

X	Y1 (Training Funktion)	Y2 (Training Funktion)	Y3 (Training Funktion)	Y4 (Training Funktion)
x1	y11	y21	y31	y41
…	…	…	…	…
xn	y1n	y2n	y3n	y4n

Tabelle 2: Tabelle der idealen Funktionen

X	Y1 (Ideale Funktion)	Y2 (Ideale Funktion)	…	Y50 (Ideale Funktion)
x1	y11	y21	…	y41
…	…	…	…	…
xn	y1n	y2n	…	y4n

Tabelle 3: Test-Daten Tabelle

X (Test Funktion)	Y1 (Test Funktion)	Delta Y (Abweichung)	Nummer der Idealen Funktion (z.B. Funk37)
x1	y11	y21	y31
…	…	…	…
xn	y1n	y2n	y3n

1.3. Anmerkungen

Der Datensatz für diese Aufgabe wird auf Anfrage für jeden einzelnen Studenten zur Verfügung gestellt. Daher sollte ein Ticket für den Tutor geöffnet werden, woraufhin der Zugriff auf die Daten gewährt wird. Eine Kopie wird an die verantwortlichen Personen gesendet - so wird eine spätere Manipulation durch die Studierenden verhindert.

Es wird erwartet, dass Dein gesamter Quellcode im Anhang Deiner schriftlichen Aufgabe enthalten ist, damit Dein gesamtes Programm einschließlich der Ausgaben getestet werden kann. Deine Eingabedaten sind nicht erforderlich.

Ziel ist es, Deine Arbeit, Deine Entscheidungen und Deine Einschätzung des Aufgabenergebnisses durch Deine Abgabe vollständig zu rekonstruieren.

2. ZUSATZINFORMATIONEN ZUR BEWERTUNG DER HAUSARBEIT

Bei der Konzeption und Erstellung der Hausarbeit sollten die im Prüfungsleitfaden aufgeführten Bewertungskriterien und Erläuterungen berücksichtigt werden.

Bezüglich **Einführung und thematischer Abgrenzung** sollte darauf geachtet werden, dass diese im gewählten Lösungsansatz der Aufgabe demonstriert werden.

Die Bewertung der **Struktur** bezieht sich auf das Design des Programms, Klassenstruktur, Wahl von Verallgemeinerungen im Programm und die Komposition des Programms.

In der **Argumentation** werden die finale Funktionalität und die Richtigkeit der Ausführung des Programms bewertet.

Der **Abschluss** soll einen wissenschaftlich adäquaten Text und eine Diskussion der Vor- & Nachteile des gewählten Lösungsansatzes, im Speziellen eine Diskussion der Abgrenzung zu anderen möglichen Lösungen, der Programmstruktur und der Module und Frameworks, die für die Lösung verwendet wurden, darstellen.

3. BETREUUNGSPROZESS

Für die Betreuung der Hausarbeit stehen grundsätzlich mehrere Kanäle offen. Die jeweilige Inanspruchnahme liegt dabei im eigenen Verantwortungsbereich. Der/Die Tutor:in steht per E-Mail für fachliche Rücksprachen zur Themenwahl einerseits sowie für formale und allgemeine Fragen zum wissenschaftlichen Arbeiten andererseits zur Verfügung. Eine Abnahme von Gliederungen, Textteilen oder –entwürfen durch den/die Tutor:in ist hierbei jedoch nicht vorgesehen, da die eigenständige Erstellung Teil der zu erbringenden Prüfungsleistung ist und in die Gesamtbewertung einfließt. Es werden jedoch Hinweise zu Gliederungsentwürfen gegeben, um den Einstieg in die Strukturierung einer wissenschaftlichen Arbeit zu erleichtern.