

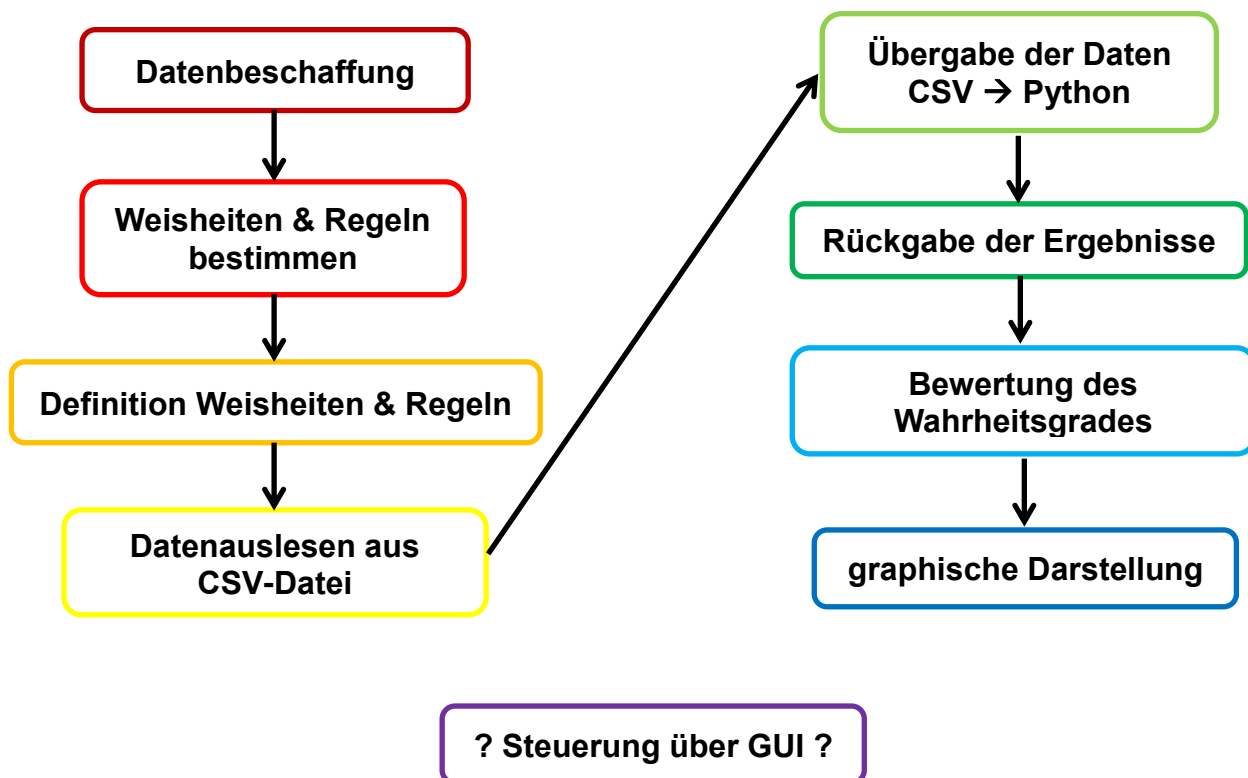
Projektplan zum Projektseminar „Angewandte Informationswissenschaft“

Idee: Analysetool zur Prüfung von Wetterdaten auf Korrektheit von Wetterweisheiten und Bauernregeln für den Raum Düsseldorf

Frage: „Wie können Wetterdaten in Bezug auf Wetterweisheiten und Bauernregeln für den Raum Düsseldorf, mit Python automatisch analysiert und deren Genauigkeitsgrad graphisch dargestellt werden?“

Programmiersprache, Tools: Python, D3, Matplotlib

Ablaufplanung:



Umsetzung: **Umsetzungsidee** / **Probleme**

Vorbereitung:

1. Datenbeschaffung:

- Wetterdaten in einem Austauschformat (z.B. „.csv“) beschaffen
- ✓ **DWD Datenbank (WESTE)**
- ❖ **Datenbeschaffung kompliziert evtl. andere Datenquelle**
- ❖ **pro CSV-Datei max. 10.000 Einträge möglich**

2. Weisheiten und Regeln bestimmen:

- Weisheiten bzw. Bauernregeln beschaffen und benötigte Werte prüfen
- ✓ **Nachschlagen z.B. Wetter.de**
- ❖ **Umfang prüfen, Mittelmaß zwischen zu wenig und zu viel**
- ❖ **Welche Weisheiten/Regeln lassen sich mit den gegebenen Werten prüfen?**

Kernstruktur:

3. Definition der Weisheiten/Regeln:

- Definieren der benötigten Werte jeder Weisheit/Regel
- Bestimmen der Referenzwerte, wann trifft Regel zu, wann nicht
- ✓ **Übersetzung der Weisheiten/Regeln zu Python-Code**

4. Datenauslesen aus CSV-Datei:

- Funktion zur Identifizierung der gesuchten Werte definieren
- ✓ **Suchfunktion in Python für CSV-Datei (Zeile/Spalte) oder (Suchwort/Zeile/Spalte)**
- ❖ **Ordnung der Daten nicht kontinuierlich, evtl. sortieren**
- ❖ **Sonderzeichen**

5. Übergabe der Daten aus CSV-Datei zu den Python-Funktionen

- Funktion zur Übergabe der benötigten Werte
- ✓ **Übergabe direkt aus der CSV-Datei oder aus Zwischenformat**
- ❖ **Format und Vollständigkeit**

6. Rückgabe der Ergebnisse nach der Analyse der Daten

- Rückgabe der Ergebnisse als Vorbereitung für graphische Darstellung
- ✓ **Ergebnis mit Referenzwert und Namen der Weisheit/Regel wird zurückgegeben**
- ❖ **Format der Ergebnisse muss graphisch darstellbar sein**

7. Bewertung des Wahrheitsgrades der Wetterweisheiten/Bauernregeln

- Funktion zur Bewertung, ob die Weisheiten/Regeln wahr oder falsch sind bzw. Grad der Abweichung vom Referenzwert
- ✓ Ergebnisse mit Referenzwert vergleichen und bewerten
- ❖ Format der Ergebnisse muss graphisch darstellbar sein

8. graphische Darstellung der Ergebnisse

- Ergebnisse mit D3 oder notfalls mit Matplotlib darstellen
- ✓ Ergebnisse und Abweichungsgrad übersichtlich darstellen
- ❖ geeigneter Maßstab und Darstellung der Abweichung, bei großen Unterschieden und kleinen Unterschieden

Erweiterungen:

9. Erstellung und Steuerung GUI

- Bedarf einer GUI prüfen
- Auswahl verschiedener Jahre und/oder Orte
- ✓ GUI erstellen und Steuerung der Funktionen festlegen
- ❖ Aufwand/Nutzen der GUI abwägen

Beispiel:

Siebenschläfer = „Wenn's am Siebenschläfer gießt, sieben Wochen Regen fließt.“

Siebenschläfer ist der 27. Juni, wenn es an diesem Tag regnet, soll es gemäß dieser Weisheit sieben Wochen (bis zum 16. August) regnen. Um diese Weisheit zu prüfen, muss der Wert der Niederschlagshöhe in der CSV > 0 sein. Anschließend müssen die Niederschlagshöhen vom 28. Juni bis zum 16. August erfasst werden. Diese werden in einer Liste gespeichert und anschließend werden die Nullwerte gelöscht, sowie die Anzahl der Einträge gezählt. Um den Grad der Abweichung bestimmen zu können, wird der errechnete Wert vom Referenzwert (7 Tage x 7 Wochen = 49) abgezogen. Somit wäre bei nur 19 Regentagen eine Abweichung von 30 Tagen → 61.2%.

Vorbereitung:

1. **Datenbeschaffung:** Der Deutsche Wetterdienst (DWD) betreibt verschiedene Datenbanken, dazu gehört WESTE XL. Mit dieser Datenbank ist es möglich für einen bestimmten Zeitraum unterschiedliche Werte (Lufttemperatur Tagesmittel, Lufttemperatur Tagesminimum, Lufttemperatur Tagesmaximum, Niederschlagshöhe, Schneehöhe, Neuschneehöhe, Sonnenscheindauer,

Windspitze und relative Luftfeuchte), entweder stündlich, oder täglich nachzuschlagen. Da jede „Kategorie“ und jede Wetterstation einzeln ausgewählt werden muss, kann die Beschaffung viel Zeit in Anspruch nehmen. Die gewählten Daten können in einem Austauschformat (z.B. als CSV-Datei) gespeichert werden. Dabei ist zu beachten, dass pro Datei maximal 10.000 Einträge möglich sind, weshalb evtl. mehrere CSV-Dateien nötig sein werden. Anschließend sollten die Daten „Bereinigt“ werden, wodurch nicht verwendete Spalten gelöscht werden können.

Element	Messstation	Datum	Wert	Einheit	Geo-Breite	Geo-Länge
Lufttempera	DÄ%sseldorf	01.09.2001	11,5	Grad C	51,289	6,781
Lufttempera	DÄ%sseldorf	02.09.2001	14,6	Grad C	51,289	6,781
Niederschlag	DÄ%sseldorf	01.09.2001	0	mm	51,289	6,781
Niederschlag	DÄ%sseldorf	02.09.2001	0	mm	51,289	6,781
SchneehÄ%h	DÄ%sseldorf	01.09.2001	0	cm	51,289	6,781
SchneehÄ%h	DÄ%sseldorf	02.09.2001	0	cm	51,289	6,781
Neuschneeh	DÄ%sseldorf	01.09.2001	0	cm	51,289	6,781
Neuschneeh	DÄ%sseldorf	02.09.2001	0	cm	51,289	6,781
Sonnenschei	DÄ%sseldorf	01.09.2001	5,5	hour	51,289	6,781
Sonnenschei	DÄ%sseldorf	02.09.2001	3,3	hour	51,289	6,781
Windspitze	DÄ%sseldorf	01.09.2001	3,9	m/s	51,289	6,781
Windspitze	DÄ%sseldorf	02.09.2001	5,7	m/s	51,289	6,781
Lufttempera	DÄ%sseldorf	01.09.2001	18,1	Grad C	51,289	6,781
Lufttempera	DÄ%sseldorf	02.09.2001	18,2	Grad C	51,289	6,781
Lufttempera	DÄ%sseldorf	01.09.2001	4,6	Grad C	51,289	6,781
Lufttempera	DÄ%sseldorf	02.09.2001	8,4	Grad C	51,289	6,781
Relative Luft	DÄ%sseldorf	01.09.2001	85	%	51,289	6,781
Relative Luft	DÄ%sseldorf	02.09.2001	86	%	51,289	6,781

Abb.1 Beispiel für CSV-Datei von WESTE XL

- Weisheiten und Regeln bestimmen:** Wetter.de bietet eine Aufzählung von Wetterweisheiten bzw. Bauernregeln an. Dabei gibt es tages- bzw. monatspezifische Besonderheiten. Da es eine sehr große Menge an Regeln gibt, sollte ein Mittelmaß ausgewählt werden. Dabei ist auf die benötigten meteorologischen Daten zu achten.



Bauernregeln März

Januar Februar März April Mai Juni Juli August September Oktober November Dezember

Regnet's stark zu Albinus, macht's dem Bauern viel Verdruss.
Erklärung: Der 1. März ist der Tag des heiligen Albinus von Angers (um 496-554). Er wurde zunächst Abt im Kloster Tincillacense in Westfrankreich und im Jahre 529 Bischof von Angers.

Wenn es an Sankt Albin regnet, gibt es weder Heu noch Stroh.
Erklärung: Der 1. März ist der Tag des heiligen Albinus von Angers (um 496-554). Er wurde zunächst Abt im Kloster Tincillacense in Westfrankreich und im Jahre 529 Bischof von Angers.

So wie der erste März, so der Frühling. So wie der 2. März, so der Sommer. So wie der 3. März, so der Herbst.

Märzenstaub und Märzenwind, guten Sommers Vorboten sind.

Ist Kunigunde tränenschwer, bleibt oft die Scheune leer.

Eine kühle und trockene Nacht
Der Wolken- und Regenfilm für Deutschland für die nächsten 48 Stunden.

Anzeige

Abb.2 Beispiel für Bauernregeln (März) von Wetter.de

Kernstruktur:

- 3. Definition der Weisheiten und Regeln:** Die ausgewählten Regeln müssen in Python-Code „übersetzt“ werden. Hierbei ist zu beachten, welche Werte die Regeln wirklich brauchen und was als Ergebnis erwartet wird. Dabei muss ein Referenzwert für jede Regel festgelegt werden, damit später der Grad der Abweichung bestimmt werden kann.
- 4. Datenauslesen aus CSV-Datei:** Allgemeine Suchfunktion für CSV-Datei definieren. Hierbei ist auf die Struktur der Datei zu achten. WESTE XL trägt die verschiedenen ausgewählten Werte zeilenweise ein. Aus diesem Grund muss die Suchfunktion erst die Zeile und dann die betreffende Spalte erkennen können. Es wäre auch möglich eine Suchfunktion über das NLTK zu definieren, hierbei müsste man die gesuchte Spalte mit einem Suchwort bzw. Regulären Ausdruck definieren. Hierbei muss besonders auf Sonderzeichen geachtet werden.
- 5. Übergabe der Daten aus CSV-Datei zu den Python-Funktionen:** Beim Aufruf der Funktionen müssen die benötigten Daten aus der CSV-Datei übergeben werden. Dies erfolgt entweder durch die in 4. definierte Suchfunktion, oder direkt aus der CSV-Datei. Hier sollte auf Vollständigkeit und Format der Daten geachtet werden.
- 6. Rückgabe der Ergebnisse nach Analyse der Daten:** Nachdem das Ergebnis der Analyse ermittelt wurde, wird dies zusammen mit dem Referenzwert und dem Namen der Weisheit/Regel zurückgegeben. Diese Werte werden später für die graphische Darstellung benötigt. Aus diesem Grund sollte schon hier auf das Format der Ergebnisse geachtet werden.
- 7. Bewertung des Wahrheitsgrades der Wetterweisheit/Bauernregel:** Um bestimmen zu können, ob die Weisheit/Regel zutrifft, müssen diese Werte verglichen werden. Falls es zu einer Abweichung kommt, wird der Grad der Abweichung bestimmt. Wie im 6. Schritt muss auch hier das Format für die graphische Darstellung eingehalten werden.
- 8. graphische Darstellung der Ergebnisse:** Die ermittelten Ergebnisse sollen mit D3 graphisch dargestellt werden. Eine mögliche Darstellung wäre ein gedoppeltes Balkendiagramm als Übersicht aller Weisheiten/Regeln, sowie Vektordiagramme für einzelne Regeln. Ob beide Versionen realisierbar sind,

hängt vom Gesamtaufwand ab. Falls die Darstellung mit D3 misslingt, könnte diese auch durch Matplotlib erstellt werden.

Erweiterungen:

- 9. Erstellung und Steuerung einer GUI:** Je nach Aufwand der Kernstruktur (Schritt 1.-8.) wäre es möglich das Analysetool über eine GUI zu steuern, anstelle einer Python-Shell innerhalb der CMD. Dies wäre sinnvoll, wenn mehrere Jahre oder unterschiedliche Orte auswählbar sein sollen. Hierbei ist zu beachten, dass durch diese Erweiterungen auch der Datenpool vergrößert werden muss, wodurch der Aufwand für die Kernstruktur ebenfalls wächst.

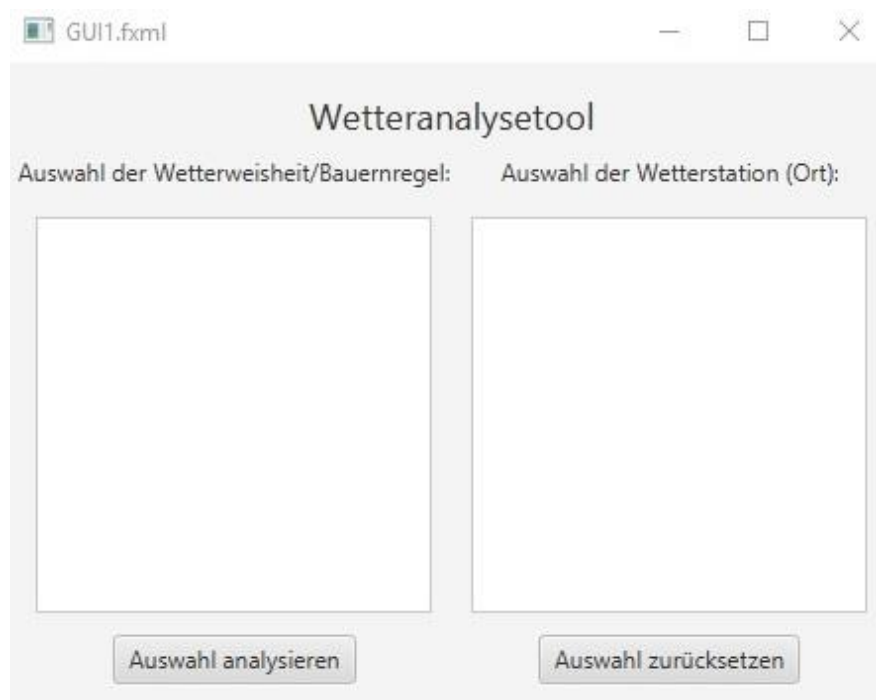


Abb.3 Beispiel für GUI zur Tool Steuerung