

Zusammenfassung: «Reveal the Pain»

Web Programming Lab

Patrick Bucher

22.09.2019

Inhaltsverzeichnis

Projektidee	1
Technische Dokumentation	2
Berechnung der Korrelation	2
Fazit	3
Reflexion	3
Arbeitsjournal	3
Quellen	3

Projektidee

Ein körperliches Leiden, z.B. Kopfschmerzen, kann viele Ursachen haben. Mithilfe von Self-Tracking einer Vielzahl möglicher Einflussgrössen können die Rohdaten zur Ursachenforschung erhoben werden. Mithilfe eines Phi-Koeffizienten kann anhand dieser Rohdaten die Korrelation zwischen Einflussgrössen und Krankheitssymptom berechnet werden. Damit können für ein Krankheitssymptom möglicherweise förderliche Einflüsse sondiert werden.

Die «Reveal the Pain»-Web-Applikation hilft einerseits beim täglichen Self-Tracking und führt andererseits die Berechnung der Korrelationskoeffizienten automatisch durch. Um etwa den Ursachen für Kopfschmerzen auf den Grund zu gehen könnten folgende Einflussgrössen getrackt werden: Alkoholkonsum, Bewegung, Stress, Wetterumschwung, Koffein, Rauchen, Lärm, usw.

Die Idee für diese Anwendung stammt aus dem Buch «Eloquent JavaScript», wo die Phi-Korrelation zwischen Pizzakonsum und der nächtlichen Verwandlung in ein Eichhörnchen untersucht wird (Haverbeke, Kapitel 4, Seite 66: *The Lycanothrope's Log*).

Technische Dokumentation

Berechnung der Korrelation

Der Benutzer hat während einer Woche seine Verhaltensweisen getrackt, um auf die Ursache für sein Rückenleiden zu kommen. Das Journal könnte folgendermassen aussehen (die Zielvariable *Rückenschmerzen* wird wie jeder andere Tag erfasst):

- Montag: Ausschlafen, Gewichtheben, Büroarbeit, Gartenarbeit, Rückenschmerzen.
- Dienstag: Büroarbeit, Alkoholkonsum, Einkaufen, Rückenschmerzen.
- Mittwoch: Büroarbeit, Müdigkeit, Einkaufen, Rückenschmerzen.
- Donnerstag: Ausschlafen, Gartenarbeit, Alkoholkonsum, Rückenschmerzen.
- Freitag: Büroarbeit, Müdigkeit, Einkaufen.
- Samstag: Ausschlafen, Spaziergang, Putzen, Gartenarbeit, Rückenschmerzen.
- Sonntag: Ausschlafen, Alkoholkonsum, Spaziergang, Velotour.

Nun möchte der Benutzer die Korrelationen verschiedener möglicher Einflussgrössen für die Zielvariable *Rückenschmerzen* berechnen lassen. Für jeden Tag (jede erklärende Variable im Journal, ausgenommen die Zielvariable *Rückenschmerzen*), wird für jedes Datum folgende Klassifizierung vorgenommen:

- Gruppe 1: Weder Ziel- noch erklärende Variable eingetragen.
- Gruppe 2: Zielvariable fehlt, erklärende Variable vorhanden.
- Gruppe 3: Zielvariable eingetragen, erklärende Variable fehlt.
- Gruppe 4: Ziel- und erklärende Variable eingetragen.

Für obiges Journal ergäbe sich dadurch folgende Klassifizierung der erklärenden Variablen *Büroarbeit* (EV) und Zielvariablen *Rückenschmerzen* (ZV):

	EV fehlt	EV vorhanden
ZV fehlt	1 (So)	1 (Fr)
ZV vorhanden	2 (Do, Sa)	3 (Mo, Di, Mi)

Die Gruppen können nun folgendermassen bezeichnet und mit Werten belegt werden:

- EV und ZV fehlt: $n_{00} = 1$
- EV vorhanden/ZV fehlt: $n_{01} = 1$
- EV fehlt/ZV vorhanden: $n_{10} = 2$
- EV und ZV vorhanden: $n_{11} = 3$

Der Phi-Koeffizient kann nun folgendermassen berechnet werden:

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{(n_{10} + n_{11})(n_{00} + n_{01})(n_{01} + n_{11})(n_{00} + n_{10})}}$$

Mit obigen Werten eingesetzt ergibt das:

$$\phi = \frac{3 \times 1 - 2 \times 1}{\sqrt{(2+3)(1+1)(1+3)(1+2)}} = \frac{1}{\sqrt{5 \times 2 \times 4 \times 3}} = \frac{1}{\sqrt{120}} = \underline{\underline{0.0913}}$$

Die Skala reicht von -1 (negative Korrelation) bis +1 (positive Korrelation). Für Werte mit einem Betrag ab 0.5 kann von einer Korrelation die Rede sein. Die erfassten Daten ergeben somit keine Korrelation zwischen Büroarbeit und Rückenschmerzen. Der Benutzer muss weiter Daten erfassen, um seinem Rückenleiden auf die Spur zu kommen.

Fazit

- Die Blockwoche und v.a. die Projektarbeit haben nicht nur mein Interesse an der Web-Entwicklung wieder geweckt, sondern auch an der funktionalen und asynchronen Programmierung.
- Zur praktischen Anwendung müsste die Möglichkeit einer Registrierung eingebaut werden.

Reflexion

- Die ursprünglich geplante Umsetzung mit der redundanten Datenspeicherung konnte umgangen werden. Dafür müssen mehr Abfragen gegen den Key-Value-Store getätigt und ausgewertet werden. Mit kleineren Datenmengen hat sich das nicht als problematisch ausgestellt.
- Ursprünglich wollte ich das Backend mit der Programmiersprache Go umsetzen, weil ich damit schon verschiedene kleine HTTP-Server-Anwendungen entwickelt habe. Nach der Einführung in Node.js und Express.js habe ich mich aber dazu entschlossen, diese Technologien einzusetzen, um dabei etwas Neues zu lernen.
- JavaScript und Express.js erlauben es die Endpoints sehr einfach und mit wenig Code zu definieren und zu implementieren. Gewöhnungsbedürftig war hingegen, dass viele Libraries (z.B. die Redis-Library zum Zugriff auf den Key-Value-Store) asynchron arbeiten, was jedoch nötig ist, um den globalen Event-Loop von Node.js nicht mit unnötig grossen Aufgaben zu blockieren. Hat man sich aber erst einmal an den Promise-Mechanismus gewöhnt, kann man damit schnell übersichtlichen und hochwertigen Code schreiben.

Arbeitsjournal

Quellen

Marjin Haverbeke: *Eloquent JavaScript. A Modern Introduction to Programming. Third Edition.*