Bericht der Gruppe „Ufo“

Eine Analyse der Abhängigkeit der Wetters auf vermeintliche Ufo-Sichtungen.

Lars Thomsen, Uzeyir Mammadov
Martin-Luther-Universität Halle-Wittenberg
[lars.thomsen][uzeyir.mammadov]@student.uni-halle.de

Abbildung 1: NUFORC.

ABSTRACT

KEYWORDS

Übung „Big Data Analytics“, Sommersemester 2021, Ufo

1 EINLEITUNG

Auf den ersten Blick klingen vermeintliche Ufo-Sichtungen immer nach an den Haaren herbeigezogenen Erfindungen und Lügengeschichten. Doch wie viel Wahrheit steckt in diesen Sichtungen, und lassen sich gegebenenfalls Muster in diesen erkennen?

Im Zuge unseres Projektes haben wir uns folgende Forschungsfrage gestellt: Existieren Abhängigkeiten zwischen vermeintlichen Ufo-Sichtungen, und welchen Einfluss nimmt das zum Zeitpunkt der Sichtung herrschende Wetter?

Der Bericht gliedert sich wie folgt: Kapitel 2 beschreibt die verwendeten Daten und Datensätze. In Kapitel 3 wird die Forschungsfrage und ihre Bearbeitung vorgestellt. Kapitel 4 diskutiert die Ergebnisse unserer Frage. Im Schlusskapitel geben wir unser Fazit und runden den Bericht ab.

2 DATEN

Die verwendeten Datensätze stammen aus einer Datenbank vom „National UFO Reporting Center“ (NUFORC). In diese Datenbank kann jede Person ihre vermeintliche Ufo-Sichtung – entweder online über ein Formular oder per Telefon – eintragen. Des Weiteren senden gewisse Messstationen (z.B. MADAR Nodes) auffällige Messdaten automatisch an die Datenbank. Um offensichtlichen Fakes entgegenzuwirken werden die eingereichten Daten vor der monatlichen Veröffentlichung von den Betreibern des Portals überprüft.

Jeder Eintrag des Datensatzes besteht aus dem Datum und der Uhrzeit der Sichtung, der Stadt sowie dem Kürzel des dazugehörigen Bundesstaates (ausschließlich für die USA), einer klassifizierten Beschreibung der Form des Ufos und gegebenenfalls einer kurzes Zusammenfassung und Beschreibung aus der Sicht des Einsenders.

Für diese Daten haben wir eine einfache Datenbank erstellt, um auch offline auf diese zugreifen zu können. Um die Daten einfacher verarbeiten zu können, bereiten wir diese während der Abfrage von unserer Datenbank auf und treffen eine Vorauswahl an brauchbaren Daten. Die Kriterien für die Vorauswahl wurden durch stichprobenartige Abfragen festgelegt. Dem entsprechend werden Einträge von vorhin beschriebenen externen Messstationen, Einträge mit fehlenden Inhalten oder „?“ als Inhalt nicht betrachtet. Der Datentyp aller Attribute ist der Einfachheit halber nur string. Zu einem späteren Zeitpunkt werden Datum und Uhrzeit in ein datetime-Format überführt. Mit dem Wissen aus den stichprobenartigen Abfragen legen wir und auf die zwei gängigsten Formate fest – 'm/d/y H:M' und 'm/d/y'. Andere in dem Datensatz vorkommenden Formate oder von den Einsendern eigenständige Angaben beachten wir nicht. Die aufbereiteten Daten werden letztendlich durch das Tripel `datetime`, `city` und `state` beschrieben.

Als Quelle für die Wetterdaten haben wir uns letztendlich für „Meteostat“(Link) entschieden. Auf diese Daten wird über die dazugehörige Python-Library zugegriffen. Essenziell für unser Projekt sind die Attribute `tsun` (Anzahl der Sonnenminuten) und `coco` (Klassifizierter Zustand des Himmels). Die Daten sind für unsere Vorhanben bereits eine ausreichende Qualität, sodass in diesem Fall keine weitere Aufbereitung und Säuberung der Daten nötig ist.

Tabelle 1: Kennzahlen des verwendeten Datensatzes.

Datensätze	Anzahl
Gesamt	96 924
Davon einzigartige Orte	25 234
Orte mit Wetterdaten, gesamt	2 020
Davon mit Sonnenminuten pro Stunde	26
Davon mit Sonnenminuten pro Tag	116
Davon mit condition codes	1 886

3 DIE ERSTEN VERSUCHE

Die erste Idee war es, die Wetterdaten für unsere Forschungsfrage von den „National Centers for Enviromental Information“ (NCEI) zu beziehen. Diese Datensätze beinhalten für unsere Analyse zwei wesentliche Attribute: `HourlySkyCondition` und `HourlyVisibility`. Diese beiden Werte beschreiben zum Ersten den klassifizierten Zustand des Himmels (z.B. Regenwolken, Schleierwolken, Nebel etc.) und zum Zweiten die prozentuale Bedecktheit des Himmels. Die Liste aller verfügbaren Stationen sind ebenfalls beim NCEI verfügbar(Link Stationen).

Um die Daten einer Ufo-Sichtung mit den dazugehörigen Wetterdaten zu verknüpfen, wird das jeweilige eindeutige `city, state`-Tupel der am nahestehendsten Wetterstation zugeordnet. Um die Ergebnisse nicht zu sehr zu verfälschen, wurde das zulässige Einzugsgebiet einer Wetterstation auf 20 Kilometer, mit der Station als Zentrum, begrenzt.

Diese Idee konnte leider nicht umgesetzt werden. Es standen zwar zwei Versionen der API vom NCEI zur verfügung, allerdings greifen diese auf unterschiedliche Stationen zurück. Dabei waren zwei Probleme die Hauptursache für das Aufgeben dieser ersten Idee: Zum Ersten waren die internen Bezeichnungen der Stationen bei Version 1 der API verschieden zu denen der uns zur verfügung stehenden Liste. Eine einheitliche Liste aller Stationen, welche durch Version 1 der API bedient werden war nicht verfügbar. Zum Zweiten wurde von Version 2 der API eine verschiedene zur bereits verarbeiteten Liste an Stationen verwendet, welche nur unter Umständen einsehbar war. Somit war es nicht möglich mit den verwendeten Datensätzen die gestellte Forschungsfrage zu bearbeiten.

4 FORSCHUNGSFRAGE

Nachdem wir uns dazu entschlossen haben, die APIs des NCEI nicht mehr zu verwenden, haben wir uns auf die Suche nach neuen APIs und Anwendungen gemacht, um an für uns passende Wetterdaten zu kommen. Der Service von Meteostat passte dabei perfekt in unsere Planung. Die Abfrage von Wetterdaten erfolgte durch die Python-Library von Meteostat und bekam als Input den Zeitraum der gewünschten Daten, sowie den Längen- und den Breitengrad des gewünschten Ortes.

Um die entprechenden geographischen Koordinaten für die Abfragen zu erhalten, wurde die API von „MapQuest“ verwendet. Als Parameter wurden das `city, state`-Tupel aus dem bereinigten Datensatz übergeben. Den Output bildeten ein Tupel aus den resultierenden Längen- und Breitengraden der gewünschten Orte. Um redundante Anfragen zu vermeiden, wurden zunächst einzigartige `city, state`-Paare aus dem Datensatz extrahiert. Somit reduzierte sich die Anzahl der zu bewältigenden Geocoding-Anfragen von 96 924 auf 25 234.

Mit den abgeschlossenen Geocoding-Anfragen waren alle benötigten Daten für die Anfragen an die Meteostat-Datenbank lokal verfügbar. Als Basis für die Abfragen galten die Daten aller Sichtungen „ `sightings.csv` “ mit dem Tripel `datetime, city, state`. Über das Tupel `city, state` wurden die entsprechenden Längen- und Breitengrade aus „ `cities_coords.csv` “ für die entsprechende Anfrage als Punkt gespeichert. Als Richtwert für den Zeitpunkt der abzufragenden Wetterdaten gilt das Attribut `datetime`. Da Meteostat für jede Anfrage einen Start- und Endzeitpunkt benötigt, wird als Endzeitpunkt eine Sekunde auf den Startzeitpunkt addiert.

Tabelle 2: Datensätze

Datensatz	Anzahl Ergebnisse
hourly sunshine	26
daily sunshine	116
condition codes	1886

Tabelle 3: Condition Codes

Condition Code	Anzahl
0	15
1	227
2	603
3	363
4	93
5	186
7	219
8	51
9	9
12	1
14	42
15	2
16	1
17	9
18	23
25	35
26	5
27	2

Für die eigentliche Abfrage der Wetterdaten wurden zwei varianten benutzt. Zum Ersten sich stündlich aktualisierende Daten und zum Zweiten sich täglich aktualisierende Daten. Der Ablauf der Anfrage ist in beiden Fällen identisch, der einzige Unterschied ist, dass die Condition Codes für den Himmel nur in den stündlichen Daten vorhanden waren. Da die Stationen nicht identisch sind, lieferte nicht jede Station für jedes relevante Attribut die entsprechenden Messwerte. Es wurden dementsprechend nur die Ergebnisse der Anfragen gespeichert, wenn einer der drei relevanten Werte vorhanden war. Die Ergebnisse der Anfrage bestanden letztendlich aus der Liste `city, state, time_date, sun_hourly, sun_daily, condition_code`, welche im letzten Schritt visualisiert wurden.

Die Visualisierungen erfolgten alle über die „matplotlib“-Library. Visualisiert wurden die drei Messwerte als Graph, die Condition Codes als Histogramm und Tortendiagramm sowie die stündlichen und täglichen Sonnenminuten als Boxplots.

5 EVALUATION

Some evaluation section if appropriate. You might want to refer to some table with results in this section (e.g., to Table 1).

6 FAZIT