Fuzzylogik – Non-Brand-Reporting im SEO

Fuzzylogik – Non-Brand-Reporting im SEO – websiteboosting.com

Viele SEO-Reports zeichnen Woche für Woche dasselbe Bild. Die Kennzahlen organische Sessions, Bounce-Rate, Umsatz und diverse Sichtbarkeitsindizes werden gerne undifferenziert an das Management berichtet. Nur die wenigsten wissen, dass diese Kennzahlen nicht ausreichen, um die Performance von SEO…
Viele SEO-Reports zeichnen Woche für Woche dasselbe Bild. Die Kennzahlen organische Sessions, Bounce-Rate, Umsatz und diverse Sichtbarkeitsindizes werden gerne undifferenziert an das Management berichtet. Nur die wenigsten wissen, dass diese Kennzahlen nicht ausreichen, um die Performance von SEO richtig zu bewerten. Je nach Größe und Wachstum der eigenen Marke korreliert ein großer Anteil des SEO-Erfolgs mit anderen Marketingaktivitäten, die die SEO-Erfolgsmessung maßgeblich verzerren. Um diesen verfälschten Blick zu schärfen, ist es zwingend notwendig, die SEO-Ergebnisse differenziert zu betrachten. Eine Aufteilung des organischen Traffics zwischen Brand und Non-Brand ist hier ein erster richtiger Schritt.

Die Wege eines Kunden von der ersten Produktwahrnehmung bis hin zum finalen Kauf (Customer Journey) werden zunehmend komplexer. Diverse Marketingaktivitäten nehmen maßgeblichen Einfluss auf das Kaufverhalten der Kunden. Abbildung 1 soll diese Abhängigkeiten veranschaulichen.

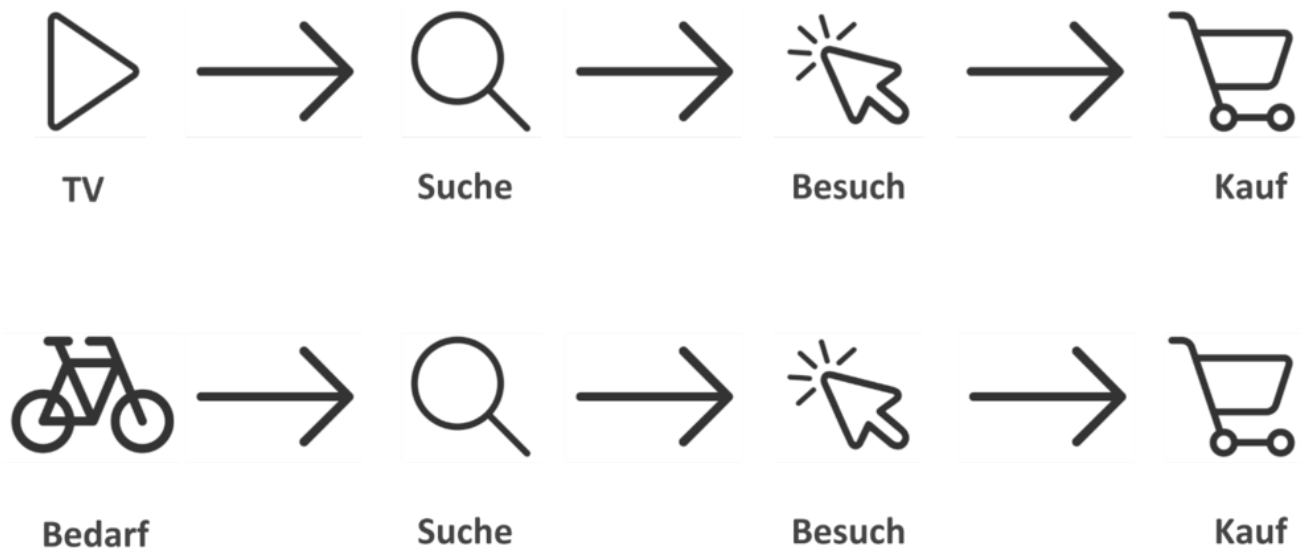

Abbildung 1: Customer Journeys

In der ersten Customer Journey sieht der Kunde einen TV-Werbespot und sucht anschließend auf seinem Smartphone nach einem generischen Keyword plus Markenname. Mit dem Vince-Update aus dem Jahr 2009 erhalten Marken bei Google eine höhere Relevanz. Die Suchmaschine erkennt anhand von Suchbegriffen eine womöglich präferierte Marke und spielt diese bei entsprechenden Suchanfragen auf den oberen Positionen aus. Top-Positionen sind bei diesen Suchanfragen garantiert. SEOs sollten sich dementsprechend auf die Optimierungen im Non-Brand-Bereich konzentrieren.

Die zweite Customer Journey zeigt einen Kunden, der sein Fahrrad reparieren möchte. Es sucht nach passenden Ersatzteilen anhand diverser generischer Suchbegriffe und technischer Bezeichnungen. Dieser Kunde ist nicht voreingenommen und besitzt keine Markenpräferenz. Die Wahrscheinlichkeit, dass hinter dieser Customer Journey ein Neukunde steckt, ist hoch. Genau diese Suchanfragen sind es, die sich im Non-Brand-Traffic widerspiegeln und anhand derer der Performance-Marketing-Kanal SEO gesteuert und gemessen werden sollte.

Vorbereitung und Daten, Daten, Daten

Im Statistikunterricht wurde eines klar: Je größer die Stichprobe ist, desto genauer ist das Endresultat. Google stellt für Webmaster unter dem Namen Google Search Console (GSC) ein Tool zur Verfügung, mit dem sich die Performance der eigenen Website in der Google-Suche messen lässt. Öffnet man den Performance-Report in diesem Tool, ist es leider nur möglich, maximal 1.000 Zeilen eines konfigurierten Filters als CSV herunterzuladen. Bei bekannten Marken stecken in diesem Datensatz bereits viele Falschschreibweisen, die es möglichst genau zu kategorisieren gilt, um den Non-Brand-Traffic am Ende präzise bestimmen zu können.

Mehr Daten erhält man mit der von der GSC zur Verfügung gestellten Programmierschnittstelle (eng.: Application Programming Interface, kurz: API), mit der man mehr als 25.000 Datensätze pro Filter bekommt. Dieser umfangreiche Datensatz sorgt für eine höhere Genauigkeit im Endergebnis.

Da die Schnittstelle keine grafische Oberfläche zur Verfügung stellt, mit der die Daten gefiltert werden können, geschieht dies durch ein JSON(JavaScript Object Notation)-Objekt. Für diese Auswertung interessieren lediglich ein Start- und Enddatum sowie die Dimension Query (deutsch Suchbegriff). Das aktuelle RowLimit seitens Google liegt bei 25.000. Sollte dieser Filter mehr als 25.000 Zeilen liefern, können durch einen Offset und Paginierung weitere Zeilen abgefragt werden.

```
request = {
    'startDate': '2018-11-01',
    'endDate': '2018-11-08',
    'dimensions': ['query'],
    'rowLimit': 25.000
}
```

Abbildung 2: JSON-Filter

Das Ergebnis dieser Abfrage (Request) ist eine Liste (“rows”:

[]) mit Objekten ({…}). Jedes Objekt steht für eine Zeile, so wie sie im Frontend der Google Search Console zu sehen ist. Ein Objekt besitzt die in Abbildung 3 gezeigte Struktur.

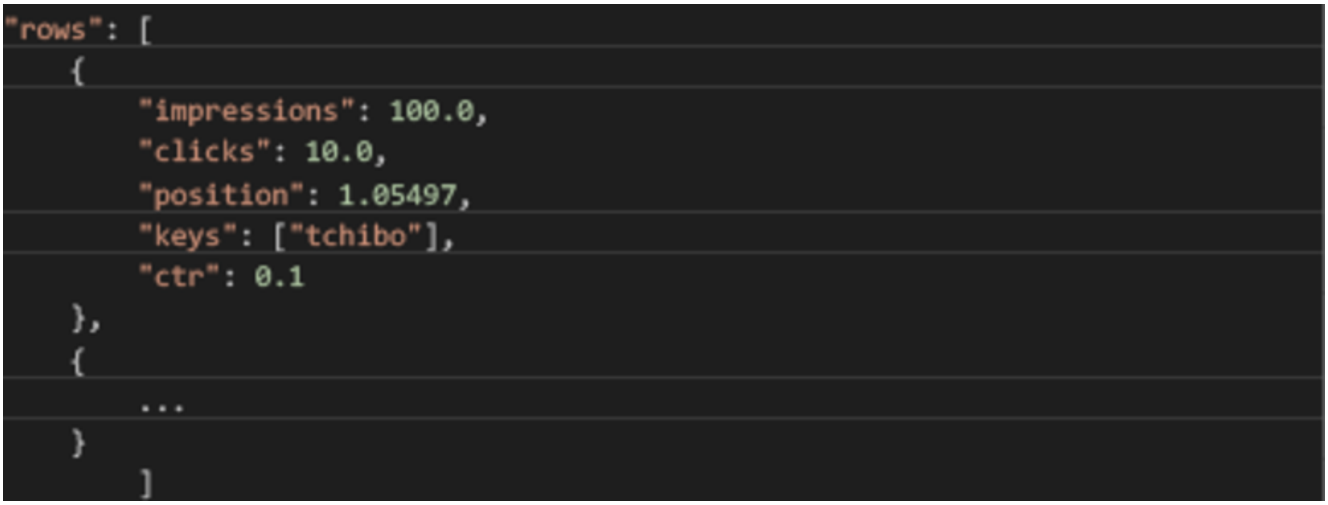

Abbildung 3: Ergebnis der API-Abfrage

Google Search Console API mit Python abrufen

Um die Google Search Console API abfragen zu können, sind zwei Dinge wichtig: Einrichtung eines Dienstkontos unter console.cloud.google.com und die Authentifizierung gegenüber der API mit einem Client.

Zunächst wird nach erfolgreichem Log-in unter console.cloud.google.com ein Projekt und unter dem Menüpunkt IAM & Verwaltung/Dienstkonto ein neues Dienstkonto angelegt, bei dem darauf zu achten ist, dass ein JSON-Key generiert wird (siehe Abbildung 4). Dieser Key dient dem Programm als Authentifizierung.

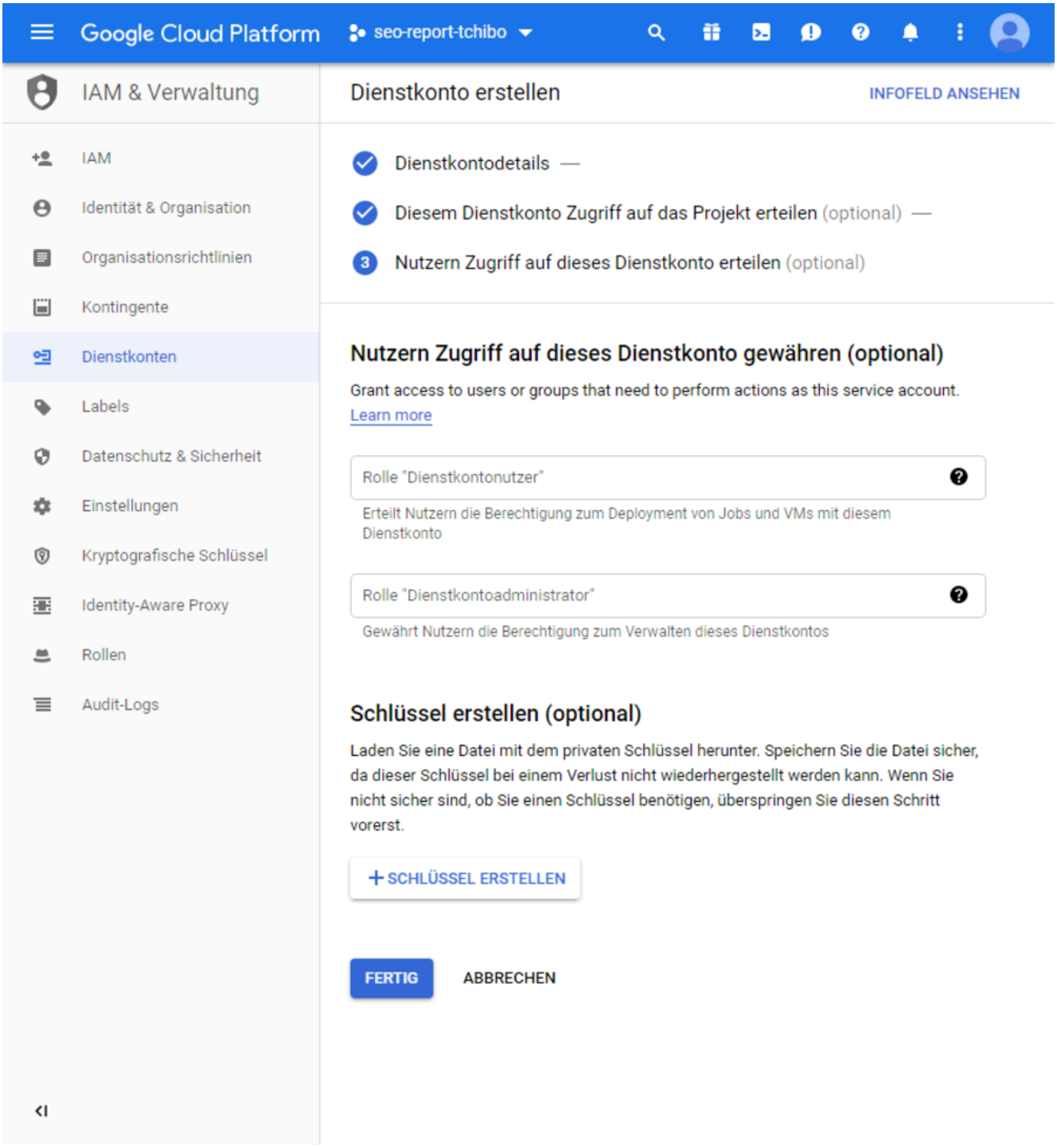

Abbildung 4: Service-Account erstellen

Die E-Mail-Adresse des JSON-Keys muss anschließend als lesender Nutzer hinter die entsprechende Google Search Console Property gelegt werden. Nachdem der Service-Account erstellt und die E-Mail-Adresse eingerichtet ist, lässt sich die im Git Repository hinterlegte Datei api_requests.py ausführen. Dabei ist darauf zu achten, dass der Name des JSON-Key angepasst wird und die in Zeile 21 hinterlegte ‚siteURL' mit der URL der Property übereinstimmt. Die Zeilen 14–19 repräsentieren den in Abbildung 2 zu sehenden JSON-Filter.

Traffic klassifizieren

Mehr als 25.000 Datensätze händisch nach Brand- und Non-Brand-Begriffen zu klassifizieren, wäre zu aufwendig. Ein programmatischer Ansatz ist bei dieser Auswertung nicht von der Hand zu weisen. Um Brand-Falschschreibweisen zu definieren, könnte man auf die Idee kommen, eine Regular Expression (kurz: regex) zu definieren. Dies wird die Brand-Begriffe sicherlich sehr präzise definieren, jedoch rutschen möglicherweise auch ähnliche Non-Brand-Keywords in diesen Filter, die das Bild erneut verzerren. Beispielsweise würde die regex r\t(.)*ch(.)* die Brand-Falschschreibweise „tschibo" sowie das Wort „tisch" als Brand-Keyword klassifizieren, was falsch ist. Regular Expressions sind also zu fehleranfällig, um ein klares Bild des Brand- und Non-Brand-Traffics zu erhalten.

Abhilfe schafft hier eine unscharfe Suche (oder auch Fuzzy-Suche), die nicht darauf aus ist, einen exakten Suchbegriff in einer Zeichenkette zu finden, sondern ähnliche Suchbegriffe. Ein bekannter Algorithmus in diesem Kontext ist die sogenannte Levenshtein-Distanz, welche u. a. in der Rechtschreibprüfung oder in der Duplikat-Erkennung angewandt wird.

Für die Programmiersprache Python steht das Modul „fuzzywuzzy" zur Verfügung, mit dem die Levenshtein-Distanz zwischen zwei Wörtern bestimmt werden kann. Möchte man beispielsweise die Ähnlichkeit zwischen den Strings „this is a test" und „this is a test!" bestimmen, reicht ein simpler Aufruf der Methode ratio() am importierten Fuzz-Objekt. Die Levenshtein-Distanz pendelt zwischen den Grenzwerten 0 und 100, wobei 100 eine exakte Übereinstimmung definiert.

```
>>> fuzz.ratio("this is a test", "this is a test!")
97
```

Abbildung 5: Levenshtein-Distanz in Python

Sample Daten und Programmcode

Um den Artikel übersichtlich zu halten, analysiert dieser Beitrag einen Datensatz von 10 Zeilen mit fiktiven Keyword- und Klickdaten (siehe Tabelle 2).

Da Suchende bei ihrer Suchanfrage den Markenbegriff entweder voranstellen oder anhängen, müssen die Keywords zunächst in einzelne Suchwörter zerlegt, sortiert und Dubletten entfernt werden. Anschließend wird zu jedem Suchwort die Levensthein-Distanz (LD) zu dem Begriff „tchibo” bestimmt (siehe Tabelle 1).

Suchwort	Levenshtein-Distanz	LD – 67
Tchibo	100	33
Tchibio	92	25
Tchibop	92	25
Tchobi	67	0
Tisch	55	-12
Schuhe	33	-34
Brotbackschale	30	-37
duschschlauch	21	-46
bett	20	-47
mode	20	-47
online	17	-50
bademantel	12	-55
bettgestell	12	-55
kaffee	0	-67
kaufen	0	-67

Tabelle 1: Suchwörter mit Levensthein-Distanz

Interessant ist der Übergang der LD zwischen 55 und 67. Ab dieser Schwelle können Keywords nach subjektiver Wahrnehmung

Brands zugeordnet werden. Besitzt eine Kernmarke mehrere Untermarken, müssen die Schwellenwerte entsprechend pro Brand-Begriff festgelegt werden. Diese mehrdimensionale Analyse würde jedoch den Umfang dieses Beitrags sprengen. Beispielsweise besitzt die Kurzschreibweise TCM (Tchibo Certified Merchandise) bei Tchibo Brand-Charakter. Der Schwellenwert für diesen Begriff ist aufgrund der geringen Zeichenanzahl relativ hoch.

Um Keywords (insbesondere über mehrere Markenbegriffe) besser zuordnen und vergleichen zu können, wird der Schwellenwert von der LD subtrahiert. Dadurch können alle Einzelwörter des Keywords mit einer modifizierten Levenshtein-Distanz von größer/gleich 0 eindeutig Brand zugeordnet werden.

```
for i in data['rows']:
    res = []
    for x in i['keys'][0].split(' '):
        res.append(fuzz.ratio(x, 'tchibo') -67)
        i['brand'] = max(res) >= 0
```

Abbildung 6: Programmcode zur Berechnung der LD

Nachdem der Schwellwert (eng. Threshold) bestimmt wurde, gilt es, den Testdatensatz zeilenweise zu iterieren (siehe **Abbildung 6**, Zeile 1). Für jede Zeile wird eine neue Liste (res) angelegt, die am Ende die modifizierte LD pro Suchwort beinhaltet. In Zeile 3 wird jedes Keyword am Leerzeichen getrennt. Für die daraus resultierenden Suchwörter wird die modifizierte LD berechnet und in der Liste res gespeichert. Ist das Maximum dieser Liste größer oder gleich 0, wird die Zeile um die Spalte Brand erweitert und erhält den Wert True (Zeile 5 & Abbildung 7) andernfalls den Wert False. Das Ergebnis dieses Durchlaufs ist der Tabelle 2 zu entnehmen.

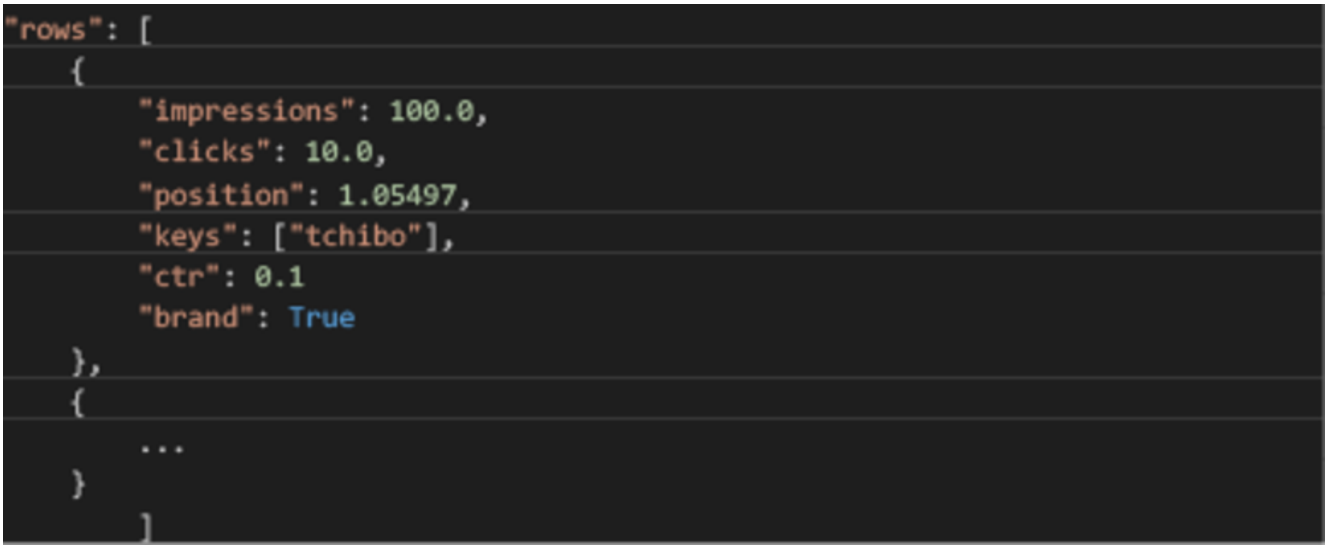

Abbildung 7: JSON Objekt um Spalte „brand“ erweitert

Keyword	Klicks	Matrix (Result)	Brand
tchibo duschschlauch kaufen	145	[**33**, -46, -67]	True
tchibo mode online kaufen	356	[**33**, -47, -50, -67]	True
tchibop kaffee kaufen	478	[**25**, -67, -67]	True
bett online	25	[-47, -50]	False
tchibio bademantel kaufen	789	[**25**, -55, -67]	True
bettgestell kaufen	389	[-55, -67]	False
schuhe online kaufen	323	[-34, -50, -67]	False
tisch kaufen	35	[-12, -67]	False
tchobi tisch kaufen	398	[**0**, -12, -67]	True
tchibo brotbackschale kaufen	334	[**33**, -37, -67]	True

Tabelle 2: Ergebnis des Programmcodes

Non-Brand-Verhältnis berechnen

```
def calc_non_brand_ratio(data):
    nonbrand_clicks = 0
    brand_clicks = 0
    for i in data:
        if i['brand']:
            brand_clicks += i['clicks']
        else:
            nonbrand_clicks += i['clicks']
    return (brand_clicks, nonbrand_clicks, (nonbrand_clicks/(nonbrand_clicks +
brand_clicks)))
```

Abbildung 8: Programmcode zur Berechnung des Non-Brand-Verhältnisses

Die Bestimmung des Non-Brand-Verhältnisses ist mathematisch relativ überschaubar. Der nun um die Spalte Brand erweiterte Datensatz (siehe Tabelle 2) wird zeilenweisen iteriert. Wenn der Brand-Wert einer Zeile den Wert True aufweist, werden die Klicks dieser Zeile auf die Variable brand_clicks addiert. Bei dem Wert False werden die Klicks entsprechend auf die Variable nonbrand_clicks addiert. Das Non-Brand-Verhältnis in dem festgelegten Zeitraum ist dann entsprechend:

$$NonBrand\ Ratio = \frac{nonbrand_{clicks}}{nonbrand_{clicks} + brand_{clicks}}$$

$$NonBrand\ Ratio = \frac{772}{772 + 2500} = 0,2359$$

$$NonBrand\ Ratio\ = 23,59\%$$

Fazit und Zusammenfassung

Tchibo hat in Deutschland eine Markenbekanntheit von 99 %. Diese enorme Markenbekanntheit verursacht unzählige Falschschreibweisen und „Vertipper-Keywords“ auf deutschen Tastaturen. Mit dieser Methodik haben wir eine valide Grundlage geschaffen, die Erfolge unserer SEO-Abteilung fair zu bewerten.

Dass diese differenzierte Betrachtung nicht ausreicht, um die Performance von SEO zu beurteilen, ist eindeutig. Jedoch ist diese Methodik eine Grundlage für nachfolgende stichwortartige Gedankengänge, um den Non-Brand-Blick weiter zu schärfen:

- Verbindung zwischen GSC und Google-Analytics-Daten via URL, um Non-Brand-Umsatz definieren zu können.
- System sehr fehleranfällig bei generischen Marken wie bspw. „stoffe.de" oder „fahrrad.de".
- Non-Brand-Traffic pro URL/Kategorie und Zeiteinheit bestimmen, um den Erfolg von SEO-Tests zu messen, bspw. Änderungen der CTR im Non-Brand-Traffic bei der Änderung des Meta-Titles und der Meta-Description auf bestimmten URL.
- Non-Brand-Ratio nicht sehr aussagekräftig! Interessant wird es, wenn Non-Brand-Traffic ins Verhältnis zu Suchvolumen-Potenzial gesetzt wird.
- Report automatisiert pro Zeiteinheit ausführen und grafisch festhalten.