https://www.netidee.at/preiswert

Preis.wert User Guide

Dieser User Guide beschreibt die **Visualisierung und Analyse eines Datensatzes**, der mit dem preis.wert Scraping Tool zur Erfassung von dynamischer und personalisierter Preisgestaltung, gewonnen wurde. Für eine Installationsanleitung und Details zu Konfiguration und Erstellung eines Datensätzen verweisen wir auf den **preis.wert Technical Guide**. Die im Projekt erhobenen Datensätze werden in diesem Dokument exemplarisch, zur Beschreibung der Funktionalität aus Endanwendersicht, genutzt und über das Code-Repository in anonymisierter Form vollständig bereitgestellt. Die Funktionalität wird anhand der beschriebenen Beispieldatensätze erklärt und potentiell auffälliges Verhalten interpretiert.

Inhalt

Installationsanleitung

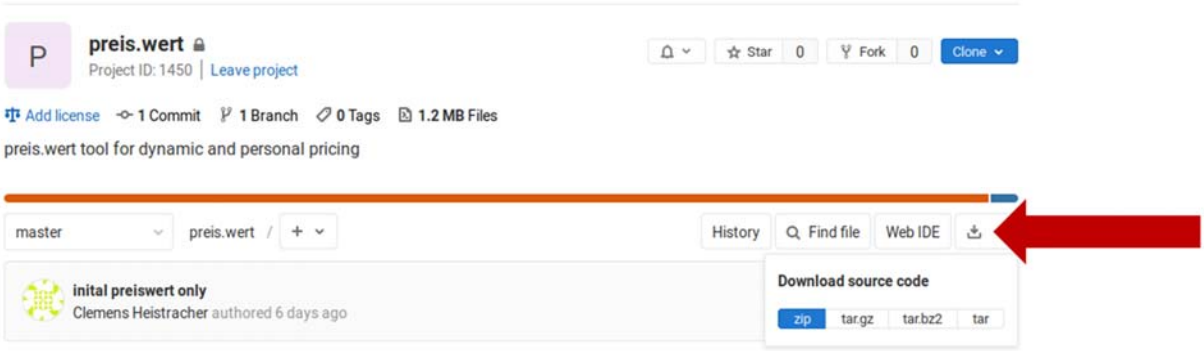
Falls Sie bereits der Anleitung des **preis.wert Technical Guide** gefolgt sind kann dieser Abschnitt übersprungen werden.

Clone git repository:

```
git clone https://git-service.ait.ac.at/im-public/preis.wert
```

https://www.netidee.at/preiswert

Oder Download als zip und extrahieren:



Voraussetzung: Python >= 3.3

```
python3 --version
```

Pip installieren.

```
sudo apt update
```

```
sudo apt install python3-pip python3-dev
```

Pip Installation überprüfen.

```
pip3 --version
```

Benötigte Bibliotheken installieren.

```
pip3 install --user -r requirements.txt
```

Abmelden und Anmelden des Users im Betriebssystems.

Erste Schritte mit Jupyter Notebooks

Bei Jupyter Notebooks handelt es sich um interaktive Dokumente in den sowohl Code als auch Ergebnisse wie Abbildungen und Tabellen dargestellt werden können. Das Dokument gliedert sich in einzelne Zellen, die Dokumentation oder Code enthalten und den jeweiligen Output. Grundsätzlich werden Notebooks so gestaltet, dass mit einem Klick alle Zellen chronologisch durchlaufen werden können. Zusätzlich können einzelne Zellen durch „STRG“ + „ENTER“ ausgeführt werden – dadurch wird eine interaktive Anpassung und Evaluierung der einzelnen Zellen möglich. Dabei ist zu erwähnen, dass die Zellen nicht vollständig unabhängig voneinander sein müssen. Im Zweifel ist es immer möglich alle vorherigen Zellen von oben nach unten Auszuführen, um den passenden Zustand herbeizuführen.

Um das preis.wert Evaluierungsnotebook zu starten geben Sie im Terminal „jupyter notebook“ und des Pfad zu „preiswert_evaluation.ipynb“ ein.

```
jupyter notebook evaluation/preiswert_evaluation.ipynb
```

In einem Jupyter Notebook werden einzelne Zellen durch „STRG“ + „ENTER“ ausgeführt. Dazu muss eine einzelne Zelle durch einen Mausklick ausgewählt werden (blaue Markierung).

Anpassung der Dateipfade

In Zelle 2 wird der Pfad zu den Ergebnissen des Scapings (z.B. results.csv) angegeben. Bei Bedarf kann auch der Pfad, unter dem Abbildungen und Tabellen gespeichert werden, verändert werden.

Um zu testen ob eine Datei gelesen werden könnte führen Sie die ersten drei Zellen aus. Bei korrekter Eingabe des Pfades werden die ersten fünf Zeile des Datensatzes unter Zelle drei dargestellt.

Evaluation preis.wert

This notebook is used for post-processing and visualisation of data scraped for preis.wert. For a detailed documentation of this notebook see preis.wert **User Guide**. For documentation on the preis.wert scraper see preis.wert **Technical Guide**.

Table of contents

Load Dataset

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import re
import os
```

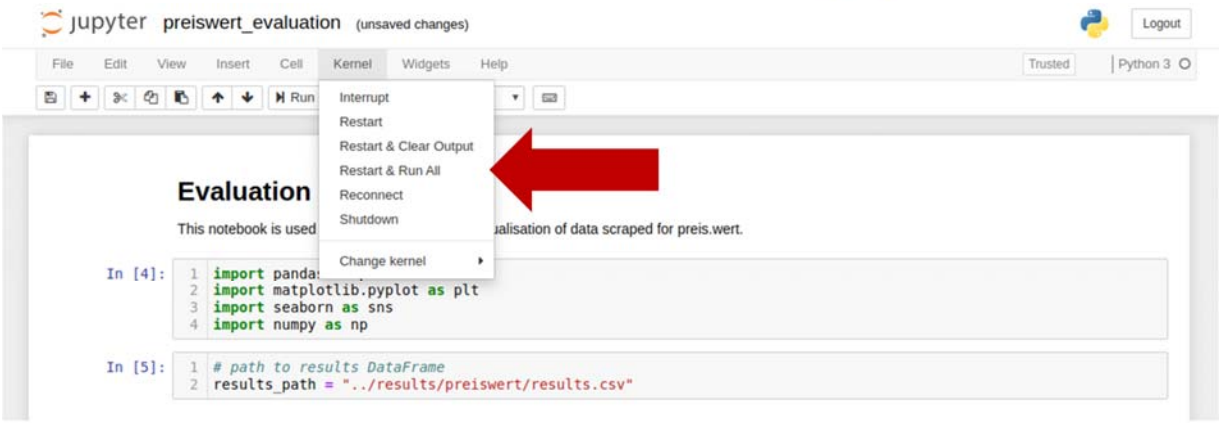
In [2]:
```
# path to results DataFrame
results_path = "../results/preiswert/results.csv"

# where to store figures and tables
output_path = "../results/preiswert/evaluation"
if not os.path.exists(output_path):
    os.makedirs(output_path)
```

In [61]:
```
df = pd.read_csv(results_path)
df = df.drop_duplicates()
df.reset_index()
df.timestamp_parsed = pd.to_datetime(df.timestamp_parsed)
df.head()
```

Out[61]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	product	site_name	url	price	timestamp_parsed	screenshotname	html_name	git_hash	cookies	cookies_sen
0	0	0	0	galaxy_s9	Shop_A	dummy	519,00	2019-11-05 11:10:55.033116	dummy	dummy	2d7b9a7	dummy	Tru
1	1	1	0	switch	Shop_B	dummy	309.00	2019-11-05 11:10:58.823388	dummy	dummy	2d7b9a7	dummy	Na
2	2	2	0	switch	Shop_B	dummy	309.00	2019-11-05 11:11:02.157856	dummy	dummy	2d7b9a7	dummy	Tru
3	3	3	0	switch	Shop_B	dummy	309.00	2019-11-05 11:11:05.446231	dummy	dummy	2d7b9a7	dummy	Na

Wenn der Pfad des Datensatzes korrekt ist können Sie alles Zellen des Notebooks ausführen. Dazu klicken Sie auf „Kernel“ → „Restart & Run All“.



Beispieldatensatz 1 – Analyse der erfolgreichen Abfragen

Bei diesem Datensatz handelt es sich um die Daten eines einzelnen Scraping Vorganges und dient als Beispiel für eine Kontrolle der Konfiguration. Aus diesem Datensatz können noch keine Erkenntnisse zu dynamischer oder personalisierter Preisgestaltung gewonnen werden da er einem einzelnen Messpunkt entspricht. Ein solcher Datensatz kann erzeugt werden durch den einmaligen Aufruf von:

```
sudo python3 start_preiswert.py
```

Um diesen Datensatz zu verwendend, ändern Sie results_path in Zelle 2 zu

```
results_path = "../results/preiswert/results_single_scrape.zip"
```

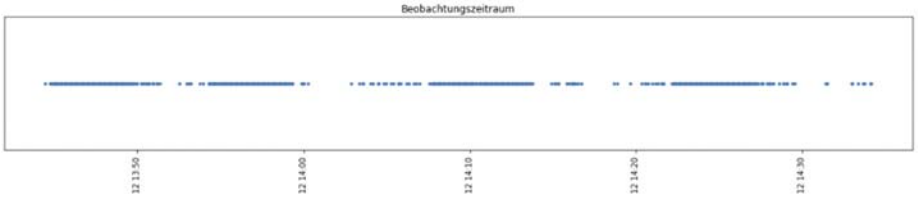
Anschließend können Sie alle Zellen ausführen („Kernel“ → „Restart & Run All“)

Übersicht über den Datensatz verschaffen

Beobachtungszeitraum

Hier werden die Zeitpunkte aller Abfragen erfasst. Dadurch wird der Beobachtungszeitraum dargestellt.

```
Dataset start: 2019-11-12 13:44:27.794772
Dataset end: 2019-11-12 14:34:09.435982
Number of extracted prices: 465
```



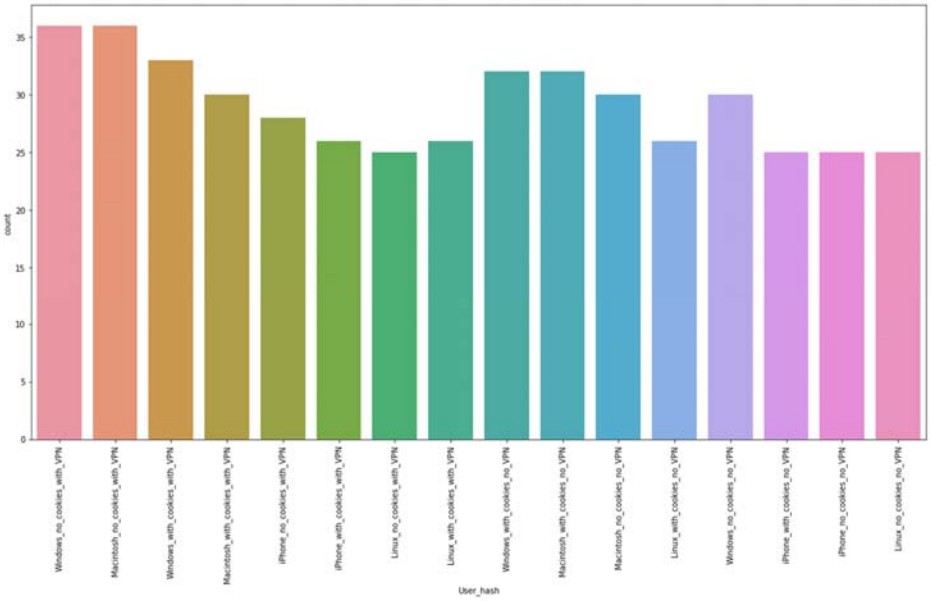
Aktuellster Preis

Der aktuellste Preis wird in dieser Tabelle dargestellt.

	site_name	product	price
0	Shop_A	Jack Wolfskin Funktionsjacke »365 INFLUENCER J...	139.99
16	Shop_A	switch	369.99
13	Shop_A	samsung_fernseher	1269.99
11	Shop_A	matebook	1699.99
7	Shop_A	Superdry Kapuzensweatshirt »VINTAGE LOGO POP E...	69.95
6	Shop_A	Seidensticker Hemd Damen (hellblau)	59.99
9	Shop_A	galaxy_s9	519.00
4	Shop_A	Nike Kapuzensweatshirt »Team Club«	37.89
3	Shop_A	Lee Damen Jeans Marion Straight Legs	89.95
1	Shop_A	Jeans Herren, G-Star Raw Slim Fit Denim	54.99
5	Shop_A	Reebok Classic, weiße Sneakers, Club C85	49.99
17	Shop_B	switch	297.00
10	Shop_B	galaxy_s9	649.00
12	Shop_B	matebook	1699.00
14	Shop_B	samsung_fernseher	1289.00
8	Shop_D	Superdry Kapuzensweatshirt »VINTAGE LOGO POP E...	79.99
2	Shop_D	Jeans Herren, G-Star Raw Slim Fit Denim	119.95
15	Shop_E	samsung_fernseher	1268.30
18	Shop_E	switch	344.65

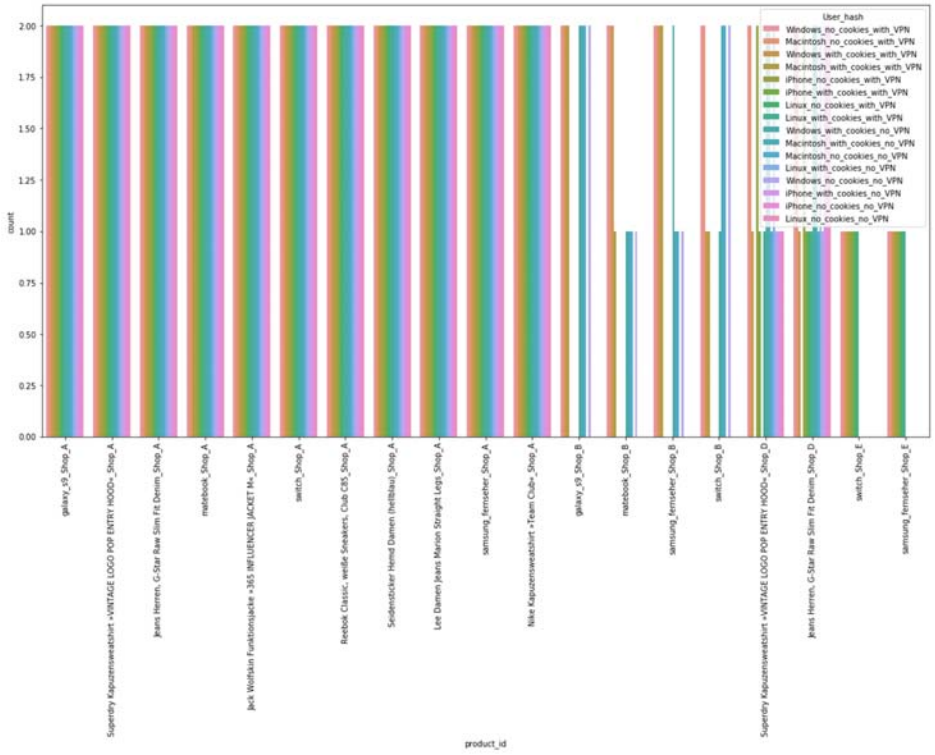
Erfolgreiche Abfragen je Identität

Hier wird die Anzahl der erfolgreichen Preisabfragen je Identität in einem Balkendiagramm dargestellt. In diesem Datensatz wurden 18 Produkte, 2 VPNs und 4 User Agents definiert. Da für jede konfigurierte VPN Verbindung eine Kontrollabfrage gestartet wird, wird jedes Produkt zwei Mal von jeder Identität abgefragt. (Bei den Identitäten wird nur berücksichtigt ob, und nicht welches, VPN verwendet wurde). Da nicht alle Abfragen 36 counts erreichten, waren nicht alle Abfragen erfolgreich. Da es aus verschieden zufälligen Gründen zu Fehlern im Verbindungsaufbau kommen kann, ist dies alleine jedoch nicht aussagekräftig.



,

Die obige Abbildung kann auch auf die einzelnen Produkte einzelner Shops angewandt werden. Dabei wird ersichtlich, dass einzig bei Shop_A alle Abfragen erfolgreich waren.

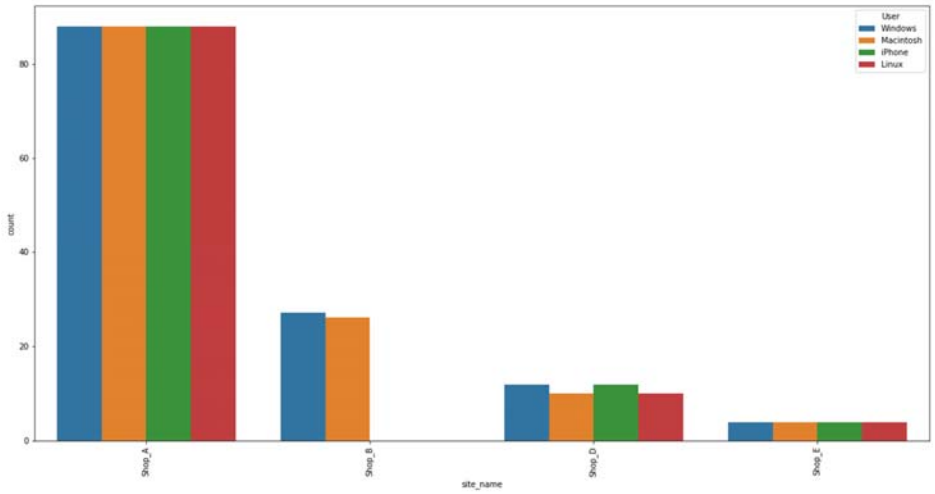


Diskriminierung nach User Agents

Eine Aufschlüsselung der User_Agents nach Shops zeigt auffälliges Verhalten bei Shop_B. In diesem Fall ist zu beachten, dass die Höher der Balken nur die Anzahl der Produkte wiedergibt und das Verhältnis der einzelnen Balken je Shop relevant ist. Wenn alle Balken die gleiche Höhe haben werden alle Kategorien gleichbehandelt. In diesem Fall liegt keine Unterscheidung nach User Agent (Shop_A) vor. Konträr dazu wurden bei Shop_B User Agents unterschiedlich behandelt. Für das Profil "Iphone" und "Linux" konnten keine Daten gesammelt werden. Es findet daher eine Unterscheidung nach User Agents bei Shop_B statt.

Anmerkung: Dies ist jedoch noch kein Beweis für Preisdiskriminierung. Da es sich bei den Usern "Linux" und "Iphone" um Android bzw. Apple Smartphones handelt, ist diese Unterscheidung vermutlich auf die "mobile Darstellung" zurückzuführen. Jedoch zeugt dies von einer unterschiedlichen Behandlung nach Browseridentität, der durch eine weitere manuelle Betrachtung der erhobenen Datensätze, nachgegangen werden sollte.

Für jede Preisabfrage wird auch der HTML code abgespeichert. In results.csv ist in der Spalte "html_name" ist der Dateipfad des HTMLs zur jeweils zugehörigen Abfrage gespeichert. Ein Vergleich der HTML Codes zeigt, dass Smartphone User Agents tatsächlich einen unterschiedlichen Seitenaufbau, jedoch mit gleichen Preisen aufweisen. In diesem Fall muss ein xpath Selector für die Mobilversion der Seite hinzugefügt werden. (siehe preis.wert Technical Guide – Konfiguration der Preisextraktion)

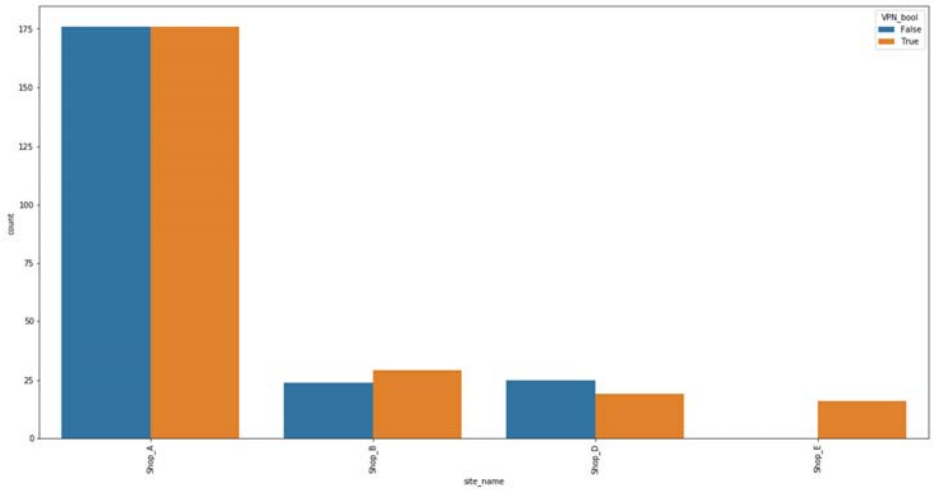


Diskriminierung nach Cookies

Die Aufschlüsselung nach Cookie Einstellungen zeigt keine eindeutige Unterscheidung bei der Anzahl der erfolgreichen Preisextraktionen. (blaue = keine Cookies, orange = mit Cookies)

Diskriminierung nach Location

Die Aufschlüsselung nach VPN und Shops zeigt eine Auffälligkeit bei Shop_E. In der Abbildung werden Verbindungen die VPNs verwenden (orange) und Verbindungen ohne VPNs unterschieden (blau). Bei Shop_E waren ausschließlich Abfragen von VPNs erfolgreich. Dies lässt auf die unterschiedliche Behandlung nach IP Adressen schließen, ist aber wieder noch kein Beweis für Preisdiskriminierung.



Da eine Analyse des Preisverlaufs zwei Messpunkten wenig sinnvoll ist werden die weiten Abschnitte des Evaluations Notebooks anhand eines größeren Datensatzes präsentiert

Beispieldatensatz 2 – größeres Sample

Dieser beinhaltet Preismessungen über mehrere Monate und wird verwendet um die Analyse der Preisverläufe und Preisänderungen zu illustrieren. Über den Untersuchungszeitraum von vier Monaten wurde das Tool hierbei bei laufender Datenerfassung weiterentwickelt. Beispielsweise wurde zunächst

zufällige Proxys und zufällige User Agents verwendet und im weiteren Verlauf durch VPNs und Personengruppen entsprechenden User Agents verwendet/abgelöst. Dadurch wurden die definierten Profile nicht gleichmäßig verwendet und Aussagen über die Erfolgsquote von Abfragen (siehe Beispieldatensatz 1) sind nicht zulässig.

Anzahl der Preisänderungen

Um Beispieldatensatz 2 zu verwenden, ändern Sie „results_path“ in Zelle 2 zu:

```
results_path = "../results/preiswert/example_long.zip"
```

Führen Sie anschließend alle Zellen neu aus („Kernel“ → „Restart & Run All“) und wechseln Sie zum Abschnitt “Anzahl der Preisänderungen”. In diesem werden die Anzahl der Preisänderungen je Shop dargestellt - diese Tabelle wird auch in „output_path“ als CSV abgespeichert. Hier ist auffällig, dass Shop_D verhältnismäßig viele Preisänderungen durchführt. Jedoch sind weniger als 10% der Preisänderungen größer als 10€. Eine mögliche Erklärung dafür könnte sein, dass der Wettbewerber, seine Preise adaptiert um als einer der günstigsten Anbieter in bei einem Marktplatz oder einem Preisvergleichsportal aufzuscheinen. Die Bewerber scheinen sich dabei häufig um wenige Cent zu unterbieten.

Im Untersuchungszeitraum von 2019-05-17 09:32:37.144343 bis 2019-09-23 08:50:26.276173 wurden folgenden Preisänder ungen festgestellt.

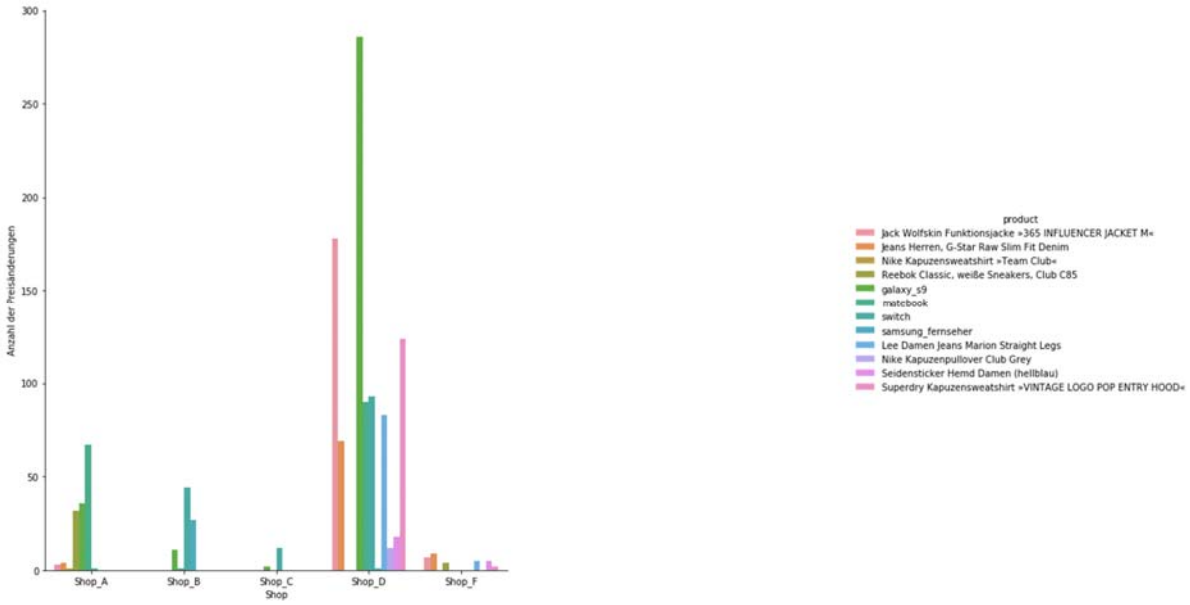
site_name	Preisänderungen größer als 0 €	Preisänderungen größer als 1 €	Preisänderungen größer als 10 €
Shop_A	144	144	110
Shop_B	83	55	35
Shop_C	14	13	6
Shop_D	954	395	91
Shop_F	32	31	11

Des Weiteren werden die Preisänderungen je Produkt angegeben. Dadurch wird ersichtlich, dass sich lediglich einzelne wenige Produkte für den Großteil der Preisänderungen verantwortlich zeigen.

Im Untersuchungszeitraum von 2019-05-17 09:32:37.144343 bis 2019-09-23 08:50:26.276173 wurden folgenden Preisänder ungen festgestellt.

product	Preisänderungen größer als 0 €	Preisänderungen größer als 1 €	Preisänderungen größer als 10 €
Jack Wolfskin Funktionsjacke »365 INFLUENCER JACKET M«	188	104	11.0
Jeans Herren, G-Star Raw Slim Fit Denim	82	16	10.0
Lee Damen Jeans Marion Straight Legs	88	12	1.0
Nike Kapuzenpullover Club Grey	12	5	NaN
Nike Kapuzensweatshirt »Team Club«	1	1	1.0
Reebok Classic, weiße Sneakers, Club C85	36	36	22.0
Seidensticker Hemd Damen (hellblau)	23	20	NaN
Superdry Kapuzensweatshirt »VINTAGE LOGO POP ENTRY HOOD«	126	6	NaN
galaxy_s9	335	239	100.0
matebook	158	86	75.0
samsung_fernseher	28	28	25.0
switch	150	85	8.0

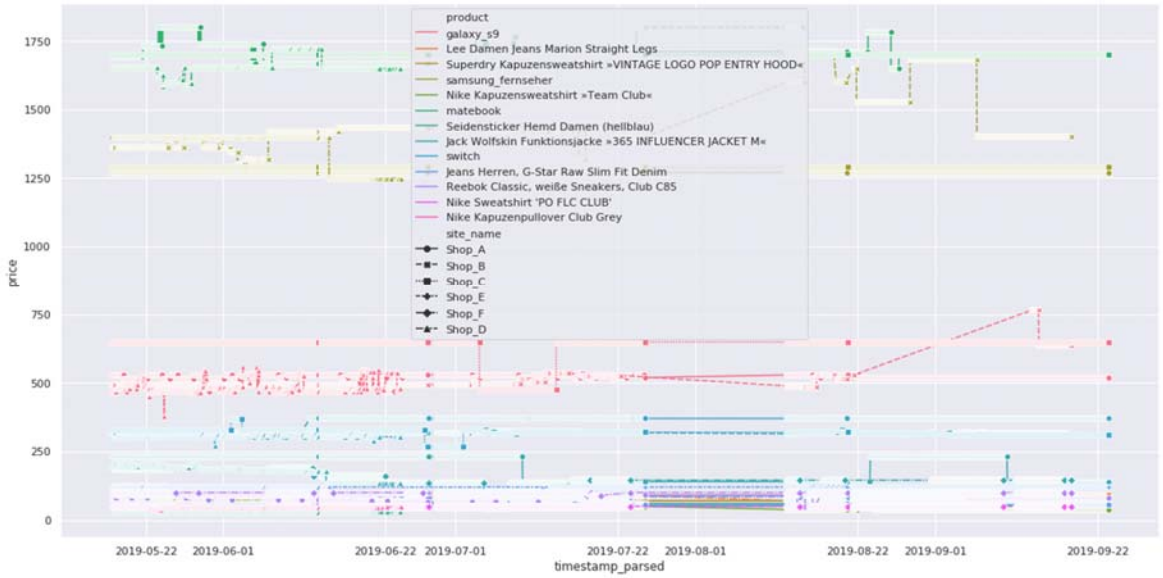
Für jede der drei Kategorien (alle Preisänderungen, Preisänderungen > 1€ und Preisänderungen größer als 10€) wird ein Balkendiagram für alle gemessenen Produkte erstellt und gespeichert. Im Folgenden Beispieldiagramm mit einem Überblick über die Anzahl dieser Preisänderungen je Kategorie.



Preisverlauf

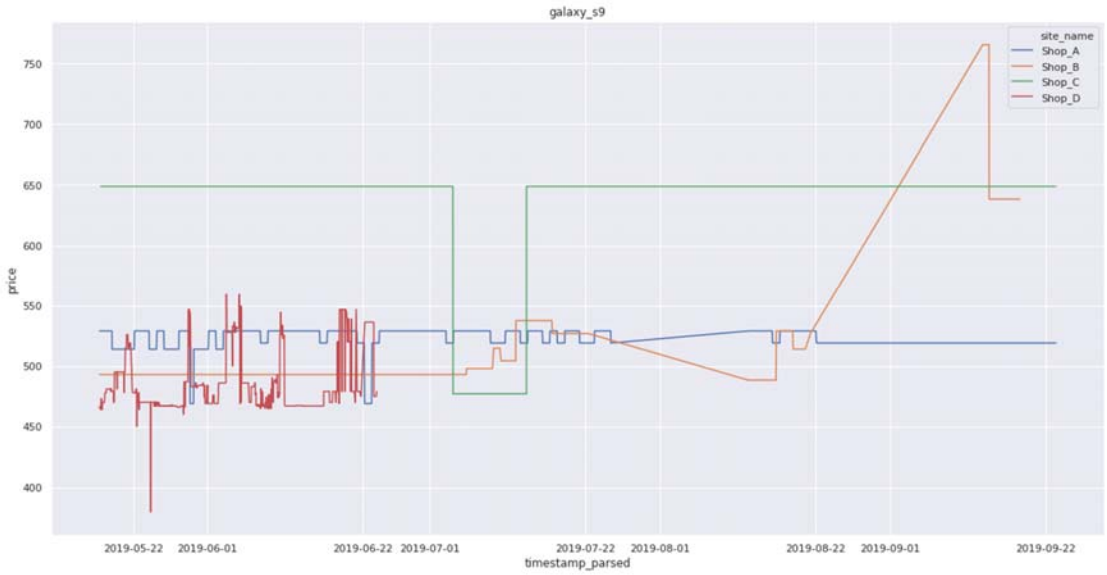
Preisverlauf aller Produkte in einer Abbildung

Um einen Überblick über alle extrahierten Preise je Produkt zu erhalten wird eine zusammenfassende Abbildung erstellt.



Preisverlauf je Produkt

Des Weiteren wird für jedes Produkt eine Abbildung erstellt in der die Preisentwicklung, aufgeschlüsselt nach Shops, dargestellt wird.

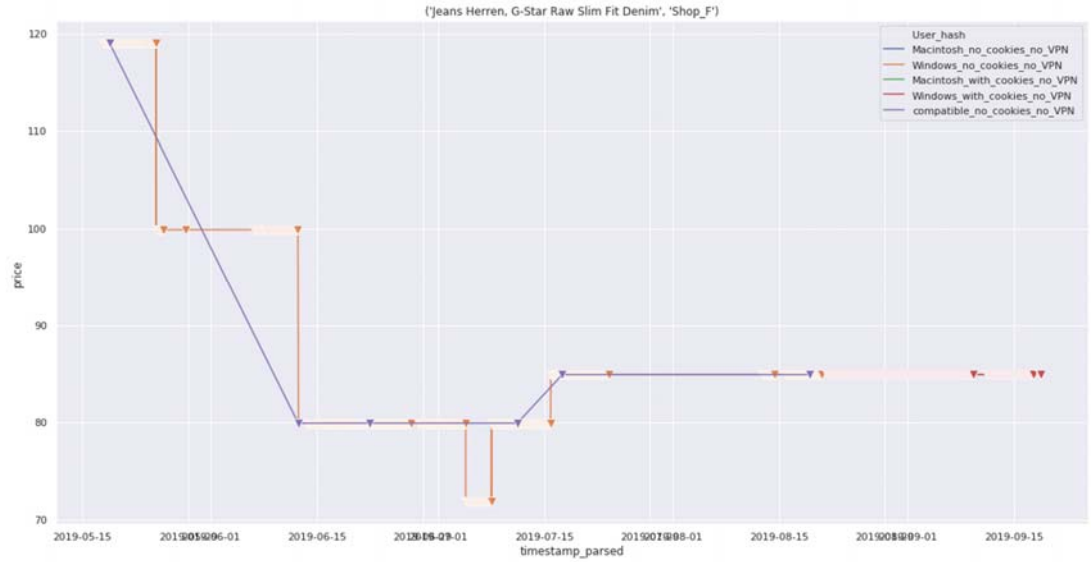


Preisverlauf je Produkt und Shop nach Identität

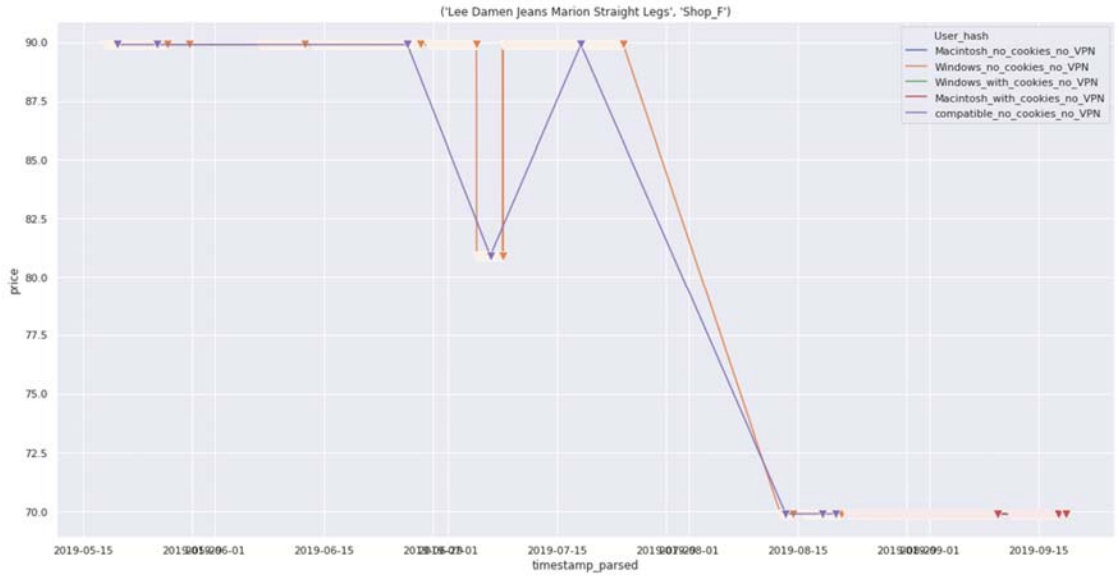
Um potentiell personalisierte Preisgestaltung detektieren zu können wird eine Abbildung für jedes Produkt bei einem Shop einzeln auch dargestellt und nach verwendeten Identitäten aufgeschlüsselt.

Dabei sind die einzelnen Abfragen durch Dreiecke dargestellt. Die Verbindungslinie zwischen den Datenpunkten dient ausschließlich der Illustration und entspricht hier keinem tatsächlichen Preis. Ein mehrfaches Abweichen der Datenpunkte bei verschiedenen Identitäten könnte ein Hinweis auf personalisierte Preisgestaltung sein.

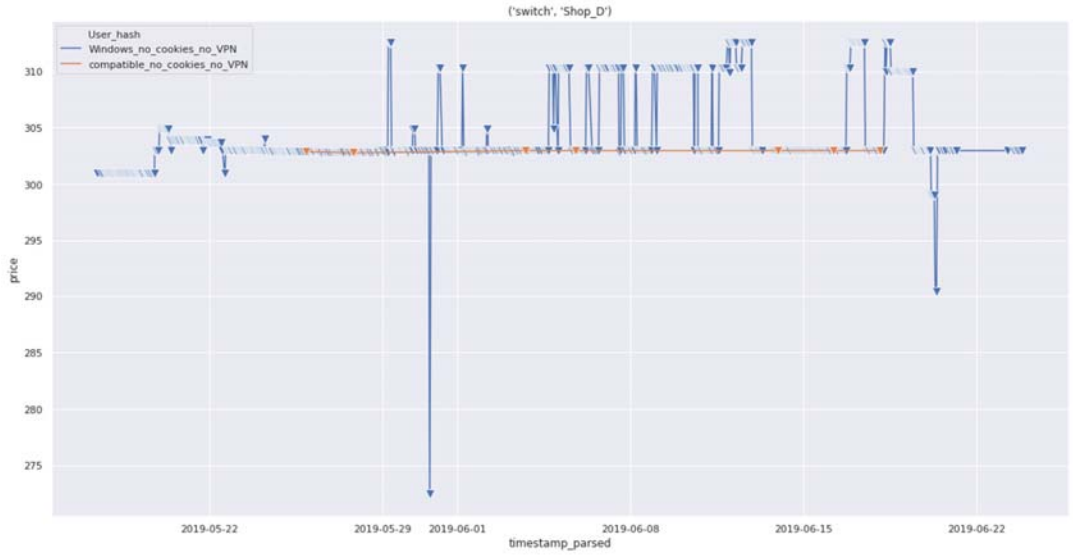
In der folgenden Abbildung sind häufige Preisänderungen zu sehen. Jedoch sind keine Preisunterschiede zwischen den abgefragten Identitäten zu erkennen. Es sei darauf hingewiesen, dass manche Datenpunkte in der Abbildung nicht erkennbar sind, da sie von den Datenpunkten einer anderen Identität überdeckt werden.



Auch in diesem Fall sind Preisänderungen, aber keine Preisdiskriminierungseffekte zu sehen.



In diesem Beispiel weisen einzelne Abfragen erhöhte Preise aus. Ob es sich hierbei um kurzfristige Preisänderungen für alle Kunden oder um personalisiere Preise handelt ist mit einem Datenpunkt nicht festzustellen.



Zoom bei einzelnem Produkt

Um einen Vergrößerten Ausschnitt für einen der obigen Abbildungen zu erstellen wählen die ProduktID aus, welche aus dem Titel der Abbildung zu einem Wort zusammengesetzt wird aus. Z.B.

```
# select a product from cell above
selection = "matebook_Shop_A"
```

In der folgenden Zelle wählen Sie das Datumsintervall für die Abbildung im Format: (Jahr, Monat, Tag, Stunde, Minute). Z.B.

```
# Follow the following format pd.Timestamp(year, month, day,hour, minute)
timestamp_min = pd.Timestamp(2019,7,3,12,20)
timestamp_max = pd.Timestamp(2019,7,5,12,20)
```

In der nächsten Zelle wird der ausgewählte Ausschnitt angezeigt.