**Bonprix Coding Challenge**

Die nachfolgende Dokumentation führt durch meine abgegebenen Programme zur Bonprix Coding Challenge (<https://www.bonprix.de/corporate/karriere/coding-challenge/>). Bei Fragen, Rückmeldungen oder Kommentaren bin ich erreichbar per Mail unter [mail@larissahaas.de](mailto:mail@larissahaas.de) oder unter allen genannten Kontakt-Wegen auf [www.larissahaas.de](http://www.larissahaas.de).

# Inhalt

[Übersicht Dateien 1](#_Toc7010766)

[NLP Klassifizierung 1](#_Toc7010767)

[Web-Anwendung 1](#_Toc7010768)

[Starten 1](#_Toc7010769)

[Manuelle Installation 1](#_Toc7010770)

[Installation mit Docker 1](#_Toc7010771)

[Pythonanywhere Server 1](#_Toc7010772)

[Funktionen 1](#_Toc7010773)

[Erweiterungsmöglichkeiten 1](#_Toc7010774)

# Übersicht Dateien

Im mitgesendeten zip-Order befinden sich die Dateien, die erstellt wurden, um den gelieferten Datensatz in die gewünschte Form zu bringen. Die Dateien bestehen aus drei Jupyter-Notebooks (jeweils ein Notebook für jede Datenbanktabelle der späteren Anwendung). Zusätzlich werden weitere Dateien benötigt, die sich beim Ausführen der Jupyer-Notebooks im gleichen Ordner wie diese befinden müssen: *add-stopwords.txt* (enthält häufige Wörter der Bewertungen, die aussortiert werden sollen)sowie *nltk-german-classifier-data.pickle* (ein selbst erstellter Datensatz, um deutsche Wörter mit POS-Tags zu versehen). Außerdem befinden sich die eben erwähnten drei Output-Tabellen in dem Ordner, die nach der Erstellung in die Datenbank der Anwendung geladen wurden: *topics.csv*, *reviews.csv* und *products.csv.* Der Ordner wurde außerdem als Github-Repository hochgeladen (<http://github.com/LarissaHa/bonprix-nlp.com>), dort sind vor allem die Jupyter-Notebooks einsehbar, auch wenn lokal kein Python/IPython installiert sein sollte.

Neben dem mitgesendeten Ordner gibt es noch ein Github-Repository unter <http://github.com/LarissaHa/bonprix.com>, welches die Dateien für die Web-Anwendung enthält. Dieser Code kann geklont und lokal ausgeführt werden (mehr dazu unter Starten), eine Kopie davon läuft aber auch auf einem Pythonanywhere-Server unter:

<http://bonprixchallenge.pythonanywhere.com>

# NLP Klassifizierung

Die Analyse der Bewertungen wurde mit Python in der Umgebung von Jupyter-Notebooks durchgeführt. Die Voraussetzungen für das lokale Ausführen der Notebooks werden in der mitgelieferten *requirements.txt* aufgelistet. Außerdem sind die Notebooks (also Code sowie Ausgabe/Ergebnisse) im bereits erwähnten Github-Repository (<http://github.com/LarissaHa/bonprix.com>) zu finden.

Die Analyse von Schlagworten ist im Jupyter-Notebook *topics-csv-gen.ipynb* zu sehen.

# Web-Anwendung

Die Web-Anwendung läuft komplett mit den Dateien aus dem oben erwähnten Github-Repository. Die Anwendung läuft mit Django, dem Web-Framework von Python. Das Backend, in dem auch die Daten einzusehen sind, ist erreichbar unter *root/admin/* und kann mit der Nutzer-Passwort-Kombination „bonprix“-„bonprix-challenge“ eingesehen werden.

## Starten

Die Dateien können aus dem Github-Repository geklont und lokal auf zwei Arten gestartet werden: Manuelle Installation oder Installation mit Docker. Außerdem ist die Anwendung aus dem Repository auf Pythonanywhere einsehbar:

<http://bonprixchallenge.pythonanywhere.com>

Die Seite wurde in Windows geschrieben und unter Windows getestet. Die folgenden Anweisungen beziehen sich auf eine Nutzung unter Windows. Ein gutes Tutorial, das einen Umgang mit Django unter allen Betriebssystemen erklärt, bieten Django-Girls:

<https://tutorial.djangogirls.org/de/>

### Manuelle Installation

Um die Anwendung lokal laufen zu lassen, ist Python 3.6 notwendig sowie die in der *requirements.txt* angegebenen, zusätzlichen Pakete: Django (2.0.10) für das Management der Web-Anwendung sowie Pillow, um im Admin-Bereich Bilder für die Produkte hochladen zu können. Wurden diese Voraussetzungen geschaffen (am besten in einer virtuellen Entwicklungsumgebung, um eventuelle Überschneidungen zu anderen Python-Projekten zu verhindern), kann das Projekt lokal gestartet werden.

Dafür reicht der Konsolenbefehl *python manage.py runserver --insecure* in dem Order, in dem sich die manage.py befindet, also der obersten Ebene in der Dateistruktur. Der Zusatz *insecure* ist wichtig, da sich die Seite nicht mehr im Entwicklungsmodus befindet und somit aus Sicherheitsgründen die statischen Dateien nicht mehr geladen werden. Die Seite ist nun unter *localhost:8000/* zu finden.

### Installation mit Docker

Im Github-Repository befindet sich ebenfalls ein *Dockerfile* mit allen notwendigen Informationen, die notwendig sind, um mit Docker die benötigte Umgebung für die Django-Anwendung zu schaffen. Um Docker zu verwenden sollten Docker sowie Python 3.6 installiert sein. Das weitere Vorgehen wäre wie folgt: mit einer Konsole den Projektorder aufrufen, den Befehl *docker build -t bonprix:1 .* eingeben (den Punkt am Ende nicht vergessen), anschließend *docker run bonbrix:1 -p 8000:8000* eingeben. Nun sollte der Container laufen und die Seite sollte über *localhost:8000/* verfügbar sein.

### Pythonanywhere Server

Neben der lokalen Installation ist die Seite auch auf einem Pythonanywhere-Server einsehbar:

<http://bonprixchallenge.pythonanywhere.com>

Die Version auf dem Server ist eine aktuelle Kopie des Github-Repositorys und wurde ansonsten in keiner Weise verändert.

## Funktionen

## Erweiterungsmöglichkeiten

Leider war es mir nicht möglich, die Topic-Klassifizierung dynamisch in die Anwendung einzubauen (also dass die Klassifizierung automatisch durchgeführt wird, wenn ein neuer Kommentar geschrieben wurde). Allerdings habe ich alle Dateien eingefügt, die ich als notwendig erachte, um diese Funktion zu aktivieren (wenn Sie eine Idee haben sollten, warum meine Funktion nicht funktioniert, bin ich über Rückmeldungen sehr dankbar!). Die Kern-Funktionen stehen in der Datei *reviews/nlp.py* und werden von *views.py* aufgerufen, wenn dort die auskommentierten Code-Zeilen aktiv eingefügt werden. Zusätzlich werden dort die gelieferten Topics in die Datenbank geschrieben und sollten dann auf der Seite einsehbar sein.

Weitere Erweitungsmöglichkeiten wären weitere **Sortierfunktionen**, zum Beispiel nach Länge der Review oder nach Farbe/Größe/Ausführung des bestellten Artikels (wenn diese Informationen vorliegen). Außerdem könnten zusätzlich zu den Topics noch die **Sentiments** zu den Topics analysiert werden, also nicht nur „Wird über die Passform geschrieben?“, sondern auch „Wurde die Passform positiv oder negativ bewertet?“. Einen Ansatz für diese Funktion wurde mit dem „gefilterten“ Sternendurchschnitt bereits eingeführt.

# Über mich

Nach der technischen Erklärung ein paar kurze Worte über mich: Ich studiere im Moment im 5. Semester im Mannheim Master in Data Science und schreibe gerade an meiner Master-Thesis. Vor dem Master habe ich meinen Bachelor in Politikwissenschaften (mit Beifach Medien- und Kommunikationswissenschaften) gemacht und habe mich dann durch den Spaß an der Statistik immer mehr für Informatik interessiert. In meiner Freizeit spiele ich Hockey, schreibe Geschichten und bastle Webseiten für Freunde und Familie.

## Nachweis Immatrikulationsbescheinigung

