Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

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 oder unter allen genannten Kontakt-Wegen auf www.larissahaas.de.

Inhalt

Übersicht Aufgabe

Die gestellte Aufgabe forderte, die zur Verfügung gestellten Produktbewertungen mithilfe von Natural Language Processing zu analysieren und nach Schlagworten zu sortieren. Diesen Schritt habe ich (statisch) mit **Python** gelöst. Die Kommentare werden einmal eingelesen, verarbeitet und als Schlagwort-Tabelle wieder ausgegeben. Der zweite Schritt der Aufgabe, die (klickbare) Darstellung der Schlagworte und die automatische Filterung der Kommentare nach dem gewählten Schlagwort, wurde mit **Django**, dem Python-Web-Framework, realisiert.

Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

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-nlp.com) zu finden.

Die Analyse von Schlagworten ist im Jupyter-Notebook *topics-csv-gen.ipynb* zu sehen. Nachdem verschiedene Pakete und Klassen importiert und geladen wurden, werden die Daten eingelesen. Generell wird für die Analyse das Python Paket *Pandas* verwendet, denn es bietet einen angenehmen und intuitiven Umgang mit Daten in Tabellenform. Jeder Analyseschritt wird von der Funktion *tqdm* begleitet, die den Fortschritt der Bearbeitung in einem Balken anzeigt (und auch berechnet, wie lange der Schritt ungefähr noch dauern wird). Nach jedem Schritt gibt es außerdem die Möglichkeit, den Stand der Analyse als *json*-Datei zwischen zu speichern.

Dann beginnt das Natural Language Processing: Der Reihe nach werden die Kommentare in Sätze unterteilt, die Sätze in Wörter und Punkte und Kommata werden entfernt. Die darauffolgenden Zellen sind alle auskommentiert, denn sie beinhalten das Training eines deutschen POS-Taggers, also eines Algorithmus, der

Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

Wörtern in Sätzen automatisch ihre Wortarten annotiert. Da dies nur einmal gemacht werden muss (und die Ergebnisse in der Datei *nltk-german-classifier-data.pickle* zwischengespeichert wurden), bleiben diese Zellen inaktiv. Der Algorithmus muss nur geladen und auf den Datensatz angewandt werden. Nun können mithilfe des Pakets *germalemma* die Lemmata generiert werden. Ein **Lemma** ist nach Wikipedia „die Grundform eines Wortes, also diejenige Wortform, unter der man einen Begriff in einem Nachschlagewerk findet“.[1] Dies stellt sicher, dass die Schlagworte später nur „sinnvolle“ Begriffe beinhalten. Letztendlich werden die Lemmata noch normalisiert, also alle in die kleingeschriebene Form umgewandelt. Hat man sich nun angesehen, was die Lemmata vor allem beinhalteten, so war es eine Mischung aus sinnvollen Schlagworten, unnötigen Füllwörtern und produktbezogenen Wörtern wie „Kleid“ oder „Schuh“. Die beiden letzten Fälle wurden nun (mithilfe der zusätzlichen Datei *add-stopwords.txt*) im Zuge des Stopword Removal entfernt. Schließlich wurden die Schlagworte noch in das richtige Format gebracht, also pro Schlagwort eine Tabellenzeile, verknüpft durch die Produktnummer sowie eine Review-ID.

Zusätzlich wurden noch zwei Jupyter-Notebooks angelegt, die dabei helfen sollen, die Web-Anwendung mit den richtigen Daten zu versorgen. *Reviews-csv-gen.ipynb* bringt die Tabellenspalten in die richtige Reihenfolge und fügt ein Autor- sowie ein Datums-Feld ein, das später vom Web-Framework benötigt wird, um die Bewertungen zu sortieren.

In *products-csv-gen.ipynb* wird eine eigene Produkt-Tabelle angelegt. Diese Tabelle wird einerseits mit durchschnittlichen Stern-Bewertungen gefüllt, dies ist eine Vorbereitung darauf, wenn die Web-Anwendung später mit einer hohen Anzahl von Bewertungen umgehen können muss und die Durchschnittswerte nicht jedes Mal aus den aufgerufenen Bewertungen errechnet werden sollen. Im Moment werden diese Werte allerdings weder benutzt noch bei neuen Bewertungen gepflegt. Andererseits werden in diesem Notebook Informationen hinzugefügt, die ich anhand der Produktnummern von der bonprix-Webseite einsehen konnte, so zum Beispiel der Name des Produkts oder ein Foto davon. Das Foto (bzw. der Pfad zur Bilddatei) kann auch später über das Django-Admin-Backend hochgeladen und verändert werden.

Web-Anwendung

Die Web-Anwendung läuft komplett mit den Dateien aus dem oben erwähnten Github-Repository (http://github.com/LarissaHa/bonprix.com). 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.

[1] Wikipedia: Die freie Enzyklopädie (http://de.wikipedia.org/wiki/Lemma_(Lexikographie))

Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

Starten

Die Dateien können aus dem Github-Repository geklont und lokal auf zwei Arten gestartet werden: über die Manuelle Installation oder die 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.

Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

Funktionen

Design

Der Entwurf für das HTML-Design stammt nicht von mir, sondern wurde von einer Seite entnommen, die HTML-Templates unter CC BY Lizenz öffentlich zur Verfügung stellt. Die Quelle des Templates kann hier eingesehen werden:

https://colorlib.com/wp/template/elen/

Alle Produkte

Unter „alle Produkte" werden die Produkte der Reihe nach mit Foto angezeigt. Klickt man auf ein Produkt, gelangt man zur Produkt-Detailseite. Die Überschrift enthält sowohl den Namen des Produkts als auch die Produktnummer. Darunter wird neben dem Foto zunächst die durchschnittliche Sternen-Bewertung angezeigt (die ausgefüllten und leeren Sterne orientieren sich am Durchschnitt, bei >= .5 kommt ein ausgefüllter Stern dazu). Darunter werden die einzelnen Sterne aufgeschlüsselt und mit Prozent versehen dargestellt. **Stern-Filterung**: Wenn man auf einen Balken oder den Stern davor klickt, werden die dargestellten Bewertungen darunter nach dem geklickten Stern gefiltert (d.h. man sieht zum Beispiel nur noch 3-Sterne-Bewertungen). Der Button „Filterung nach Sternen aufheben" führt einen zurück zur allgemeinen Produkt-Detailseite.

Unter der Stern-Übersicht befindet sich der Button „bewerte das Produkt". Mehr dazu unter Bewertung abgeben. **Themen-Filterung**: Unter „Häufig erwähnte Themen" befindet sich eine Auflistung der zehn häufigsten Themen, die nach den NLP-Ergebnissen in den Kommentaren angesprochen wurden. Die Nummer in der Klammer hinter dem Thema gibt dabei die absolute Anzahl an, wie oft dieses Thema genannt wurde. Klickt man auf ein Schlagwort, werden die angezeigten Kommentare nach diesem Thema gefiltert. Dabei ist die Anzahl bei dem Thema angezeigter Kommentare ist meist ein wenig kleiner als die oben angegebene Nummer, da manche Kommentare ein Themen-Schlagwort mehrmals enthalten. Die Kommentare sind nach Datum sortiert. Unter der Überschrift „Bewertungen über…" ist eine zusätzliche Durchschnittsbewertung von Sternen zu sehen, dabei werden nur Kommentare des angeklickten Themas zur Berechnung des Durchschnitts verwendet. Ein Klick auf den Button „Filterung nach Thema aufheben" führt einen zurück zur allgemeinen Produkt-Detailseite.

Unter den Schlagworten finden sich die Kommentare, die standardmäßig nach Datum sortiert sind. Dies ist gekennzeichnet durch die dunkle Hervorhebung des Buttons „neueste zuerst". **Sortierungsoptionen**: Außerdem können die Kommentare mit Klick auf den jeweiligen Button nach Sternen sortiert werden, also von hoch zu niedrig, oder von niedrig zu hoch.

Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

Blätter-Funktion: Bei sämtlichen Ansichten und Filter- bzw. Sortierungsoptionen befindet sich am Ende der Seite eine Blätter-Funktion, mit der man sich durch alle Kommentare klicken kann, jeweils in 10er Schritten.

Aktuelle Bewertungen

Unter „aktuelle Bewertungen" werden die 10 letzten Kommentare (produktübergreifend) angezeigt. Ein Klick auf das Produktfoto oder den Produktnamen führt zur Produkt-Detailseite. Unten auf der Seite steht ebenfalls die Blätter-Funktion zur Verfügung.

Bewertung abgeben

Sowohl über den Navigationsverweis „neue Bewertung abgeben" oder über den Button „bewerte das Produkt" kann man neue Kommentare eingeben und speichern. Der einzige Unterschied besteht darin, dass bei der letztgenannten Option die Produktnummer automatisch vorausgewählt wurde und man somit einen Kommentar zu ebendiesem Produkt abgibt, das man sich gerade angeschaut hat. Bei dem allgemeineren „neue Bewertung abgeben" kann man das Produkt aus einer Dropdown-Liste auswählen.

Zusätzlich kann man den Kommentar-Text eingeben und die Anzahl an Sternen auswählen, die man vergeben möchte. Zusätzlich wären noch weitere Felder wie *Name* oder Produkteigenschaften wie *Größe*, *Farbe* oder Ähnliches denkbar und möglich. Mit „absenden" speichert man den Kommentar in der Datenbank. Automatisch wird das aktuelle Datum als Erstellungsdatum gesetzt, somit erscheint der Kommentar nun in den Kommentarauflistungen der Seite, natürlich für das ausgewählte Produkt. Mehr über das Abgeben und Speichern von Kommentaren im nächsten Abschnitt Erweiterungsmöglichkeiten.

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 **Sortierungsoptionen**, 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

Larissa Haas
Raiffeisenstraße 35
67489 Kirrweiler
mail@larissahaas.de

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


Universität Mannheim
Studienbescheinigung

Frau
Haas, Larissa Laura
geb. am 26.04.1994
ist im FSS 19
an der Universität Mannheim eingeschrieben

Hochschulsem 12
Urlaubssem: 0

1. Abschluss: Master Fachsem
MA Master in Data Science 5