Key

Repo Struktur

DB normalisieren

-replace venv with pipenv (pipenv why better)

-use pytest

durch test im file-name erkennt pytest diese als test files:

.py for functions instead of ipynbs

put .py files in ordner "lib_py"

static definiere. Beispiel

import os
INPUT_DIR = "../input"

FAZIT: mit dict oder listen so weit wie möglich kommen und pandas nur bei bedarf.

`into_dwh` file (data warehous) den output in z.b. sqlite

Readme:

1. Einzeiler, was es ist.
2. ein *quickstart* hinzufügen, z.b. mit install und run, pip install -r requirements.txt t.b. eventuell auch n *setup.script* nutzen zum generieren von input/output ordner und anderen lokalen dateien.
3. Key Features
4. erweiterte code beispiele

idee ziel db

prefix, suffix, pre-context, post-context, date_published, newspaper

analyse: countd by data und newspaper, gibt pro tag ti anzahl von klima wieder.

Arbeit/report:

Einleitung darf ¼ der arbeit, sein, also alles erklären was ist das thema, ki vs blockchain, warum sind diese begriffe interessant, was gibt es dazu, woher kommt das, warum besteht die annahme. vorüberlegungen, auch mit quellen (3-10, 5) untermauern. auch nicht-wissenschaftliche daten dürfen als beweise herhalten.

Datenmethoden: woher kommen die daten, welche analyse methoden, wie wurde sql db aufgebaut, WIE ist vorgegangen.

Ergebnisse: es gibt keine objektivität, aber so objektiv wie möglich sein. "ich sehen dass, trump 3x hüfiger genannt wird bei ntv." darstellungen, wenige aber informative. weitere können in anhang

Erst dann diskussion: hier kann interpretiert werden, warum etwas so ist. eventuell auch einleitung bestätigen oder widerrufen.

anhang: ausgeführtes jupyternotebook mit analysen, bzw. das repo als zip ohne data

zu den grafiken:

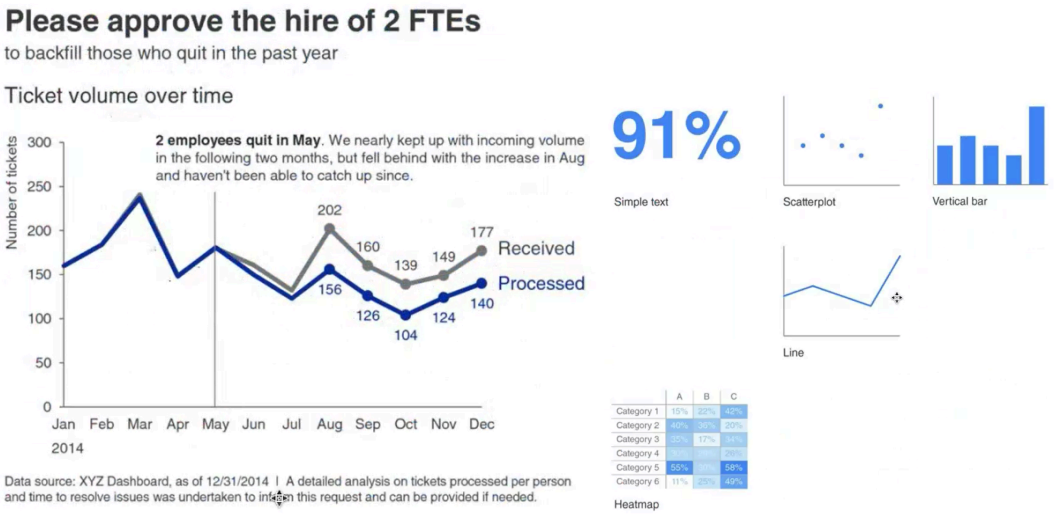
1-2 grafiken, die wesentlich sind im fließtext

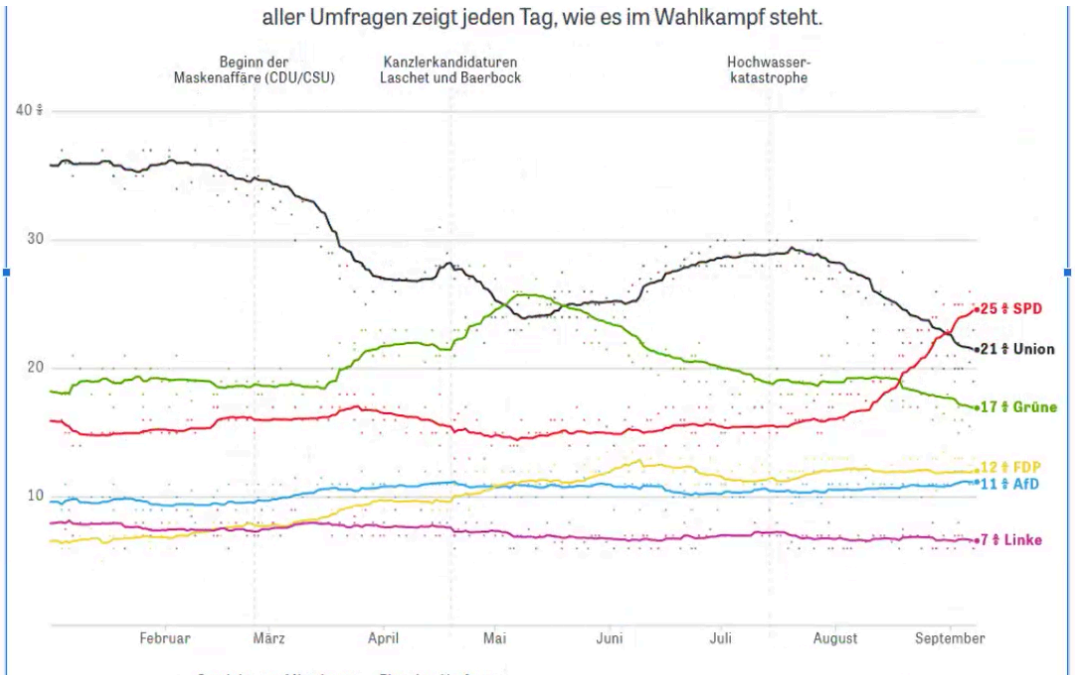
schwarz weiß

only highlight colored

no lines or clutter

in first picture below: it's a story, since we have a breakpoint, and explanation for that one and we are focused (can't catch up)





archiv

quell dateien, also gescrapte html newspaper:

https://dbu-my.sharepoint.com/personal/marcel_hebing_dbuas_de/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fmarcel%5Fhebing%5Fdbuas%5Fde%2FDocuments%2Fnews%2Dscraper%2Dzips&ga=1

TEIL 1 Freitag vorm

anwesend (man verzeih mir falsche schreibweise): eva, Konstantin, toni, shawn, janik, sarah, maria, nina, niklas, talap? yannick … niklas alap?

, edwin

warmup:

wo informiert ihr euch tatsächlich über data science themen: youtube, medium, …

diskussion / takeaways:

-Def. **DB normalisieren**: keinerlei redundante Informationen in der Datenbank ablegen.

-developer hour vs computer hour: überlegung anstellen “was ist billiger, vorher oder nachher über fehler nachdenken und testen?”

-replace venv with pipenv (pipenv why better)

-use pytest

durch **test im file-name erkennt pytest diese als test files**:

datei heißt main.py, entsprechender Test heißt dann: main_test.py

function heißt:

test_function_name():

assert function_name(x) = ZIEL

-use **.py for functions instead of ipynbs**

with `main: if__ name__ == __ main__`: (für schnelles ausführen, aber in py also lib nicht mehr nötig)

`function x()...`

put .py files in ordner “pylib”

-than use functions in jupyter notebook with **import e.g.**

Buchempfehlung: Don't make me think (Steve Krug)

TEIL 2 freitag nachm

beatifulsoup. html scrapen

tipp: in notebook: wenn iwo confs oder Dateinamen sind, in zweiter Zeile nach Imports als **static definiere. Beispiel**

import os
INPUT_DIR = "../input"

-ab wann im code sollte ich lists+dicct ODER dataframes nutzen?

> df für reine analyse nutzen, also wenn daten fertig für analyse sind. so lange wie möglich darauf verzichten. also zu anfang python lists oder dicct nutzen etc. ABER man kann auch mal früher schon pandas.read_csv() nutzen, und dann wieder mit dicct arbeiten.**FAZIT: mit dict oder listen so weit wie möglich kommen**.

-use standard dictionary (https://dsm.impactdistillery.com/ads-01/02-git/15-standard-directory-layout.html)

-notebook als explorative datei nutzen, später on .py exportieren. alles was langfristig ist, in .py dateien, hier ist auch der rekursive aufruf anderer dateien (module) und function leichter/convenience. *from dateiname import funktionsname*

-**.pickles datei in temp ordner** schreiben

TEIL 3 samstag

heutige aufgabe

wie die komplexität vom einfachen beispiel wordcount erhöhen:

1. mit externen daten kombinieren, oder anreichen
    a. oder anreichern mit externen events/ereignissen, neue epochen, also ukraine, chatgpt. auch das kann als art von externen daten gesehen werden
2. auch die tags genauer beim scraping diversifizieren, z.b. nur überschriften beachten

dicct anlegen für jeweilige Zeitungen verschiedene Parameter anlegen, eigentlich ne art konfigurationsdatei. also zeitung a > h2 scrapen, bei zeitung b > h3 scrapen.

Diskussionen

mit *#! /usr/bin/python python* (shebang) kann man editor/bash sagen, mit was das script ausgeführt werden soll

auch kann man das format angeben, als hilfreicher hint

wir können bei dem `into_dwh` file (data warehous) den output in z.b. pickle oder csv oder **sqlite** (nutzt prof marcel) schreiben.

datum können wir je nach usecase alle einmalig in dhw schreiben, cooler ist natürlich, wenn man datum / daten übergibt, die dann hinzugefügt werden.

Readme:

1. **Einzeiler, was es ist.**
2. **ein *quickstart* hinzufügen, z.b. mit install und run, pip install -r requirements.txt t.b.**
   eventuell auch n *setup.script* nutzen zum generieren von input/output ordner und anderen lokalen dateien.
3. **Key Features**
4. **erweiterte code beispiele**

idee ziel db

prefix, suffix, pre-context, post-context, date_published, newspaper

analyse: countd by data und newspaper, gibt pro tag ti anzahl von klima wieder.

Arbeit/report:

Einleitung darf ¼ der arbeit, sein, also alles erklären was ist das thema, ki vs blockchain, warum sind diese begriffe interessant, was gibt es dazu, woher kommt das, warum besteht die annahme. vorüberlegungen, auch mit quellen (3-10, 5) untermauern. auch nicht-wissenschaftliche daten dürfen als beweise herhalten.

Datenmethoden: woher kommen die daten, welche analyse methoden, wie wurde sql db aufgebaut, WIE ist vorgegangen.

Ergebnisse: es gibt keine objektivität, aber so objektiv wie möglich sein. “ich sehen dass, trump 3x hüfiger genannt wird bei ntv.” darstellungen, wenige aber informative. weitere können in anhang

Erst dann diskussion: hier kann interpretiert werden, warum etwas so ist. eventuell auch einleitung bestätigen oder widerrufen.

anhang: ausgeführtes jupyternotebook mit analysen, bzw. das repo als zip ohne data

zu den grafiken:

1-2 grafiken, die wesentlich sind im fließtext

schwarz weiß

only highlight colored

no lines or clutter

in first picture below: it's a story, since we have a breakpoint, and explanation for that one and we are focused (can't catch up)