Automated Data Visualization to help Doctors and Administrators of an Early Lung Cancer Detection Program

Propulsion Academy

T. Klingler

July 21, 2017

Structure

Data Science in the Practice

I-ELCAP Project

Building a Data Product

Conclusion

Data Science in the Practice

- Bundesgesetzes über das elektronische Patientendossier (EPDG)
- Current database systems in many practices are designed for data collection rather than analytics
- Short-term solution: Building data products to bridge the gap

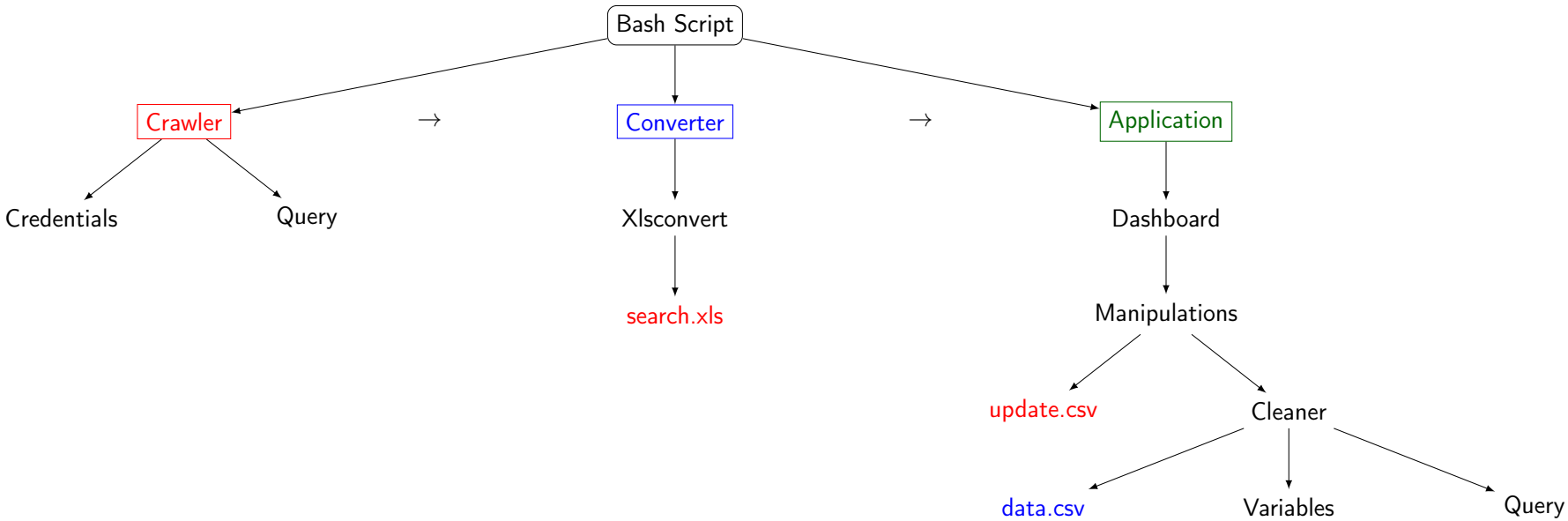
The I-ELCAP Project

- International program for early lung cancer detection
- Most lung cancers are only detected at a late stage: 5 year survival rate in Switzerland at 15%
- National program in Switzerland: privately funded by Stiftung für Lungendiagnostik and conducted in cooperation with the LungenZentrum Hirslanden, Zürich
- Admissions to the program only for high-risk patients: 50 years and older with 20 or more pack-years. Younger patients are admitted in case they have had lung cancer previously

Project Data

- Data sources: Patient intake forms and CT Evaluation forms which are filled out by radiologists with 380 features
- Problems:
  - The data is being stored in the United States inside an old database (ca. 1999)
  - Spreadsheets can be produced through the front end, but querying is complex and the data is not clean
  - Examining the aggregate data takes too much time for doctors and the administrative staff
- Solution: Building an automated visualization Dashboard with input from domain experts

Building a Data Product

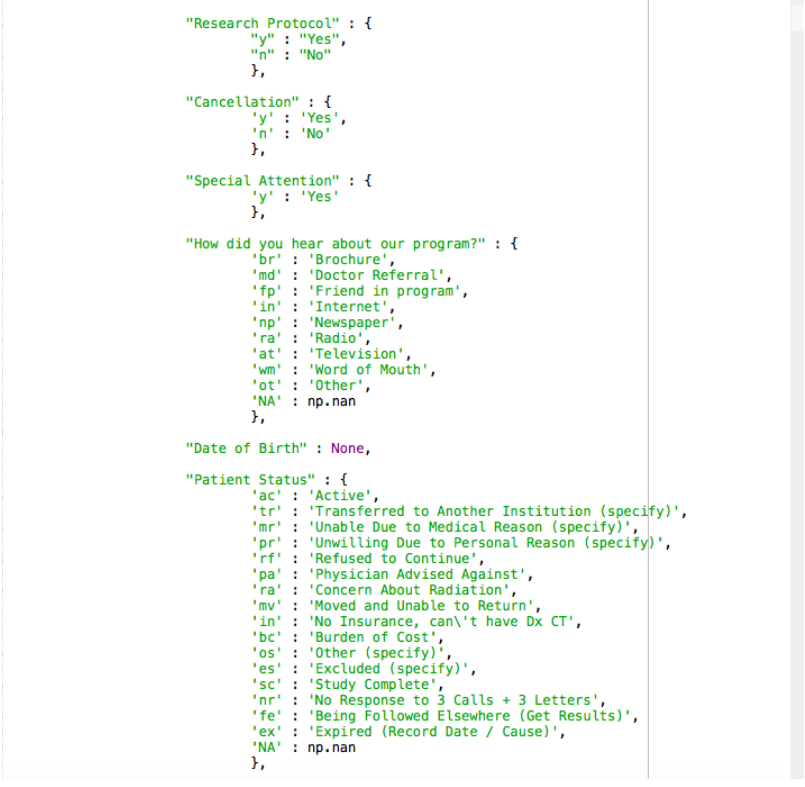


Crawler and Converter

- Webpages are rendered in CGI through javascript click events: No access to backend
- Access to data only through complex querying and manually downloading a messy and broken spreadsheet
- Conversion of the broken .xls file to a .csv was required to be able to work with it in Python

Cleaning and Manipulation

ce.cenncn	ce.cennod	ce.cect1ch	ce.cect2ch	ce.cect3ch	ce.cect4ch	ce.cect5ch	ce.cect0ch	ce.cect1en	ce.cect2en
7	7	n	n	n	n	n	n	no	rml
7	7	n	-	-	-	-	-	no	-
5	7	pw	pn	pw	pw	-	-	-	no
5	7	n	n	n	n	-	-	no	no
3	3	n	n	n	-	-	-	no	no
15	22	n	-	-	-	-	-	no	-
15	22	pn	-	-	-	-	-	no	-
0	0	-	-	-	-	-	-	-	-
0	1	n	-	-	-	-	-	no	-
6	7	-	-	-	-	-	-	-	-
4	4	n	n	n	n	-	-	no	no
1	1	pn	pw	-	-	-	-	no	-
2	2	n	n	-	-	-	-	no	no
10	20	-	-	-	-	-	-	-	-
40	50	n	-	-	-	-	-	no	-
4	6	n	n	n	n	n	n	no	no
7	9	-	-	-	-	-	-	-	-
7	9	-	-	-	-	-	-	-	-
1	1	n	-	-	-	-	-	no	-
18	22	n	n	n	n	n	n	no	no
2	3	n	n	n	-	-	-	no	rll
4	8	pn	rn	-	-	-	-	no	no
4	8	n	-	-	-	-	-	no	-
60	60	n	n	-	-	-	-	no	no
60	60	pn	pn	-	-	-	-	no	no
9	10	pn	-	-	-	-	-	no	-
9	10	n	-	-	-	-	-	no	-
5	6	pn	pd	pn	pn	pw	pn	no	no
5	5	n	n	n	n	n	-	no	no
1	2	n	n	-	-	-	-	no	no
0	4	n	n	n	n	-	-	no	no
6	8	pn	pn	pn	pn	pn	pn	no	no
6	7	n	n	n	n	n	n	no	no
2	2	n	n	-	-	-	-	no	no
2	2	pn	pn	-	-	-	-	no	no
0	0	-	-	-	-	-	-	-	-



```
"Research Protocol" : {
        "y" : "Yes",
        "n" : "No"
        },

"Cancellation" : {
        'y' : 'Yes',
        'n' : 'No'
        },

"Special Attention" : {
        'y' : 'Yes'
        },

"How did you hear about our program?" : {
        'br' : 'Brochure',
        'md' : 'Doctor Referral',
        'fp' : 'Friend in program',
        'in' : 'Internet',
        'np' : 'Newspaper',
        'ra' : 'Radio',
        'at' : 'Television',
        'wm' : 'Word of Mouth',
        'ot' : 'Other',
        'NA' : np.nan
        },

"Date of Birth" : None,

"Patient Status" : {
        'ac' : 'Active',
        'tr' : 'Transferred to Another Institution (specify)',
        'mr' : 'Unable Due to Medical Reason (specify)',
        'pr' : 'Unwilling Due to Personal Reason (specify)',
        'rf' : 'Refused to Continue',
        'pa' : 'Physician Advised Against',
        'ra' : 'Concern About Radiation',
        'mv' : 'Moved and Unable to Return',
        'in' : 'No Insurance, can\'t have Dx CT',
        'bc' : 'Burden of Cost',
        'os' : 'Other (specify)',
        'es' : 'Excluded (specify)',
        'sc' : 'Study Complete',
        'nr' : 'No Response to 3 Calls + 3 Letters',
        'fe' : 'Being Followed Elsewhere (Get Results)',
        'ex' : 'Expired (Record Date / Cause)',
        'NA' : np.nan
        },
```

- I spent less than 80% of my time performing these tasks

Dashboard

- The dashboard was built with plot.ly's Dash framework:
  - It is based on ReactJS for the frontend, Python Flask for the backend and plot.ly's beautiful and interactive plotting library for Python.
  - Allows for quick prototyping with minimal Full-Stack development knowledge and without leaving the Python ecosystem.
  - Drawbacks: HTML has to be coded using Dash's html components in Python and CSS cannot be hosted locally (yet).
- Live Demo

Conclusion: Planned Features

- Replacing Google Chrome with PhantomJS for headless crawling
- Improving some of the plot labels and designs
- Automated e-mails to the administrative staff when new CT evaluations have been uploaded to the database.
- Next stage: using pipeline for image classification of CT scans.