Analyze Prometheus Metrics Like a Data Scientist

Georg Öttl
Promcon 2017, Munich

About me / experiences

- Enterprise Software Dev.
- Data Science Services
- Dev / DevOps / Ops
- Developer who likes Math

Twitter: @goettl

Objective talk

Pushing the limits of prometheus: can I have a more reliable alerts model with insights from datasience?

- Journey on how to improve alerts / dashboards with insights from datasience
- Integration points to open source datasience tools
- Bring light into the dark (like prometheus did)

... should I?

Don't use deep learning and datasience when a straight-forward 15 minute rule-based system does well.

Datascience can help you to detect patterns and facts in your metrics you can't see.

What is already available. When do I start?

- Great architecture to get high quality data
- Numerical data
  - Apply mathematical functions on it
- Easy and fast navigable (promql)
- Alert / rule model
- Chart / histogram vis with Grafana

Next step: get data out of prometheus

… to be used in Open Source datascience tools

What data to export?

- Raw metrics data, no functions applied on it
- As much as possible
    - Without putting too much load on prometheus / running into a timeout

Two ways to get data out of prometheus

- **HTTP API (Poll)**
    - Exploratory data analysis
- REMOTE API (Push)
    - Streaming analysis

HTTP API - /api/v1/query_range

```
requests.get(
  url = 'http://127.0.0.1:9090/api/v1/query_range',
  params = {
    'query': 'sum({__name__=~".+"})  by (__name__,instance)',
    'start': '1502809554',
    'end'  : '1502839554',
    'step' : '1m'
  })

{"data": {...,  "resultType": "matrix",
"result": [{
  "metric": {"method": "GET",...},
  "values": [[1500008340,"3"], ... ]},...]
}}
```

Target format for datascience tools (tabular, csv)

x

id	time	value	req_dur	...
A	1	1	4	...
A	2	2	5	...
B	1	2	3	...
B	2	3	2	...

y

id	time	value
A	1	1
A	2	1
B	1	0
B	2	0
...	...	...

Easyiest way to export

- Grafana
- Python (robustperception blog entry)

Reduce data: use domain knowledge to select relevant data subset

```
{__name__=~".+"}
```

Tip: Use alerts as initial set of training labels

```
y = ALERTS{name="high_latency"}
```

tidy up, verify true positives, annotate manually, ...

Normalize prometheus datatypes

- Gauges, histograms are ok
- Counters have to be processed
    - No repetition in counters. No statistical value in that.
    - Use e.g derivative function to convert a counter to a gauge equivalent

Examples

Applied datasience on prometheus metrics

Example 1

I can predict the latency of http requests

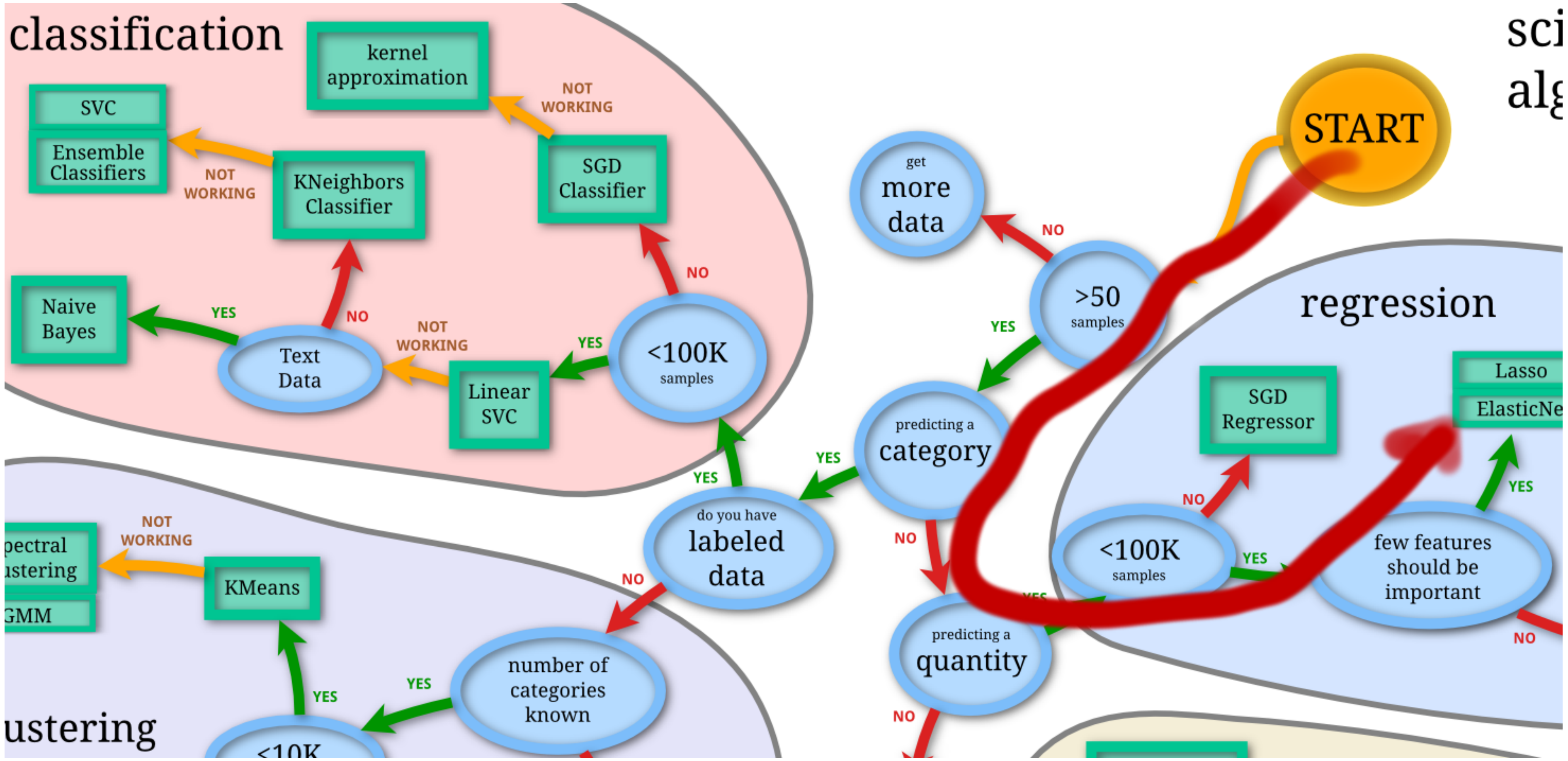
- Can I use the prometheus function predict_linear?
- Are there other predictions possible?

↡↡ R Notebook predict_linear↡↡

Example 2

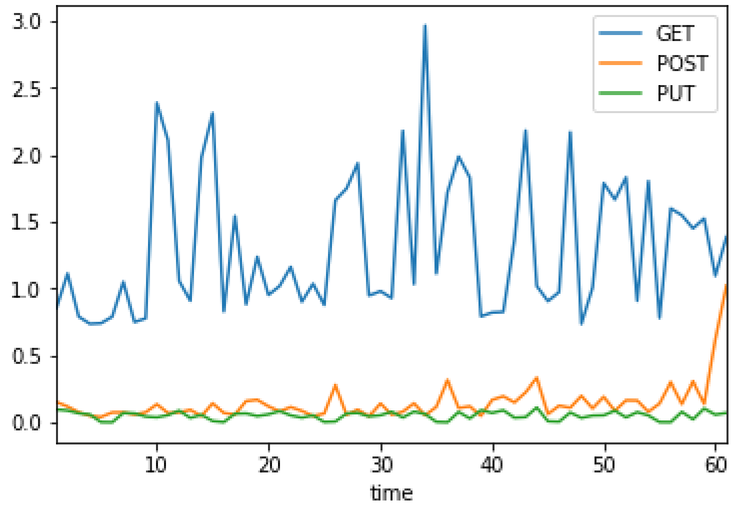
There are a better suited metrics to predict http5x failures than the one I use

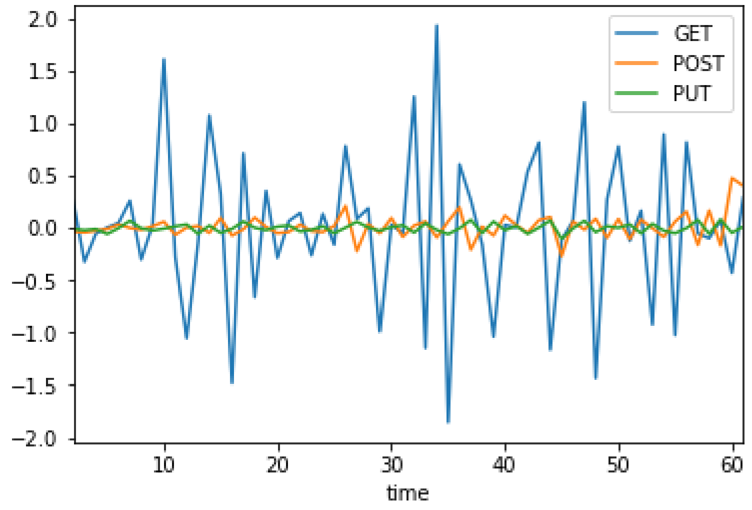
Choose method



Get metrics into the right format for method

- Training data with labels needed (X,y)
- Seasonally adjust





Apply feature selection algorithm

```
from sklearn.feature_selection import RFE
from sklearn.ensemble import RandomForestRegressor
...
# perform feature selection
rfe = RFE(
        RandomForestRegressor(
            n_estimators=500,
            random_state=1,
            min_samples_split=5
      ), 1)
fit = rfe.fit(X, y)
...
```

Selected Feature: POST

Feedback cycle

Rewrite your alerts and dashboards to use label POST to better predict http 5x errors

Example 3 - metrics / feature selection with library tsfresh

- Metrics selection / ranking similar to example 1
- Metrics extension by applying functions to metrics

https://github.com/blue-yonder/tsfresh

Prometheus datascience mantra

- Create hypothesis about your system and metrics
- Get metrics (devops) and convert them into the right format
- Use statistical methods to verify hypothesis
- Feedback results to system, the dashboards and alerts

Lessons learned

- Alert model improves with insights from descriptive statistics and ML!
- Depending on the result, correct, discard or handle data differently
- Day to day usecase: e.g. reduced try and error config on predict_linear function
- No need to process metrics streaming with ML/AI yet

Thx for having me here at promcon.io 2017!

Questions?

Georg Öttl
Twitter Handle: @goettl