Anbefalinger i forbindelse med maskinlæringsprosjekter

Team

Ikke ansett mange data scientists når du egentlig trenger ingeniører

Start med å definere metrics

- **Metric**: Et tall du bryr deg om.
  - Eksempler:
    - Andel agurker kategorisert riktig
    - Klikkrate
    - Forskjellen mellom prisestimat og markedsverdi

Start med en enkel løsning

- Begynn gjerne med håndkodede regler eller enkle algoritmer, hvis mulig

Start med enkle modeller

- Gjør det enklere å debugge
- Fokuser på å få på plass en god infrastruktur
- Deploy
- Integrer

Test!

Overgangen fra algoritmer og håndkodede regler

Logg eksperimentene og modellene dine

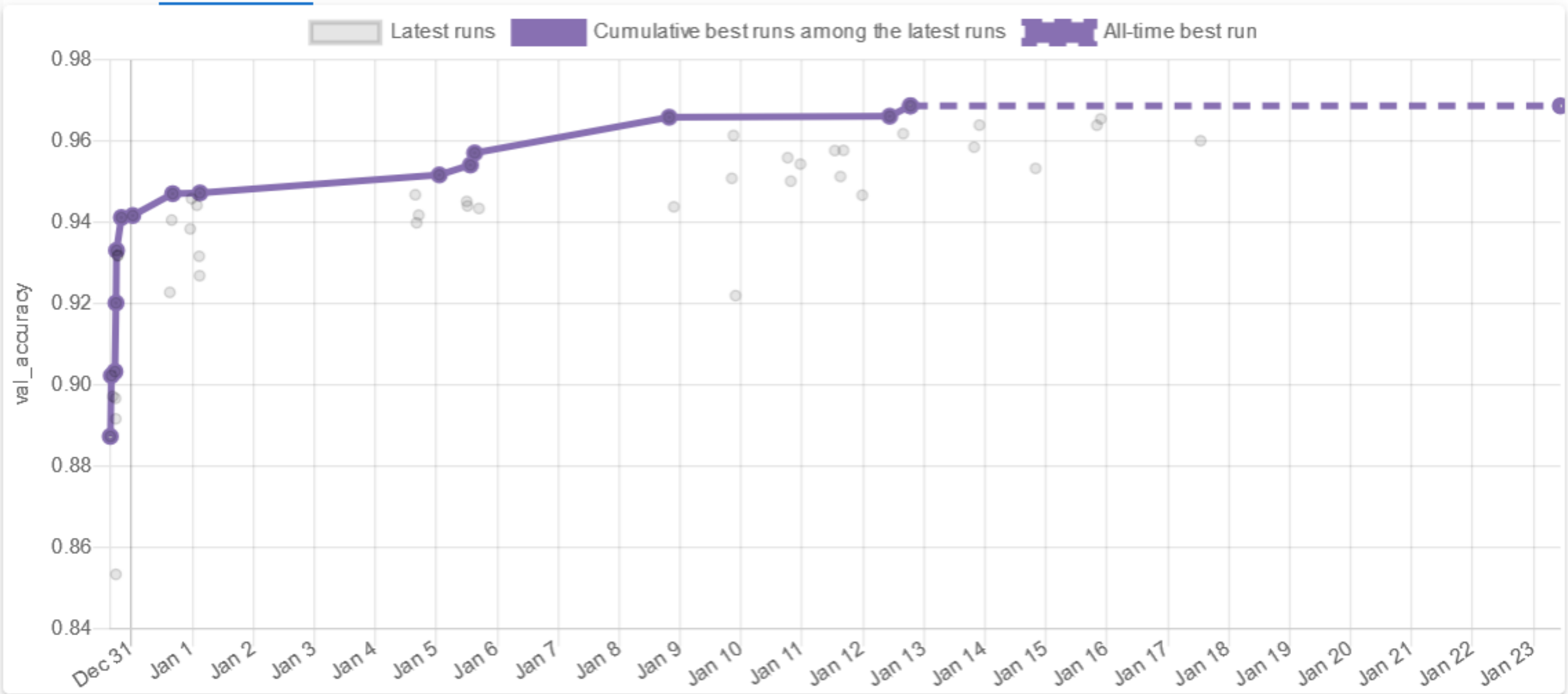
- MLflow (kan hostes på f.eks. Databricks eller Mflux.ai)
- Neptune.ml
- Comet.ml

```
$ pip install mflux-ai mlflow[extras]
```

```
import mlflow.sklearn
import mflux_ai

mflux_ai.init("your_project_key_goes_here")

mflux_ai.put_dataset(dataset, dataset_filename)
dataset = mflux_ai.get_dataset(dataset_filename)

mlflow.log_metric("validation_accuracy", validation_accuracy)
mlflow.log_param("model_type", "DecisionTreeClassifier")
mlflow.sklearn.log_model(model, "model")
```

Pronouncer

TABLE PLOT



Monitorering

- Sjekk oppetid
- Vær obs på tause feil – f.eks. datafordelinger kan endre seg
- Når du deployer en ny modell, sammenlign oppførselen med forrige modell

Sjekk edge cases


person 0.999
bird 0.995

Gene Kogan
@genekogan

CycleGAN "failure cases" are as interesting as the successes. ML papers need more blooper sections

14:17 - 20. apr. 2017

Hvor fersk må modellen din være?

- Tren modell på ferske data f.eks. hver dag eller hver uke
- Deploy nye modeller når de har bestått tester og bevist at de er best

A white cat is sitting on a windowsill

Prøv å tallfeste uønsket adferd

- Visualiser dataene, så du kan forstå hva som gikk galt
- Mål først, optimaliser etterpå

Du når et platå – hva nå?

- Feature engineering (bruk domenekunnskap!)
- Finn og integrer flere datakilder
- Mer data
- Mer komplekse modeller
- Deploy, overvåk, iterer
- Sjekk: Er det andre ting å gjøre som vil gi mer verdi?

«The data is the specification»

1. Instead of coming up with a good general solution, it is better to focus on solving specific cases of the problem.
2. Instead of trying to solve all problematic cases right away, it is better to address a proportion of the easiest cases and repeat the process multiple times.
3. Instead of writing a good specification of a solution, it is better to curate a collection of good problem cases. This collection of problem cases then essentially **becomes** your specification.

dataisspec.github.io

> *Do machine learning like the great engineer you are,*
> *not like the great machine learning expert you aren't*

-Google

Twitter: @iver56