Lab 0: Maskinlæring og PyCaret

ELMED219-2026

ELMED219

Vår 2026

Oversikt

Hva er Maskinlæring (ML)?

Maskinlæring er studiet av dataalgoritmer som forbedrer seg automatisk gjennom erfaring.

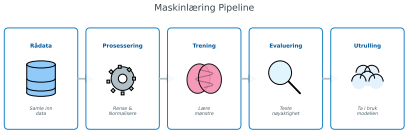
Formell definisjon (Tom Mitchell): Et dataprogram sies å lære fra erfaring **E** med hensyn til en oppgave **T** og ytelsesmål **P**, hvis ytelsen på T, målt ved P, forbedres med erfaring E.

I medisin:

- **T**: Diagnostisere en sykdom.
- **E**: Historiske pasientdata.
- **P**: Nøyaktighet (Accuracy).

Prompt

Et rent, horisontalt flytdiagram som illustrerer maskinlærings-pipelinen på hvit bakgrunn. Steg fra venstre mot høyre: 1) Rådata (database-ikon), 2) preprosessering (tannhjul), 3) Trening (hjerne-chip ikon), 4) Evaluering (forstørrelsesglass på graf), 5) Utrulling (sky). Stilen er moderne, 'Corporate Memphis' men strengt profesjonell og medisinsk-teknisk. Høy kvalitet, vektorstil."



ML-Paradigmer

- **Veiledet Læring (Supervised Learning):** Vi har input data X og fasit (labels) y. Målet er å lære en funksjon $f : X \to y$.
  - *Klassifikasjon:* y er en kategori (Syk/Frisk).
  - *Regresjon:* y er et kontinuerlig tall (Blodtrykk, Liggetid).
- **Ikke-veiledet Læring (Unsupervised Learning):** Vi har kun input data X. Målet er å finne struktur i dataene.
  - *Klynging (Clustering):* Finne pasientgrupper.
  - *Dimensjonsreduksjon:* Forenkle komplekse data.

Klassifikasjon vs. Regresjon

Klassifikasjon

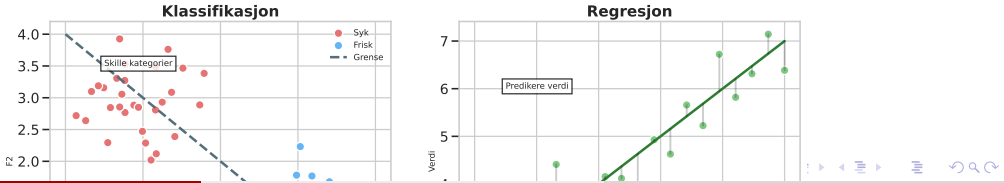
- Diskret output.
- Eks: Diagnoser (ICD-10 koder).
- Algoritmer: Logistic Regression, Decision Trees, Random Forest, SVM.

Regresjon

- Kontinuerlig output.
- Eks: Tid til hendelse, dosering.
- Algoritmer: Linear Regression, Random Forest Regressor, XGBoost.

Prompt

Delt skjerm vitenskapelig illustrasjon. Venstre side merket 'Klassifikasjon': Et spredningsplott hvor en linje skiller røde prikker fra blå prikker. Høyre side merket 'Regresjon': Et spredningsplott med en glatt kurve som tilpasses gjennom datapunktene. Ren, hvit bakgrunn, tydelige farger, høy oppløsning, lærebok-kvalitet."



Hvordan måler vi suksess?

For klassifikasjon bruker vi en **Forvirringsmatrise (Confusion Matrix)**:

	Predikert Positiv	**Predikert Negativ**
Faktisk Positiv	True Positive (TP)	False Negative (FN)
Faktisk Negativ	False Positive (FP)	True Negative (TN)

- **Accuracy:** $\frac{TP+TN}{Total}$ (Kan være misvisende ved ubalanserte data!)
- **Precision:** $\frac{TP}{TP+FP}$ (Hvor mange av de vi kalte syke, er faktisk syke?)
- **Recall (Sensitivitet):** $\frac{TP}{TP+FN}$ (Hvor mange av de syke fant vi?)
- **F1-Score:** Harmonisk gjennomsnitt av Precision og Recall.

Bias-Variance Tradeoff

En sentral utfordring i ML er balansen mellom *bias* og *varians*.

Underfitting (Høy Bias):

- Modellen er for enkel.
- Fanger ikke opp mønsteret i dataene.
- Dårlig på både trening og test.

Overfitting (Høy Varians):

- Modellen er for kompleks.
- Lærer støy"i treningsdataene.
- God på trening, dårlig på test.

Målet er å finne sweet spotsom generaliserer godt til nye data.

PyCaret: Automatisert ML

PyCaret er et low-codebibliotek som automatiserer store deler av ML-flyten.

- **setup():** Initialiserer miljøet, håndterer manglende verdier, koder kategoriske variabler, splitter data.
- **compare_models():** Trener og evaluerer mange algoritmer side-om-side.
- **create_model():** Trener en spesifikk modell.
- **tune_model():** Optimaliserer hyperparametere automatisk.
- **plot_model():** Genererer standardiserte figurer (ROC, Feature Importance).

Eksempel på PyCaret-kode

```
from pycaret.classification import *

# 1. Setup
exp = setup(data=diabetes_df, target='Class variable', session_id=123)

# 2. Sammenlign modeller
best_model = compare_models()

# 3. Analyser
evaluate_model(best_model)

# 4. Prediker
predictions = predict_model(best_model, data=new_data)
```

Oversikt over Lab 0 Notebooks

1. **01-Enkle_eksempler.ipynb**:
   - Manuell implementasjon av enkle ML-konsepter.
   - Forstå beslutningstrær og logistisk regresjon "under panseret".
2. **02-Binaer_klassifikasjon.ipynb**:
   - Klassifisere Pima Indians Diabetes datasett.
   - Fokus på datautforskning (EDA) og modellvalidering.
3. **03-PyCaret_hurtigguide.ipynb**:
   - Effektiv ML-flyt med PyCaret på samme datasett.
   - Sammenligning av hvor mye tid man sparer!

Viktige Læringspunkter

- **Data er viktigst:** Garbage in, garbage out". Bruk tid på å forstå dataene.
- **Validering:** Stol aldri på trenings-score alene. Bruk kryssvalidering eller et hold-out testsett.
- **Base-rate:** Sjekk alltid om modellen slår en enkel gjetning (f.eks. "alle er friske").
- **Tolkbarhet:** Noen ganger er en enkel modell (Lineær Regresjon) bedre enn en kompleks (Black Box) hvis vi trenger å forstå *hvorfor*.

Oppsummering

I dag har vi lagt grunnlaget for resten av kurset.

- Vi har definert ML.
- Vi har sett på forskjellen mellom klassifikasjon og regresjon.
- Vi har introdusert PyCaret som et kraftig verktøy.

I neste lab (Lab 1) skal vi se på hvordan vi kan bruke **Nettverksvitenskap** til å finne pasienter som ligner på hverandre (PSN).