

Kunstig intelligens og maskinlæring

ANBEFALINGER TIL TEKNOLOGISK RAMMEVERKSDOKUMENT

1	<i>Innledning</i>	2
2	<i>Dokumentasjon av maskinlæringsprosjekter</i>	2
2.1	Generelle krav til dokumentasjon	3
2.1.1	Model cards	3
2.1.2	«Datasheets for Datasets».....	4
2.2	Detaljerte krav til dokumentasjon	4
2.2.1	Prediksjonsresultater fra maskinlæringsmodellen skal overleveres	5
2.2.2	Ferdig trent modell skal kunne kjøres av andre	5
2.2.3	Andre skal kunne finjustere modellen – «Fine tuning»	6
2.2.4	Andre skal kunne videreutvikle modellen	6
2.2.5	Andre skal kunne reprodusere modellen	7
2.3	Livssyklusrammeverk	8
3	<i>Evalueringsmetodikk</i>	8
4	<i>Effektiv deling av data og modeller</i>	10
4.1	Gjøre eksisterende data lettere tilgjengelig for maskinlæring	10
4.2	Dele treningsdata eller resultater fra maskinlæring	10
4.3	Dele modeller	11
4.4	Tilby en felles infrastruktur for maskinlæring	11

1 Innledning

Kunstig intelligens (KI) og maskinlæring (ML) er områder under rask utvikling, hvor nye teknikker, modeller, produkter og anvendelsesområder dukker opp med høy frekvens. Mulighetene og potensialet i denne teknologien er stort og for mange aktører er god kompetanse, produkter og tjenester innen maskinlæring en viktig konkurransefordel. Dette kan være medvirkende årsaker til at det så langt finnes få eller ingen etablerte åpne standarder for maskinlæring, til tross for at mye av den underliggende teknologien er åpen kildekode. Likevel er det et stort behov for å utveksle kunnskap, data, modeller og teknikker for å videreutvikle og realisere verdien i maskinlæring. Derfor har det begynt å etablere seg noen beste praksiser som kan fungere godt som anbefalinger for hvordan maskinlæring bør gjennomføres og dokumenteres. Disse vil bidra til å sikre riktig bruk av modeller og modellresultater, gjenbruk og videreutvikling av modeller. Som et resultat finnes det også et økende utvalg av verktøy og rammeverk som gjør det enklere å følge disse anbefalingene. Dette dokumentet har som mål å presentere hvordan disse beste praksisene kan fungere som anbefalinger for å sikre bedre samarbeid, deling og verdiskapning innen kunstig intelligens og maskinlæring, som en del av den nye nasjonale geografiske infrastrukturen.

Arbeidet er utført høsten 2022 av en arbeidsgruppe bestående av følgende personer:

- Ivar Oveland (Kartverket)
- Lill Maria Johannesen (NOIS)
- Michael Pande Rolfsen (KS)
- Erik Hagen (DigDir)
- Jan Roger Andersen (Asplan Viak)
- Misganu Debella Gilo (Nibio)
- Nicolai Munsterhjelm (Nibio)
- Sigbjørn Dale (Klepp kommune)
- Mathilde Ørstavik (Norkart)
- Nils Olav Dale (Arkitektum)
- Inge Anundskås (Geodata)
- Trym Teigene (Oslo kommune)
- Harald Huse (Norkart, sekretær/leder)

2 Dokumentasjon av maskinlæringsprosjekter

Alle maskinlæringsprosjekter bør dokumenteres med mål om å gi et overblikk over modellens tiltenkte bruk, muligheter, kvalitet og begrensninger. Dokumentasjonen bør være relevant og lesbar både for teknisk personell med kjennskap til maskinlæring, men også andre som er interessert i modellens anvendelsesområder og resultater.

2.1 Generelle krav til dokumentasjon

Innenfor maskinlæringsfeltet har det ikke vært standardiserte prosesser for dokumentering av modeller og datasett, men det finnes foreslåtte rammeverk som bør benyttes.

Eksempler på dette er «Model Cards» og «Datasheets for Datasets». Disse rammeverkene har definerte punktlistor med forhold som burde belyses og ofte dokumenteres, for henholdsvis maskinlæringsmodeller og datasett.

2.1.1 Model cards

Formålet med et [Model Card](#) er å gi et kortfattet, helhetlig bilde av modellen. Hva modellen gjør, målgruppe, hvem som vedlikeholder, i tillegg til konstruksjon av modellen, som algoritme, arkitektur og treningsdata. Nøyaktighetsmål som benyttes for evaluering av modellen, i tillegg til ytelsesbegrensninger (svakheter) er også oppgitt.

Arbeidsgruppa anbefaler at modeller dokumenteres gjennom metoden «Model Cards». Denne gir en kort beskrivelse av modellen på 1 – 2 sider. Et forslag til disposisjon og innhold for et slikt kort følger under, men det oppfordres til å tilpasse inndelingen og innholdet til behovet til modellen, brukstilfellet og relevante interessenter.

- Modelldetaljer: Grunnleggende informasjon om modellen
 - Hvem har utviklet modellen? (person / organisasjon)
 - Når ble modellen utviklet?
 - Hva er modellens versjonsnummer?
 - Hva slags type modell er det? (Overordnet modellarkitektur)
 - Informasjon om treningsalgoritmer, parametere eller annen informasjon om tilnærming og metode
 - Lenke til mer informasjon eller eventuelt publikasjon
 - Riktig bruk av referanse til modellen
 - Lisens
 - Kontaktinformasjon
- Bruksområde: Hvilke brukertilfeller lå til grunn for utvikling av modellen?
 - Primære brukertilfeller
 - Primær målgruppe
 - Brukertilfeller som ikke dekkes av modellen
- Faktorer: Faktorer som kan påvirke modellens resultater f.eks. demografiske grupperinger eller miljømessige betingelser.
 - Relevante faktorer (f.eks. lysforhold, årstid, oppløsning, kjønn, alder etc.)
 - Evalueringsfaktorer (Dersom det ikke var mulig å evaluere alle faktorer, f.eks. grunnet manglende treningsdata)
- Måleparametere: Bør velges slik at de reflekterer faktiske utslag modellen kan gi ved bruk.
 - Modellresultater
 - Terskelverdier. Eventuelle terskelverdier som er benyttet
 - Usikkerhet
- Evalueringsdata: Detaljer om dataene brukt for evaluering av modellens kvalitet (se kap. 3 – Evalueringsmetodikk)
 - Hvorfor ble disse dataene benyttet til å evaluere modellen?

- Hvordan har dataene blitt prosessert for å benyttes i evalueringen?
- Treningsdata: Om mulig, tilsvarende som for evalueringsdata. Bør i det minste inneholde detaljer om den statistiske fordelingen til treningsdataenes faktorer.
- Kvantitative analyser av modellens faktorer
 - Modellens resultater for hver faktor
 - Modellens resultater ved kombinasjoner av flere av evalueringsfaktorene
- Etske hensyn: Om modellen er avhengig av personsensitive opplysninger, potensiale for misbruk etc.
- Forbehold og anbefalinger

For en komplett oversikt over et «model card», se [artikkelen](#).

2.1.2 «Datasheets for Datasets»

«[Datasheets for Dataset](#)» har som formål å gi en oversikt over forhold som bør vurderes og informeres om ved oppretting og bruk av treningsdatasett for maskinlæring.

Den gir for eksempel innsikt i hva datasettet inneholder, størrelsen på datasettet, eventuelle svakheter og innsamlingsmetode, og kan være til bistand både for videreutvikling av datasettet og for å forstå læringsgrunnlaget til eventuelle maskinlæringsmodeller.

Arbeidsgruppa anbefaler å benytte standarden ved å ta et utdrag av relevante punkter for det aktuelle prosjektet. Standarden er relativt omfattende, men eksempler på viktige punkter er:

- Beskrivelse av hva datasettet inneholder instanser av
 - F.eks dokumenter, bilder og land
- Antall bilder i datasettet
- Hvorvidt datasettet dekker alle typer instanser av det det representerer
 - Feks. at et bygningsdatasett kun inneholder instanser fra Sør-Norge
- Formatet på datasettet
- Eventuelle anbefalte måter å splitte datasettet til test-, validering- og trenings-sett
- Informasjon om eventuell støy eller mangler i datasettet
- Hvorvidt datasettet inneholder sensitive data
- Hvordan dataene ble produsert/innhentet

2.2 Detaljerte krav til dokumentasjon

Det er mange ulike prosjekter som kan benytte seg av maskinlæring, og forskjellige prosjekter vil ha ulike behov for videre bruk av modeller og resultater fra modellene. Derfor vil også kravene til dokumentasjon av datasett og maskinlæringsmodell variere, og et «Model Card» og «Datasheet for Datasets» dekker ikke nødvendigvis behovene for dokumentasjon.

Overordnet kan vi si at kravene til dokumentasjon vil variere ut ifra følgende kategorier/behov:

1. **Prediksjonsresultater fra maskinlæringsmodellen skal overleveres.** Dette er relevant dersom man bestiller en analyse som utføres av andre, og man kun har behov for sluttresultatet av analysen.

2. **Ferdig trent modell skal kunne kjøres av andre.** Dette er relevant dersom andre enn de som utvikler modellen ønsker å drifte modellen.
3. **Andre skal kunne finjustere modellen.** Dette er relevant dersom flere involverte parter som skal samarbeide om utviklingen, eller man bestiller en modell, og har intern kompetanse og treningsdata tilgjengelig for å kunne finjustere modellen (se beskrivelse under punkt 3) ved behov.
4. **Andre skal kunne videreutvikle modellen.** Dette er relevant dersom flere involverte parter som skal samarbeide om utviklingen.
5. **Man skal kunne reproducere modellen.** Dette er relevant for forskningsprosjekter hvor resultater skal være etterprøvbare.

2.2.1 Prediksjonsresultater fra maskinlæringsmodellen skal overleveres

Et slikt prosjekt vil i stor grad tilsvare andre analyseprosjekt, og her trengs ingen informasjon om maskinlæringsmodellen eller treningsdata. Derimot vil det være nyttig å få informasjon om resultatenes opprinnelse og nøyaktighet, som vil dekkes godt av et «model card».

Kort illustrasjon av brukerhistorie:

Problemstilling:	Ønsket resultat:
Byutvikler Bashim ønsker å undersøke effektene av Bilfritt Byliv, et knippe tiltak som ble igangsatt i 2016 for å redusere antall biler i Oslo sentrum. Han har derfor engasjert et konsulentselskap som har benyttet maskinlæring til å telle antall biler fra vegbilder, for å sammenligne trafikkmengden før og etter oppstart av Bilfritt Byliv prosjektet. Nå ønsker han å dele resultatene fra undersøkelsen med prosjektets interessenter og samarbeidspartnere.	Bashim mottar en rapport over antall biler som har kjørt på de utvalgte strekningene før og etter tiltakene, samt informasjon om nøyaktighetene til resultatene. Denne rapporten deles med interessenter og samarbeidspartnere.

2.2.2 Ferdig trent modell skal kunne kjøres av andre

Dersom man jobber med en ML-modell som skal overleveres og kunne kjøres/driftes av andre uten å endres, må følgende dokumenteres:

- Ferdig trent modell
- Kodemiljø og avhengigheter
- Informasjon om dataformatet som modellen tar som input

Kort illustrasjon av brukerhistorie:

Problemstilling:	Ønsket resultat:
Miljøkonsulent Mira ønsker å bruke maskinlæring og høyoppløselige satellittbilder til å kvantifisere fremveksten av Sitka-gran i Norge. Miljødirektoratet har kjøpt bistand fra et konsulentselskap som har utviklet og testet en modell for dette, for et begrenset testområde.	Mira har tilstrekkelig informasjon til å kjøre modellen fra konsulentselskapet i egen IT-infrastruktur, for å identifisere Sitka-gran på satellittbilder over hele landet.

2.2.3 Andre skal kunne finjustere modellen – «Fine tuning»

En teknikk innen maskinlæring kalt «fine tuning» innebærer at man benytter en trent modell og trener modellen videre på samme type data for å fin-tilpasse modellen. Et eksempel kan være å ta en modell som er god på bygningsgjenkjenning i Sør-Norge og trene videre på data for Nord-Norge.

For å kunne finjustere en modell trengs følgende:

- Ferdig trent modell
- Format på treningsdataene
- Arkitektur og kode for modellen
- Type data og format
- Om det er benyttet noen form for «data augmentation» under trening
 - F.eks. rotering av bilder

Kort illustrasjon av brukerhistorie:

Problemstilling:	Ønsket resultat:
GIS-konsulent Gunnar jobber i et prosjekt for fylkeskartkontoret i Møre og Romsdal som skal bruke maskinlæring til å identifisere bygninger i strandsona som ikke er matrikkelført. Siden Fylkeskartkontoret i Agder nettopp har gjennomført et liknende prosjekt ønsker han å ta utgangspunkt i denne modellen, men tilpasse den for lokale forhold.	Gunnar har tilstrekkelig informasjon til å trene modellen fra Agder ytterligere, med nye flyfoto og FKB-data fra Møre og Romsdal.

2.2.4 Andre skal kunne videreutvikle modellen

Det er flere måter å videreutvikle en maskinlæringsmodell.

En type videreutvikling kan være at man ønsker å ta utgangspunkt i arkitekturen til modellen, og gjøre videre endringer/utvidelser som f.eks. å teste modellen med andre hyperparametre (f.eks. activation functions, loss functions, batch-størrelse).

En annen type videreutvikling kalles for «transfer learning», hvor man ønsker å beholde arkitekturen, men trene modellen videre på en annen type data, og på den måten endre oppgaven modellen løser. Denne teknikken er nyttig fordi den minsker kravene til mengde treningsdata, ettersom modellen kan gjenbruke egenskaper den allerede har lært fra det tidligere løste problemet. Et eksempel kan være at man tar en modell som kan gjenkjenne katter, og trener den videre til å kunne gjenkjenne bygninger i flyfoto i stedet.

For å kunne videreutvikle en modell, trengs følgende:

- Ferdig trent modell
- Format på treningsdataene
- Arkitektur og kode for modellen

Kort illustrasjon av brukerhistorie:

Problemstilling:	Ønsket resultat:
Klimaforsker og utvikler Kine ønsker å finne isfjell i satellittbilder ved hjelp av maskinlæring. Siden det allerede finnes gode modeller for å identifisere skip i slike bilder, ønsker hun å benytte disse modellene som utgangspunkt for sitt prosjekt.	Kine har tilstrekkelig informasjon om modellen for å identifisere skip slik at hun kan trene denne modellen med egne data slik at den kan identifisere isfjell i tilsvarende satellittbilder fra Nordishavet («transfer learning»).

For å kunne videreutvikle modellen, trengs følgende:

- Ferdig trent modell
- Format på treningsdataene
- Arkitektur og kode for modellen

2.2.5 Andre skal kunne reprodusere modellen

At en modell er reproduserbar vil si at man skal kunne trene modellen på nytt, og få en modell som gir samme resultater ved prediksjon på samme data.

For å kunne reprodusere en modell trengs følgende:

- Arkitekturen til modellen (med kode, kodemiljø og avhengigheter) - anbefales, et alternativ kan være tilstrekkelig med skriftlig dokumentasjon til å kunne implementere tilsvarende arkitektur.
- Hyperparametre ved trening (som ikke fremkommer av eventuell kode)
 - Ved f.eks. deep learning ville dette kunne vært loss function, activation function, batch-size, optimizer, antall epoker trent osv
- Treningsdatasett, enten et statisk datasett eller metadata til å kunne gjenskape tilsvarende datasett
- Beskrivelse av hvordan treningsdataene ble behandlet
 - Data preparation, data augmentation/feature engineering

Livssyklusrammeverk som nevnt i 2.3 kan hjelpe til med å sikre reproduserbarhet og dokumentasjon.

Kort illustrasjon av brukerhistorie:

Problemstilling:	Ønsket resultat:
Skogkonsulent og utvikler Stine deltar i et forskningsprosjekt som har som målsetning å identifisere vindfall i skogen ved hjelp av maskinlæring og dronebilder. Arbeidet har i stor grad blitt utført av doktorgradsstudenter ved NMBU. Nå ønsker hun å verifisere modellene som har blitt produsert for å avgjøre om prosjektet bør videreføres.	Stine har tilstrekkelig informasjon til å trene den implementerte modellen som produserer samme resultater ved prediksjon som modellen utviklet av NMBU-studentene.

2.3 Livssyklusrammeverk

For å kunne dokumentere og ha oversikt over et maskinlæringsprosjekt er det mye man skal ha kontroll på. Man må ha kontroll på versjonering av modeller, versjonering av data, hvilke hyperparametre som er brukt under trening, i tillegg til kodemiljø og avhengigheter. I tillegg ønsker man ofte å kunne kjøre modellene i ulike miljøer – både lokalt og i skyen. Derfor har det blitt utviklet ulike verktøy, som kan kalles for livssyklusrammeverk, som gjør dette enklere.

Et livssyklusrammeverk bidrar med å løse utfordringer knyttet til:

- Reproduserbarhet av modeller
- Versjonering og livssyklushåndtering av modeller
- Produksjonssetting av modeller
- Versjonskontroll og dokumentasjon av trenings- og testdata

Eksempler på livssyklusrammeverk er [Mlflow](#) til modeller, og [DVC](#) til data.

3 Evalueringsmetodikk

Når resultatene fra en maskinlæringsmodell skal evalueres og dokumenteres er det flere hensyn som må tas for at evalueringen skal gi et godt bilde på modellens faktiske prestasjon. Hvilke data som brukes og hvilket nøyaktighetsmål som velges er avgjørende for å sikre en riktig og rettferdig evaluering av modellen.

Datasettet deles gjerne i tre deler:

- 1) **Trening:** til å trene maskinlæringsmodellen med
- 2) **Validering:** til å evaluere modellen underveis i treningen
- 3) **Test:** til å evaluere den endelige modellen når den er ferdig trent



Figure 1: Treningsdata deles gjerne i tre deler. Valideringsdata benyttes for å evaluere modellen underveis i treningen, mens testdata benyttes for å evaluere modellen etter den er ferdig trent.

Til modellevaluering er det viktig å benytte data som er uavhengige fra dataene som modellen har blitt og trent og validert med. Dersom man evaluerer modellen ved bruk av trenings- eller valideringsdata, vil dette ikke kunne gi en vurdering av nøyaktigheten til modellen ettersom dette er data modellen allerede har sett. I stedet skal test-delen av datasettet benyttes.

Testdataene bør være representative for hele datasettet ellers. Dersom testdataene inneholder større andel «enkle» data enn treningsdataene, vil evalueringen gi et unaturlig godt resultat.

Evalueringsdataene må derfor være realistisk fordelt i forhold til faktorer som er relevant for modellen. Faktorene vil variere fra case til case, men det kan f.eks. være geografisk plassering, årstid, bildekvalitet, og ikke minst fasitklassen. Relevante faktorer bør inkluderes i dokumentasjonen.

I tillegg må testdatasettet være av en viss størrelse for å kunne gi pålitelige resultater. Det er vanlig å sette av mellom 10-30% av det totale datasettet til evaluering av modellen.

Et nøyaktighetsmål skal være en god indikasjon på maskinlæringsmodellens prestasjon. En høyere verdi bør bety at modellen presterer bedre. Samtidig er det ofte ønskelig at nøyaktighetsmålet skal være forståelig og kommuniserbart. Det er ikke alltid mulig å oppnå begge disse ved bruk av kun ett nøyaktighetsmål.

Det finnes en rekke ulike nøyaktighetsmål med forskjellige styrker og svakheter. I tabellen nedenfor finnes noen eksempler på ofte brukte nøyaktighetsmål til tre ulike formål:

Formål	Ofte brukte nøyaktighetsmål
Klassifisering	Precision/Recall, F1, MCC, Accuracy, Kappa, ROC AUC, Forvirringsmatrise
Objektdeteksjon/Bildesegmentering	IoU/Jaccard coefficient, Dice coefficient. Alternativt et klassifiseringsmål nevnt ovenfor.
Regresjon	Mean Squared Error, Root Mean Square Error, Mean Absolute Error, R^2 , Adjusted R^2

Scikit-learn har en god beskrivelse av disse, i tillegg til mange flere: https://scikit-learn.org/stable/modules/model_evaluation.html

4 Effektiv deling av data og modeller

Effektiv deling forutsetter en infrastruktur som er godt tilrettelagt. Deling av maskinlæringsdata, modell, kode og dokumentasjon kan være utfordrende innenfor eksisterende delingsløsninger. De er i utgangspunktet ikke tilrettelagt for deling av maskinlæringsprosjekter og -resultater.

GeoNorge er en veletablert plattform for å dele geografiske data og dermed en spennende kandidat for også å dele treningsdata og maskinlæringsmodeller. Gjennom enklere tilgang til etablerte modeller og tilrettede treningsdata kan man se for seg både økt gjenbruk og samarbeid innenfor fagfeltet kunstig intelligens og maskinlæring. En eventuell tilrettelegging av GeoNorge for dette formålet kan gjøres med flere ambisjonsnivå, fra lavt til høyt:

1. Gjøre eksisterende data lettere tilgjengelig for maskinlæring
2. Dele trenings-, evalueringsdata eller resultater fra maskinlæring
3. Dele modeller
4. Tilby en felles infrastruktur for maskinlæring

4.1 Gjøre eksisterende data lettere tilgjengelig for maskinlæring

GeoNorge inneholder allerede en rekke data som kan være interessante å bruke som treningsdata i maskinlæringsprosjekter. For eksempel vil FKB-bygning være et godt utgangspunkt for å lage en modell for å identifisere bygninger i flyfoto. For å gjøre det lettere å bruke data fra GeoNorge i maskinlæringsprosjekter kan det være interessant å se på følgende tilrettelegging:

- Tilgjengeliggjøre årsversjoner av data i GeoNorge. I eksemplet hvor vi ønsker å bruke FKB-bygning som utgangspunkt for objekt-deteksjon er det viktig at FKB-dataene og flyfotoene som skal benyttes er så nær hverandre i tid som mulig. Dette for å unngå at det har skjedd store endringer fra flyfotoene til tidspunktet FKB sist ble ajourholdt.
- Tilby et grensesnitt bedre tilpasset utviklere og dataforskere. Dataene på GeoNorge er først og fremst tilrettelagt for nedlastning eller bruk i GIS-systemer via OGC-grensesnitt som f.eks. WFS eller WMS. Dette er formater som er verken er kjent eller hensiktsmessige for utviklere. Det bør vurderes å tilpasse dokumentasjon / metadata og grensesnitt (f.eks. REST) til utviklere og dataforskere som typisk vil være målgruppen for bruk av denne type data i maskinlæringsprosjekter.

4.2 Dele treningsdata eller resultater fra maskinlæring

GeoNorge er en plattform for å dele geografiske data og dermed kan man tenke seg at den også kan være velegnet for å dele trenings-, evalueringsdata eller resultater fra maskinlæring. Det er imidlertid viktig å være klar over at det kan være store forskjeller på dataene som i dag er tilgjengelig i GeoNorge og typiske treningsdata.

Data som i dag er tilgjengelig i GeoNorge er gjerne autoritative data som er kvalitetskontrollerte, godt dokumenterte og gjerne landsdekkende. Disse er ment for å brukes i temakart, i analyser og som

beslutningsgrunnlag. Treningsdata vil ofte ha helt andre egenskaper. Et eksempel kan være at man har gjort feltobservasjoner for å bruke som fasitdata til å trene en maskinlæringsmodell. Disse observasjonene vil ikke nødvendigvis være komplette, ha konsistent kvalitet/nøyaktighet eller være representative for eller dekke hele landet. De kan likevel være svært nyttige til å trene opp en modell for regional bruk, men ikke være egnet i et temakart eller som et beslutningsgrunnlag i seg selv.

Det blir dermed svært viktig å skille mellom tiltenkt bruksområde og begrensinger for data som gjøres tilgjengelig på GeoNorge. Informasjon om dette må være enkel å forstå, både for utviklere, dataforskere og GIS-ingeniører. En mulighet er å legge maskinlæringsdata i en helt egen side/portal på GeoNorge, for å unngå at de forveksles med andre datasett. Det vil også være naturlig å stille andre krav til metadata og dokumentasjon for treningsdata enn øvrige data på GeoNorge. Konkrete forslag til krav om metadata om treningsdata er belyst i øvrig tekst i dette dokumentet.

4.3 Dele modeller

Som beskrevet i kapittel 2.2, vil videre bruk av modeller sette ulike krav til dokumentasjon hva som faktisk må deles av treningsdata, evalueringsdata, parametere, kode for tilrettelegging etc. Det kan fort bli omfattende å implementere en løsning som legger til rette for at modeller kan deles med formål om å kunne gjenskapes. Her finnes det også eksisterende verktøy som f.eks. MLFlow som kanskje er bedre egnet til å ivareta og dokumentere alt som trengs for et slikt scenario. Et naturlig første mål kan være å tilrettelegge for deling av modeller på en måte som gjør at de kan kjøres av andre, men ikke gjenskapes, videreutvikles eller finjusteres.

4.4 Tilby en felles infrastruktur for maskinlæring

Både utvikling og bruk av maskinlæringsmodeller kan potensielt være svært ressurskrevende, både i forhold til tidsbruk, kompetanse og maskinvare / infrastruktur. Å tilby infrastruktur og tjenester for maskinlæring er derfor et viktig fokusområde for alle de store skyleverandørene og et område under rask utvikling. Dermed kan det være vanskelig, og gjerne utenfor omfanget av dette arbeidet, å kunne gjøre en god vurdering hvorvidt det er verdifullt å i det hele tatt vurdere en felles infrastruktur for å kjøre eller eventuelt også utvikle maskinlæringsmodeller. Tilsvarende vil det være enda vanskeligere å komme med anbefalinger eller krav til hvordan en slik infrastruktur bør se ut.