

MLOps på Google Cloud Platform

Forstudierapport

Ingvild Andersen, Jon Akselberg Langholm, Erik Flæsen Dalen

13. mai 2021



Statens vegvesen

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
02.02.2021	1.0	Første versjon av forstudierapport	Erik F.D., Ingvild A. og Jon L.
05.02.2021	1.1	Mindre revisjon med bakgrunn i tilbakemelding fra veileder Torgeir.	Jon L.
13.05.2021	1.2	Språkvask, mindre endringer i setningsoppbygginger	Erik F. D.

Innhold

1 Introduksjon	5
1.1 Oversikt over dokumentet	5
2 Bakgrunn	6
2.1 Dagens system og rutiner	7
3 Prosjektmål	9
3.1 Effektmål	9
3.2 Resultatmål	9
3.3 Prosessmål	10
3.4 Prosjektets omfang	10
3.4.1 Prosjektgruppen skal	11
3.4.2 Prosjektgruppen skal ikke	11
3.4.3 Videre begrensning av omfang	12
3.5 Produktets funksjonelle egenskaper	12
3.5.1 Ikke-funksjonelle egenskaper og krav	16
3.6 Milepæler og hovedaktiviteter	17
4 Interessenter og rammebetingelser	19
4.1 Interessentanalyse	19
4.2 Rammebetingelser	20
5 Kritiske suksessfaktorer	22
5.1 Suksessfaktorer	22
5.2 Informasjonsbehov	23
6 Risikoanalyse	25
7 Kost/nytte-analyse	28
7.1 Kvantifiserbar og ikke-kvantifiserbar nytte	29
7.1.1 Kvantifiserbar nytte	29
7.1.2 Ikke-kvantifiserbar nytte	29
7.2 Bortfall av direkte kostnader	30
7.3 Estimerte kostnader	31
7.4 Sammenstilling av kost/nytte	31

8 Retningslinjer og standarder	32
8.1 Krav til dokumentasjon	32
8.2 Krav til kvalitetsgjennomganger	32
8.3 Krav til standarder og metoder	33
8.4 Endringshåndtering	33
9 Prosjektorganisering	35
10 Anbefaling om videre arbeid	37
Referanser	38

Figurer

2 Dagens system	8
3 Milepæler for prosjektet	18
4 Stipulert arbeidsprosess	18
5 Interessenter	20
6 Suksessfaktorer	22
7 Informasjonsbehov	24
8 Risikoer	25
9 Risikoer i risikotabell	26
10 Strategier for å håndtere risikomomenter	27
11 Prosjektorganisering	35

1 Introduksjon

Alle prosjekter begynner med en idé, men ikke alle idéer bør bli prosjekter. Forstudiens hensikt er å kartlegge om en prosjektidé lar seg realisere innen gitte ressurs-, tids- og kostnadsrammer. Formålet er å avdekke om prosjektet kan gjennomføres slik oppdragsgiver ønsker, eventuelt hvilke endringer som må til for å støtte opp om gjennomførbarhet. Dokumentet skaper en avtale mellom oppdragsgiver og oppdragstaker, og bidrar til at begge parter sitter på samme type informasjon. Forstudien beskriver også hvordan det planlegges å gjennomføre prosjektet, og skal fremstilles slik at den finnes like forståelig for leser som for forfatter. I denne konteksten sitter oppdragsgiver på mye faglig kunnskap, så faglige begreper kan benyttes. Forstudierapporten gir en overordnet oversikt over prosjektets forventede resultat, men det tas forbehold om at resultatet kan endre seg gjennom den videre prosjektperioden.

1.1 Oversikt over dokumentet

Forstudien introduseres med **dokumentets hensikt (1)** og en kort **oversikt (1.1)**. Videre presenteres **bakgrunnen for prosjektet (2)**, nærmere bestemt en overordnet beskrivelse av bedriften, deres situasjon, problemer og ønsker. Neste del av dokumentet går inn på **prosjektmål (3)**, helt konkret **effektmål (3.1)**, **resultatmål (3.2)** og **prosessmål (3.3)**. Avsnittet beskriver også **prosjektets omfang (3.4)**, som avgrenser hva prosjektgruppen skal og ikke skal gjøre gjennom prosjektperioden. Her synliggjøres også **produktets funksjonelle egenskaper (3.5)** og **milepæler og hovedaktiviteter (3.6)**. I fjerde avsnitt analyseres **interessenter og rammebetingelser (4)**. Her informeres leser om hvem som er de involverte i prosjektets gjennomføring og resultat, samt prosjektets rammebetingelser. Midtveis i forstudien foreligger prosjektets **kritiske suksessfaktorer (5)**, i tillegg til en oversikt over prosjektets **informasjonsbehov (5.2)**. Dette er faktorer som kreves for at prosjektet skal lykkes, og dokumentasjon som støtter opp under disse. Femte avsnitt omfatter en **risikoanalyse (6)**, som setter søkelys på hendelser som kan ha negativ innvirkning på prosjektresultatet, faren for at disse skal finne sted, samt tiltak for å redusere konsekvensen av hendelsene. Neste

avsnitt går inn på forstudiens **kost/nytte-analyse (7)**, der man forsøker å tallfeste både nytten og kostnaden av en prosjektgjennomføring. Da disse verdiene er minimale og vanskelige å tallfeste i denne forstudien, er sammenstillingen av kost/nytte uteblitt. Videre går rapporten inn på **retningslinjer og standarder (8)**. Her presenteres krav til dokumentasjon, kvalitertsgjennomganger, standarder, metoder og endringshåndtering. Til slutt får lesere en oversikt over **prosjektorganisasjonen (9)**, som beskriver hvem som er med i prosjektet og deres arbeidsfordeling, samt en **anbefaling om videre arbeid (10)**.

2 Bakgrunn

Statens vegvesen er et nasjonalt forvaltningsorgan underlagt Samferdselsdepartementet. De har ansvar for å forvalte, planlegge, bygge, drifte og vedlikeholde de norske riksveiene.¹ Organisasjonen ønsker å bli mer datadrevet, som har ført til utvikling av dataplattformen Saga. Teamet bak plattformen består av interne ansatte fra Statens vegvesen, samt eksterne konsulenter fra selskapene Bekk og Webstep. Den underliggende infrastrukturen kjører på Google Cloud Platform. Deres oppgave er å utvikle forskjellige moduler i denne dataplattformen for å kunne samle inn og analysere store datamengder. I dag utføres eksempelvis mange manuelle oppgaver på oppdrag fra Statens vegvesen som i teorien kan automatiseres ved hjelp av maskinlæringsmodeller som trenes på disse datamengdene. En automatisering av manuelle arbeidsoppgaver kan friggi ressurser og tidsbruk, som Vegvesenet kan prioritere på andre områder. Målet er at de skal kunne bruke maskinlæringsmodeller for å gjøre forutsigelser rundt diverse formål som omfatter forvaltningsorganets virke. Oppdragsgiver ønsker å få forslag på hvordan man effektivt kan sette disse modellene i produksjon, og om dette er teknisk mulig. Dette beskrives videre som en produksjonspipeline.

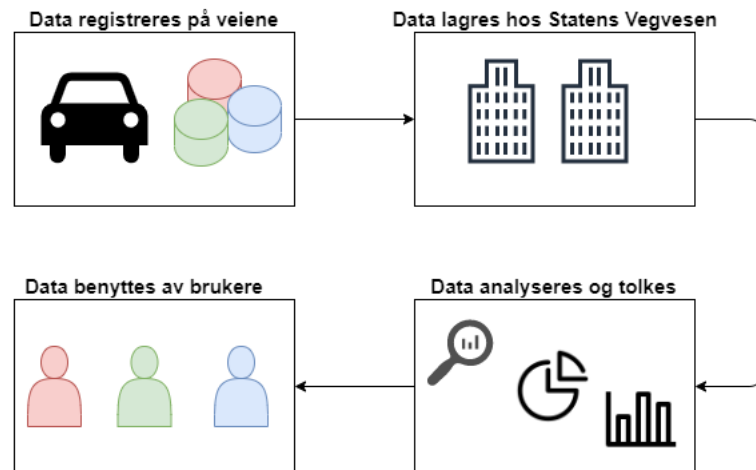
Med bakgrunn i dette behovet, har prosjektgruppen fått som oppgave å utarbeide en 'Proof of Concept' på en produksjonspipeline for en maskinlæringsmodell i Google Cloud Platform. Denne plattformen ønsker oppdragsgiver at skal brukes i flest mulig steg av produksjonspipelinen. Bedriften ønsker forslag som kan vise hvordan konseptet kan gjennomføres

konkret. Dette innebærer klart definerte suksesskriterier, dokumentasjon for gjennomførelse, en evalueringskomponent og et forslag for videre arbeid. Et 'Proof of Concept' innebærer også at man identifiserer potensielle tekniske og logistiske problemer som kan komme i veien for produksjonsprosessen.

2.1 Dagens system og rutiner

Dagens løsning eksisterer ikke, og innehar derfor ingen identifiserbare problemer. Per i dag samler Statens vegvesen inn store mengder data som kan hentes ut, leses og analyseres manuelt eller av programvare. Idéen om en løsning som går ut på å bruke disse datamengdene til å trene opp en maskinlæringsmodell for automatisering av arbeidsoppgaver, er nylig påbegynt. Behovet internt er å skape en produksjonspipeline for å standardisere prosessen ved bruk av maskinlæring fra første stund.

Selv om dagens system ikke eksisterer, er det i planleggingsfasen, orkestrert av teamet bak dataplattformen Saga. Basert på de store mengdene innhentet data skal maskinlæringsmodeller utvikles og trenes av enten innleide datavitere eller interne ansatte i Statens vegvesen. Disse maskinlæringsmodellene skal trenes og videreutvikles i Google Cloud Platform. Når modellen blir satt i produksjon, skal den kunne innhente nye datamengder. Disse datamengdene må valideres og kontrolleres, samt forprosesseres slik at de kan leses av pipelinen. Modellen må kunne trenes, justeres og evalueres slik at den alltid fungerer optimalt. Saga eies som nevnt av Statens vegvesen, som også finansierer videreutviklingen av prosjektet. Når dette prosjektet er ferdigstilt, vil produktet overtas av Saga, og videreutvikles utover dette prosjektets omfang.



Figur 2: Dagens system

Data samles inn av sensorer eller manuelle oppgaver, for så at disse lagres i store datalagre hos Statens vegvesen. Datamengdene analyseres og tolkes, før de presenteres for brukerne.

3 Prosjektmål

Prosjektmålene skal være en felles forståelse mellom oppdragsgiver og oppdragstaker. De skal lede begge parter og prosjektet mot felles mål. Målene er utarbeidet i samarbeid med oppdragsgiver, slik at man opprettholder en felles forståelse gjennom hele prosjektperioden. Prosjektmålene er delt inn i effektmål, resultatmål og prosessmål.

3.1 Effektmål

Effektmålene beskriver den endringen Statens vegvesen ønsker å få ut av prosjektet. Dette er prosjektets langsiktige virkninger for virksomheten.

- Ta i bruk mer moderne teknologi i veisektoren.
- Innhente forslag til hvordan de kan kjøre det foreslåtte systemet i produksjon.
- Et konkret produkt som kan testes ut.
- Kunne vurdere om dette er retningen de ønsker å bevege seg i, eller om de vil utforske alternativer.

3.2 Resultatmål

Resultatmål beskriver hvilke resultater som skal foreligge når prosjektet er ferdigstilt, samt hva som må gjøres for å oppnå disse resultatene. De gir altså en pekepinn på hva som konkret må gjøres.

Hovedmål:

- Et ferdigstilt 'Proof of Concept' på en produksjonspipeline for en maskinlæringsmodell i GCP.
- Ha resultatet klart innen 20.05.2021

Delmål:

- Kunne kjøre ut en maskinlæringsmodell i GCP.
- Kunne kjøre ut nye versjoner av en maskinlæringsmodell i GCP.

- Kunne eksponere en maskinlæringsmodell for input og bruke den fra andre applikasjoner.
- Finne ut hvordan man kan overvåke en maskinlæringsmodell og dens evne til å predikere riktig.
- Finne ut hvordan man kan skalere en pipeline med hensyn på kapasitet og ytelse.

Merknad: Det er ikke et mål i seg selv å ferdigstille alle punktene i oppgavebeskrivelsen eller alle disse delmålene, men å gjennomføre det prosjektgruppen evner på best mulig måte.

3.3 Prosessmål

Prosessmålene handler om hvilken verdi som skal oppnås som følge av selve prosjektprosessen. Disse målene består av både individuelle og kollektive forventninger til prosjektet, i tillegg til hva hvert enkelt gruppemedlem forventer å sitte igjen med etter endt prosjektperiode. Disse målene er kompetansebasert og består av studie-, sosiale- og tekniske kompetansemål.

- Økt kunnskap innen GCP og dens underliggende verktøy.
- Kompetanse på hvordan man planlegger og innfører en produksjonspipeline for maskinlæringsmodeller i GCP.
- Styrkede samarbeidsevner i teamarbeid med færre gruppemedlemmer.
- Økt kunnskap om individuelle styrker og svakheter i gruppesammenheng
- Erfaring rundt kommunikasjon og samarbeid med en ekstern, reell bedrift
- Høy faglig måloppnåelse

3.4 Prosjektets omfang

Denne delen av forstudien avklarer omfanget av prosjektet, og det er her man avgrenser oppgaven. En slik avgrensning fører til at oppdragsgiver og oppdragstaker er samstemte om nøyaktig hva prosjektet omfatter. Dette bidrar til at prosjektgruppen ikke bruker unødige ressurser på spesifikasjoner

som ikke inngår i prosjektomfanget. Produksjonspipelinen skal effektivisere ressursbruken i Statens vegvesen, og den legger til rette for at organisasjonen får utnyttet sine datamengder. Dette krever klare suksesskriterier for hvordan pipelinen skal operere, samt nøye utarbeidet dokumentasjon slik at oppdragsgiver enkelt og effektivt kan fortsette videre arbeid. Det er viktig å spesifisere at prosjektgruppen ikke har som ansvar å implementere eller drifte produksjonspipelinen som utarbeides, ei heller å utvikle selve maskinlæringsmodellen. Det skal heller ikke tillegges funksjonalitet som ikke er relevant for oppgavebeskrivelsen.

3.4.1 Prosjektgruppen skal

- Utarbeide et 'Proof of Concept' på en produksjonspipeline for en maskinlæringsmodell i Google Cloud Platform tilpasset behov hos Statens vegvesen
- Definere klare suksesskriterier for produksjonspipelinen
- Dokumentere hvordan 'Proof of Concept' gjennomføres slik at bedriften kan gjenskape resultatene
- Komme med forslag om videre arbeid om 'Proof of Concept' viser seg vellykket

3.4.2 Prosjektgruppen skal ikke

- Utvikle en maskinlæringsmodell
- Drifte produksjonspipeline
- Implementere produksjonspipeline
- Legge til funksjonalitet som ikke er relevant for oppgavebeskrivelsen
- Drive opplæring i nye systemer
- Sette opp miljø i Google Cloud Platform

3.4.3 Videre begrensning av omfang

Det er viktig å påpeke at oppgaven er bred, og vil sannsynligvis avgrenses i enda større grad i etterkant av forstudien. Dette er avklart og vil skje i samarbeid med oppdragsgiver. Bakgrunnen for dette er at omfanget av oppgaven også er ukjent for oppdragsgiver, da de heller ikke har tidligere erfaring med å gjennomføre et slikt prosjekt. Derav behøves det et 'Proof of Concept'. Om det viser seg at man må skalere ned omfanget, vil dette foregå etter den prioriterte oppgavebeskrivelsen som ble gitt ved prosjektstart. I prioritert rekkefølge:

1. Deploye maskinlæringsmodeller i GCP
2. Deploye nye versjoner av samme maskinlæringsmodell
3. Eksponere maskinlæringsmodeller for input
4. Bruke maskinlæringsmodeller fra andre applikasjoner
5. Overvåke maskinlæringsmodellen og dens evne til å predikere riktig
6. Skalere pipeline med hensyn på kapasitet/ytelse
7. Finne ut hvordan man kan trene maskinlæringsmodeller effektivt
8. Finne ut hvordan man kan lagre maskinlæringsmodeller på en hensiktsmessig måte

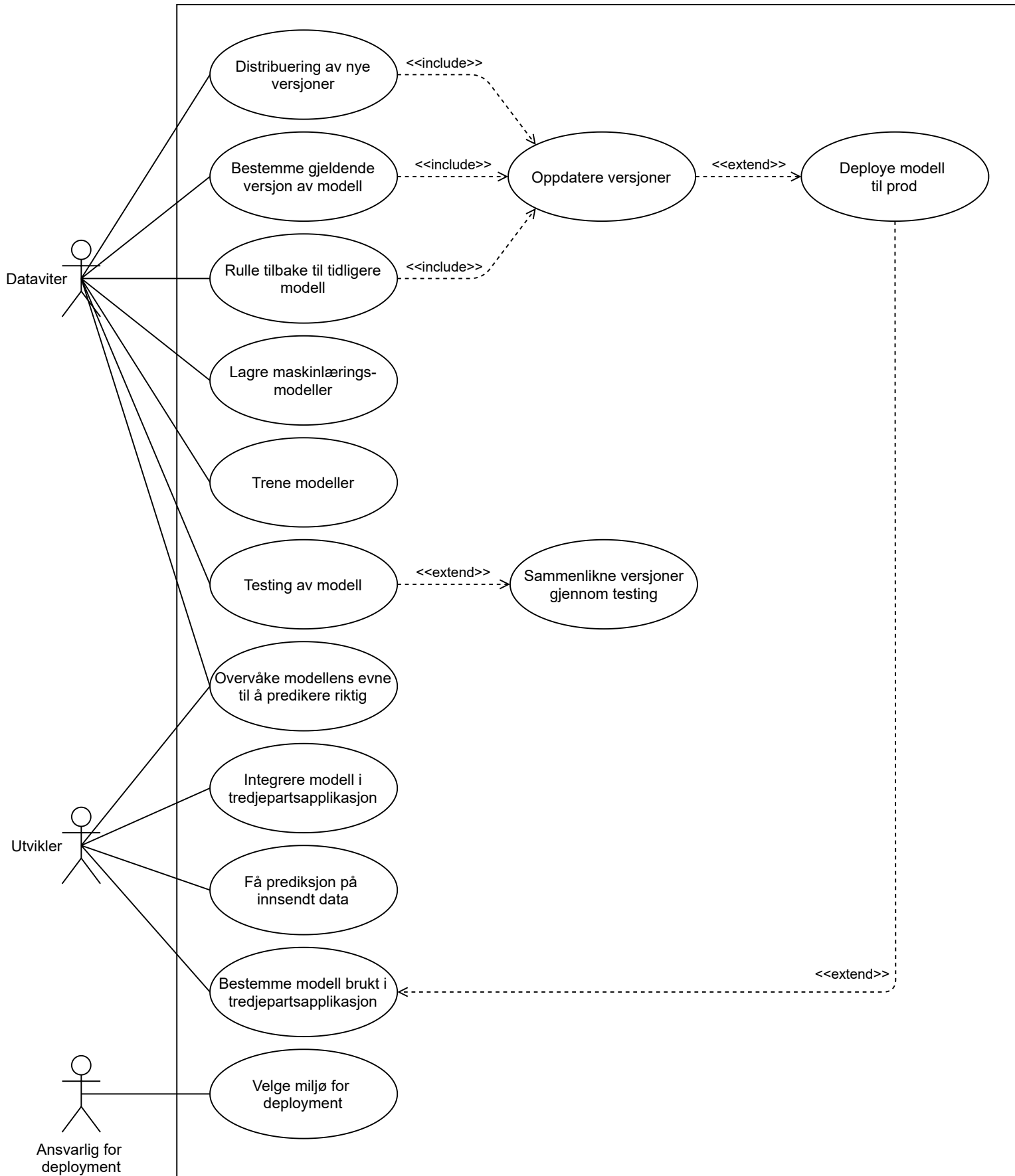
Om prosjektet må begrenses, sløyfes funksjonalitet nedenfra og opp.

3.5 Produktets funksjonelle egenskaper

En rekke verktøy må anvendes for å gjennomføre prosjektet. Det skal i hovedsak utarbeides i GCP, og prosjektgruppen vil da unytte denne plattformens funksjonalitet. De funksjonelle egenskapene til produktet kommer av punktene i kapitlet *Videre begrensning av omfang*. Produktet må altså kunne motta maskinlæringsmodeller, inndata og utdata. For å få til denne funksjonaliteten, skal forskjellige komponenter i GCP settes sammen. Dette beskrives grundigere i designrapporten, men i all hovedsak skal GCP kunne dekke all funksjonaliteten som behøves for å kunne utvikle produktet. GCP har blant annet, gjennom verktøyet *AI Platform*, innebygget funksjonalitet

for å deploye maskinlæringsmodeller, trene dem og gjøre prediksjoner. Disse oppgavene kan automatiseres ved hjelp av verktøyene *Cloud Functions* og *Pub/Sub*. Andre aktuelle verktøy i GCP er *BigQuery* og *Kubeflow*.

3.5.0.1 Use-case-diagram



3.5.0.2 Forklaring av use-caser

- **Distribuering av nye versjoner**

Dataviter skal ha metoder for å distribuere og oppdatere versjoner av modeller.

- **Bestemme gjeldende versjon av modell**

Dataviter skal kunne bestemme hvilken modell som er gjeldende i det gitte prosjektet. Dette er den modellen som skal trenes og testes.

- **Rulle tilbake til tidligere modell**

Dataviter skal kunne rulle tilbake til en tidligere versjon av modellen. Dette er i tilfeller der det skulle være feil med den nye modellen.

- **Oppdatere versjoner**

De tre use-casene over er alle avhengig av et versjonsstyringssystem. Dette systemet må ha kontroll på hvilke versjoner som eksisterer av hvilke modeller, hvordan disse er trent slik at man kan reprodusere treningen, holde styr på hvilke modeller som trenes, og hvilke som er i produksjon.

- **Produksjonssette ny modell**

Dataviteren skal deploye trente modeller til produksjon. Når en modell er deployet til produksjon, er den tilgjengelig for andre gjennom API-kall.

- **Lagre maskinlæringsmodeller**

Dataviteren skal lagre nye maskinlæringsmodeller. Modeller skal være adskilt fra hverandre slik at ikke modeller som gjør forskjellige oppgaver blandes sammen. Modellene har adskilte datasett som heller ikke skal blandes på tvers av modeller.

- **Trene modeller**

Dataviter skal kunne trene modellene som ligger i pipelinen. Trente modeller lagres som nye versjoner og med en forklaring på hva de er trent på og hvilken versjon de er trent fra.

- **Testing av modell**

Modellene skal kunne testes på testdata for å undersøke hvordan de presterer.

- **Sammenlikne versjoner gjennom testing**

Dataviteren skal kunne kjøre to modeller på det samme testdatasettet for å sammenligne de to og se hvor godt de gjør det i forhold til hverandre på de samme dataene.

- **Overvåke modellens evne til å predikere riktig**

Dataviter og utvikler skal kunne se modellenes nøyaktighet. For utvikler er dette spesielt relevant når vedkommende skal velge hvilken modell som skal brukes i applikasjonen.

- **Integrere modell i tredjepartsapplikasjon**

Utvikler skal kunne bruke prediksjoner fra en maskinlæringsmodell i sin applikasjon. Dette kan skje gjennom et API-kall i programkoden til utvikleren.

- **Få prediksjon på innsendt data**

Utvikler skal kunne sende inn mengder data til modellen, og få prediksjoner på denne.

- **Bestemme modell brukt i tredjepartsapplikasjon**

Utvikler skal ha muligheten til å selv bestemme hvilken modell som skal brukes i vedkommendes applikasjon. De tilgjengelige modellene er da de som datautvikleren har deployet til produksjon. Det skal være valgfritt for utvikleren å velge modell selv. Det skal være mulighet for å automatisk velge den nyeste modellen som dataviteren har deployet.

- **Velge miljø for deployment**

Ansvarlig for deployment skal bestemme hvilket miljø/prosjekt modellen skal deployes til.

3.5.1 Ikke-funksjonelle egenskaper og krav

Det er en rekke krav til funksjonalitet som ikke direkte passer til en use-case.

- **Deploying av modeller**

Når en ny modell skal deployes, skal man være trygg på det man deployer. Dette innebærer å være trygg på at man deployer noe som fungerer. Man skal sikre seg at man deployer noe som har den funksjonaliteten som forventes slik at det ikke skapes problemer for

brukeren av modellen, slik som utvikleren. Man må også være trygg på at modellen har god nok ytelse til det den skal brukes til.

- **Integrering med tredjepartsapplikasjoner**

Utvikleren av tredjepartsapplikasjoner skal slippe å forholde seg til maskinlæringsmodellen utover å implementere API-kallet. Nye modeller må ikke endre på formatet på svaret til API-kallet. Utvikleren må ikke endre koden sin til stadighet for å tilpasse seg modellene.

- **Versjonskontroll**

Versjoner av modeller bør lagres på en hensiktsmessig måte, og versjoneringen av modeller bør være logisk.

- **Mating av data**

Treningsdata og testdata til modellene må lagres på en hensiktsmessig måte. Treningsdataen kan gjerne skilles på eksempelvis størrelse, og testdataen kan for eksempel skilles på spesielle forhold/scenarier.

3.6 Milepæler og hovedaktiviteter

Følgende del av dokumentet beskriver hvordan prosjektgruppen forventer å disponere tiden gjennom prosjektperioden. Dette omfatter en oversikt over milepæler, som gir en overordnet oversikt over hva som må dokumenteres, og til hvilken tid dette forventes gjennomført. Milepælene har blitt satt med utgangspunkt i hovedaktivitetene, som består av forstudierapport, designrapport, driftsrapport og sluttrapport. Det utarbeides i tillegg en prosjekthåndbok fortløpende. Se tabell for detaljert oversikt over milepæler og tidsbruk, samt figur 4 for en grafisk fremstilling av stipulert arbeidsprosess.

Planlagte milepæler for prosjektet er:

Milepæler	Start	Slutt
Forstudierapport	11.01.2021	01.02.2021
Designrapport	01.02.2021	28.02.2021
Driftsrapport	01.03.2021	09.05.2021
Sluttrapport	10.05.2021	16.05.2021
Gruppevurdering	17.05.2021	20.05.2021
Innlevering	20.05.2021	20.05.2021

Figur 3: Milepæler for prosjektet

Uke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Forstudie																				
Systemkrav-/designrapport																				
Driftsdokument/driftsrapport																				
Sluttrapport																				
Vurdering av gruppesamarbeid																				
Presentasjon																				

Figur 4: Stipulert arbeidsprosess

For en mer detaljert oversikt, se Gantt-diagram i prosjekthåndboken.²

Tilgjengelighet: Prosjektgruppen tar påskeferie deler av uke 13, og vil da være mindre tilgjengelige enn vanlig. Det samme gjelder dagene som leder opp til eksamen 11. mai. Dette gjenspeiles ikke i Gantt-diagram eller stipulert arbeidsprosess.

4 Interessenter og rammebetingelser

I ethvert prosjekt er det nødvendig å kartlegge interessenter for å tydeliggjøre de enkelte interessentene, hvordan de påvirker prosjektet og hvordan de selv blir påvirket av prosjektet og/eller dets resultater. Samtidig stadfestes interessentenes forventninger og i hvilken grad interessentene selv kan forventes å bidra i prosjektet. Rammebetingelser beskriver absolutte krav som stilles til prosjektets gjennomføring, som vil ha avgjørende innflytelse på planer og valg i prosjektet. Dermed er det, ved å supplere en interessentanalyse med konkrete rammebetingelser, enklere å planlegge videre gjennomføring av prosjektet.

4.1 Interessentanalyse

Interessentene i dette prosjektet er personer, grupper eller organisasjoner som kan påvirke eller vil bli påvirket av prosjektets gjennomføring og/eller resultat. Statens vegvesen v/ produkteier Lars Meisingseth er oppgavestiller, og vil gjennom prosjektet ha kontinuerlig kommunikasjon med prosjektgruppen. Dette prosjektet ønsker Statens vegvesen å gjennomføre i fremtiden, så et Proof of Concept viser mulige løsninger. De stiller med en intern veileder (Torgeir Thoresen), som vil bidra med teknisk veiledning og fungere som sparringspartner. Samtidig stiller de med en intern sensor (Safurudin Mahic), som vil bidra med sensur av bacheloroppgaven. Ut over dette bidrar Statens vegvesen med nødvendige lisenser og programvare. Faglig veileder, Jostein Lund v/NTNU, er prosjektgruppens faglige kontakt. Veileder skal svare på spørsmål tilknyttet rapportering og prosjektgjennomføring, samt fungere som sensor ved vurdering av bacheloroppgaven. Prosjektgruppen, bestående av Jon Langholm, Ingvild Andersen og Erik Flæsen Dalen, er ansvarlige for prosjektet. Herunder faller utvikling, rapportskriving, kommunikasjon og dokumentering. Utviklere er her definert til å være brukere av selve prediksjonsdataen Statens vegvesen presenterer. Dette vil typisk være organisasjoner som ønsker å benytte tjenesten(e) Statens vegvesen leverer, eller Statens vegvesen selv, til å utvikle tredjepartsapplikasjoner eller andre applikasjoner som nytter prediksjonsdata.

Under følger en analyse av aktuelle interessenter, med fokus på hva interessenten ønsker av prosjektet, hvilken innflytelse interessenten har på prosjektet med tanke på prosess og/eller produkt, hva interessenten forventes å bidra med og hvordan hindringer kan reduseres.

Interessent	Interesse i prosjektet	Innflytelse (H,M,L)	Bidrag til prosjektet	Strategi for å redusere hindringer
Statens vegvesen	Fungerende pipeline som kan brukes i SVV, teknologisk utvikling	H (Høy)	Teknisk kompetanse, løsnings-orientering, veiledning, sensur, beslutninger	Lav-terskel kommunikasjon på Slack, Stand-up-møter ca to ganger i uken, møter hver 2.-3. uke
Faglig veileder (NTNU)	Høy måloppnåelse for studentene	H (Høy)	Prosess-veiledning, vurdering, veiledning	Møter hver 2.-3. uke
Prosjekt-gruppen	Lære om tema, høy måloppnåelse, vellykket produkt	H (Høy)	Ansvar	Daglig kommunikasjon og samarbeid, gruppekontrakt, god planlegging
Utviklere	Tilgang til SVVs ML-modeller	M (Middels)	Presentere funksjonelle ønsker	Løpende kommunikasjon

Figur 5: Interessenter

4.2 Rammebetingelser

Rammebetingelser er definert som absolutte krav til prosjektet, som har direkte og avgjørende innflytelse på planer og valg i prosessen. Betingelsene er som følger:

- Google Cloud Platform benyttes til lagring, trening og analyse av maskinlæringsmodeller
- GCP-miljø settes opp av Statens vegvesen

- Kostnader tilknyttet Google Cloud Platform har en myk grense på \$500 per måned per prosjekt
- Studentene skal jobbe 500 timer +/- 5% hver, som gir mellom 1425 og 1575 timer totalt
- Prosjektstart 5. januar 2021
- Prosjektet skal være ferdigstilt og all dokumentasjon levert innen 20. mai 2021
- Prosjektgruppens spisskompetanse er relativt lav i utgangspunktet

5 Kritiske suksessfaktorer

De kritiske suksessfaktorene er faktorer der et gunstig utfall er avgjørende for at prosjektet skal kunne anses som vellykket for alle interessenter. De rettleider arbeidsprosessen, og setter fokus på kritisk viktige forhold hos både oppdragsgiver og oppdragstaker.

5.1 Suksessfaktorer

Suksessfaktor	Kommentar
Aktiv involvering og tilrettelegging fra SVV	Nødvendig for å sikre at SVV får et produkt som stemmer overens med forventningene deres, samt tilrettelegging med tanke på nødvendige tilganger/programvare/kunnskap
Tilgjengelige veiledere	Nødvendig for prosjektgruppen å få tilbakemelding underveis
Tilgjengelighet på GCP	Nødvendig at GCP har stabil opptid for at arbeid skal kunne utføres
Bruke GCP-komponenter	Da GCP er plattformen SVV bruker må prosjektet nødvendigvis basere seg på GCP-komponenter
System som kan implementeres i produksjon	Nødvendig for å svare på oppgaven som stilles
God dokumentasjon underveis	Nødvendig for at teknikere kan reprodusere arbeidet
Deltakerne tilegner seg tilstrekkelig kunnskap underveis	Nødvendig for å greie å løse oppgaven
Regelmessig informasjonsutveksling	Nødvendig for å holde kontroll på prosessen og at ting blir gjort

Figur 6: Suksessfaktorer

5.2 Informasjonsbehov

For å holde interessentene oppdaterte og å ivareta deres behov og krav er det nødvendig å opprettholde god og kontinuerlig informering. Det er derfor viktig at formidlingen av informasjon foregår på en avtalt og forventet måte. Videre er det analysert hvilken informasjon som er nødvendig, til hvem, hvorfor, hvor og når.

Informasjon	Interessent	Formål	Kanal	Tid
Progresjon	SVV	Sikre nødvendig fremdrift	Møter med tilhørende referat, stand-up	Møter hver 2.-3. uke, stand-up to ganger i uken
Tekniske problemer, -utfordringer	SVV	Løse problemer	Slack, stand-up evt. på videomøte	Når problem oppstår
Forstudie	SVV, veileder	Sikre rammer, krav, mål	Teams	Første versjon 01.02.21, sporadisk etter dette
Design-rapport	SVV, veileder	Fastsette design og tekniske spesi-fikasjoner	Teams	Første versjon 22.02.21, sporadisk etter dette
Driftsrapport	SVV, veileder	Dokumentere oppsett av system	Teams	Første versjon 03.05.2021, sporadisk etter dette
Sluttrapport	SVV, veileder	Evaluering av prosjektet	Teams	10.05.21
Prosjekt-håndbok	Veileder	Vise prosess	Teams	Kontinuerlig
Gruppe-dynamikk	Veileder	Ta tak i eventuelle problemer	E-post	Når/hvis nødvendig
Alle rapporter	Veileder, sensor	Vurderings-grunnlag	Inspera	20.05.21
Nyttig lesestoff	Prosjekt-gruppe	Relevant kunnskap	Slack	Sporadisk

Figur 7: Informasjonsbehov

6 Risikoanalyse

En risikoanalyse identifiserer kritiske hendelser, og rangerer disse etter konsekvens og sjansen for at de inntreffer. Hensikten er å redusere risikoen for uønskede hendelser som kan skade prosjektprosessen. Hendelser med spesielt høy risiko behøver strategier for å håndteres. Ofte kan bevissthet rundt hendelsene i seg selv være preventivt, der man kan ta tak i faresignaler for å forebygge risikoen. En godt gjennomført risikoanalyse gir også prosjektmedlemmene en større trygghet rundt prosjektet. Identifikasjon av de mest kritiske hendelsene kommer senere i dette kapittelet.

Analyse av risikoene vises i tabellen under, med sannsynlighetene «Lav», «Moderat», og «Høy», og viktighetene «Ubetydelig», «Tolererbar», «Serøs» og «Katastrofal»

Kode	Risiko	Sannsynlighet	Viktighet
1	Oppgaven har for stort omfang	Moderat	Tolererbar
2	Sykdom i kritiske perioder	Moderat	Serøs
3	Produktet fungerer ikke til demo	Lav	Katastrofal
4	Konflikt i gruppa	Lav	Serøs
5	Problemer med teknologien	Moderat	Serøs
6	Oppgaver tar lengre tid enn forventet	Høy	Serøs

Figur 8: Risikoer

De uønskede hendelsene plasseres i en risikotabell for å markere deres alvorlighetsgrad. Hendelsene som havner i grønne celler er forholdsvis milde da de gjerne har lav sannsynlighet og/eller er av liten konsekvens. Disse er det i all hovedsak ikke nødvendig å redusere. Dette betyr derimot ikke at de ikke kan oppstå eller at man ikke bør ha strategier for å håndtere dem.

Hendelsene som plasseres i gule celler har moderat risiko. Disse hendelsene er av større betydning for prosjektet, og kan føre til større problemer om de

finner sted. Disse hendelsene bør ha konkrete strategier for å håndtere.

Hendelsene som plasseres i røde celler er de mest kritiske, og de må reduseres så mye som mulig. Disse hendelsene har både høy sannsynlighet for å inntreffe, og høy konsekvens om dette skjer.

	Tolererbar	Seriøs	Katastrofal
Lav		4	3
Moderat	1	2, 5	
Høy		6	

Figur 9: Risikoer i risikotabell

Med bakgrunn i disse uønskede hendelsene, har det blitt utarbeidet strategier for å håndtere dem. Disse strategiene presenteres i den underliggende tabellen.

Kode	Risiko	Strategi
1	Oppgaven har for stort omfang	Redusere oppgavens omfang. Prioritere å gjennomføre en mindre del av oppgaven. Dette blir eventuelt diskutert på møte med veileder.
2	Sykdom i kritiske perioder	Fordele arbeidsoppgavene på resten av gruppa. Ved vedvarende helseproblemer må det vurderes å redusere omfanget på oppgaven slik at de gjenværende friske prosjektmedlemmene klarer å gjennomføre prosjektet på egenhånd.
3	Produktet fungerer ikke til demonstrasjon	Finne ut hva som gikk galt og nøste opp i problemet. Forhindre at problemet gjentar seg. Hyppige utrullinger av små endringer gjør at man blir tryggere på endringene og lettere kan finne feilen.
4	Konflikt i gruppa	Bruke gruppekontrakten til å håndtere konflikten.
5	Problemer med teknologien	Lese seg opp, kontakte veileder, be om teknisk hjelp fra Vegvesenet. Eventuelt diskutere bytte av teknologi.
6	Oppgaver tar lengre tid enn forventet	Sette flere folk på oppgaven, eventuelt omprioritere. Se strategi for å håndtere risiko 1.

Figur 10: Strategier for å håndtere risikomomenter

7 Kost/nytte-analyse

Kost/nytte-analysen gjennomføres for å presentere nytten av prosjektet, satt opp mot kostnadene. Dette gir oppdragsgiver et mer omfattende grunnlag for å kunne beslutte om prosjektet bør gjennomføres eller ikke. En slik analyse ser i første omgang på hvilken effekt prosjektet vil oppnå, også betegnet som *forventet nytte*. Videre går den inn på hva prosjektet vil koste, for så at dette settes opp mot den forventede nytten. Det skilles også mellom *kvantifiserbar* og *ikke-kvantifiserbar* nytte. Den kvantifiserbare nytten er konkret og kan tallfestes, der den ikke-kvantifiserbare nytten er mer abstrakt og vanskelig å tallfeste. Analysen setter altså løsningen slik den er i dag, også kalt *0-alternativet*, opp mot den situasjonen prosjektet resulterer i.

Dette prosjektet er et moderniserings- og digitaliseringprosjekt. Bruk av kunstig intelligens er også noe som egendefineres hos hver enkelt bedrift. Dette fører til at det er vanskelig å tallfeste den kvantifiserbare nytten. Av den grunn vil en kost/nytte-analyse gi begrenset informasjon til oppdragsgiver, sammenlignet med samme analyse i et mer kostnadsrettet prosjekt. Da prosjektet går ut på å utvikle et 'Proof of Concept' tilfører det ikke bedriften en direkte nytte, annet enn å undersøke om et større prosjekt har livets rett. Den indirekte nytten, eksempelvis en mer automatisert veianalyse, er avhengig av det større prosjektet, og av den grunn blir det ikke medregnet. Når det kommer til kostnader, så er det usikkert hva et 'Proof of Concept' vil koste i denne konteksten. Maskinlæringsmodellene utvikles utenfor dette prosjektet, og blir ikke medregnet i analysen. Google Cloud Platform benyttes allerede av Statens vegvesen, og inngår i kostnadstaket. Python-programpakker støtter opp Google Cloud Platform i å utvikle produksjonspipelinene har i hovedsak åpen kildekode, og derav minimale kostnader. Prosjektgruppen tar derfor utgangspunkt i kostnadstaket som oppdragsgiver har satt. På bakgrunn av denne begrensede informasjonen, blir det derfor presentert en grovkissert kost/nytte-analyse med redusert omfang.

7.1 Kvantifiserbar og ikke-kvantifiserbar nytte

7.1.1 Kvantifiserbar nytte

Prosjektet er som beskrevet innledningsvis i kapittel 7 et moderniserings- og digitaliseringsprosjekt, fremfor et kostnadsprosjekt. Bedriftens effektmål støtter opp under dette, og består i større grad av ikke-kvantifiserbare mål enn tallfestede. Av denne grunn ble det besluttet å utelate den kvantifiserbare nytten.

7.1.2 Ikke-kvantifiserbar nytte

Da løsningen som utarbeides ikke skal direkte implementeres, men heller bevise en mulighet for implementasjon, skilles det mellom direkte og indirekte ikke-kvantifiserbar nytte. Den direkte nytten gjenspeiler de positive effektene bedriften kan innhente på kort sikt, og som et direkte resultat av prosjektet. Om prosjektet blir vellykket, kan det tas i bruk i Sagaprojektet. En vellykket implementasjon i et større prosjekt legger til rette for den indirekte nytten. Den kommer ikke som direkte følge av dette prosjektet, men forteller hvordan bedriften kan ta nytte av innføringen av maskinlæringsmodeller i sin helhet og på lang sikt.

7.1.2.1 Direkte nytte

- **Bevise at et større prosjekt er teknologisk levedyktig.** Oppgaven fra bedriften er å utarbeide et 'Proof of Concept' for å foreslå bevis for at idéen er levedyktig. Teknologiprojekter er svært kostbare, og igangsettelse av slike prosjekter uten et godt grunnlag ender ofte opp som lite vellykket. Et 'Proof of Concept' tilfører altså bedriften nytte ved å foreslå bevis for at et større prosjekt er teknologisk levedyktig.
- **Tilføre bedriften innsikt i tilgjengelige verktøy for prosessering og vedlikehold av maskinlæringsmodeller** . Google Cloud Platform inneholder utallige verktøy for å administrere, prosessere og overvåke maskinlæringsmodeller. Gjennom prosessen vil prosjektgruppen filtrere ut og ta i bruk de mest nyttige verktøyene for produksjonspipelinen, noe bedriften også kan lære av og få innsikt fra i ettertid.

7.1.2.2 Indirekte nytte

- **Modernisere og effektivisere veianalyse.** Prosjektet setter ikke søkelys på en spesifikk maskinlæringsmodell, men det kan tenkes at flere av modellene som tas i bruk av Vegvesenet i fremtiden vil kunne modernisere og effektivisere veianalyse. Dette kan være snakk om maskinlæringsmodeller som analyserer skilt, trafikk, føre, med mer. Om dette prosjektet lykkes, bidrar det til å støtte opp under det større prosjektet som kan føre til dette.
- **Modernisere arbeidsplassen** . Statens vegvesen er stort og består av både kontorarbeidere, veiarbeidere, med mer. Effektive maskinlæringsmodeller vil kunne modernisere og automatisere flere typer arbeidsoppgaver, da spesielt de praktiske. Et eksempel er en effektivisering av vedlikehold av vei. Dette kan videre føre til et redusert behov for arbeidskraft, noe som sparer bedriften for vesentlige ressurser.
- **Sikre arbeidskvalitet hos innleide veientreprenører** . Bedriften leier årlig inn et stort antall entreprenører for å utføre diverse arbeidsoppgaver, da spesielt på vinterstid. Dette omfatter blant annet strøarbeid og måking av snø på veiene. Maskinlæringsmodeller har potensialet til å kunne kontrollere arbeidsmengden hos disse entreprenørene, for å sikre at deres arbeidsoppgaver tilsvarer det som er forventet av daværende vær og veiforhold.

7.2 Bortfall av direkte kostnader

De direkte kostnadene inngår i prosjektets nytteverdi, og omfatter blant annet hva bedriften kan spare inn av kostnader på å innføre det nye systemet. Systemet som skal innføres har verken et eksisterende rammeverk eller programvare, og har derav ingen kostnader som direkte kan økes eller reduseres. Skal man se på bortfall av direkte kostnader må man tenke svært langsiktig, noe som kan oppfattes vagt. Da blir det snakk om bortfall på bakgrunn av en vellykket implementasjon i systemet, og fordi prosjektet ikke retter seg mot en spesifikk maskinlæringsmodell, kan dette omfatte svært

mye. På sikt kan en vellykket implementasjon føre til redusert behov for manuell arbeidskraft, som fører til et bortfall av direkte kostnader. Da dette ikke blir en direkte konsekvens av nåværende prosjektprosess, blir det ikke tatt hensyn til i denne omgangen.

7.3 Estimerte kostnader

Prosjektprosessen har ingen innlysende kostnader, annet enn kostnader tilknyttet ressursbruk i Google Cloud Platform og operasjonelle kostnader senere i forløpet. Da skyløsningen allerede eksisterer i bedriften, er det vanskelig å estimere en eksakt økning i ressurskostnader på bakgrunn av denne prosjektprosessen. Kostnadene er todelte, og man må også beregne arbeidskostnader. Det kan være aktuelt å øke kostnadene på digitale verktøy, om dette kan redusere behovet for manuell arbeidskraft, men det er for tidlig i prosessen til å kunne tallfestes. Av den grunn er det besluttet å ta utgangspunkt i kostnadstaket som ble satt av Statens vegvesen i forkant av forstudien. Prosjektet har et mykt tak på \$500 i måneden, dette tilsvarer per januar 2021 sirka 4500 NOK. Dette taket er satt midlertidig for å forhindre store kostnader som følge av uforutsett høy ressursbruk, og kan heves om det skulle være behov.

7.4 Sammenstilling av kost/nytte

Denne delen skal presentere en sammenstilling av verdiene som man kom frem til i kost/nytte-analysen. Da denne forstudien foreløpig ikke kan vise til konkrete verdier innen verken kostnad eller nytte, utelates denne sammenstillingen foreløpig. Dukker det opp en mulighet for tallfesting av kost og/eller nytte senere i prosjektprosessen kan dette avsnittet revideres.

8 Retningslinjer og standarder

8.1 Krav til dokumentasjon

Dokumentasjon som skal produseres i løpet av prosjektet, samt dato for ferdigstilling, rutiner for godkjenning og revisjon. Alle i prosjektgruppen er kvalitetsansvarlige, og vil lese igjennom all dokumentasjon uavhengig av hverandre.

- **Forstudierapport:** Undersøker om prosjektet kan gjennomføres innen gitte ressursrammer. Ferdigstilles innen 08.02.2021.
- **Designrapport:** Kvalitetssikrer prosjektprosessen, og viser både design og de tekniske spesifikasjonene som behøves for å få et vellykket resultat. Ferdigstilles innen 22.02.2021.
- **Driftsrapport:** Den endelige prosjektløsningen dokumenteres i denne rapporten. Dette omfatter blant annet teknologivalg og oppsett av løsning. Ferdigstilles innen 03.05.2021.
- **Sluttrapport:** En siste rapport som evaluerer prosjektarbeidet og prosessmålene som ble satt i forstudien. Den skal også gi en oppsummering og oversikt over prosjektet og prosjektresultatet. Ferdigstilles innen 10.05.2021.
- **Prosjekthåndbok:** Utarbeides parallelt med de overstående rapportene, og inneholder timelister, ukesrapporter og oversikt over møtevirksomhet. Ferdigstilles innen 20.05.2021.
- **Vurdering av gruppesamarbeid:** En individuell vurdering av hvor godt prosjektgruppa har samarbeidet. Ferdigstilles innen 20.05.2021.

8.2 Krav til kvalitetsgjennomganger

Dokumentene skal kontrolleres og revideres av både prosjektdeltakerne og de øvrige interessentene, dette gjelder spesielt de større dokumentene som forstudierapport, designrapport, driftsrapport og sluttrapport. Dokumentasjon fra møtevirksomhet er mindre formelt og behøver ikke samme nivå av kvalitetskontroll. Det stilles også større språklig krav til

hoveddokumentasjonen. Alle interessenter vil likevel få tilbud om å revidere og gi tilbakemelding på all dokumentasjon. For øvrig vil rapporter, tidsbruk og fremdrift rapporteres til, og vurderes av, intern sensor og veileder Jostein Lund, i samarbeid med ekstern veileder Torgeir Thoresen.

Endelig resultat vil vurderes og karaktersettes av Jostein Lund fra NTNU, sammen med ekstern sensor Safurudin Mahic fra Bekk på vegne av Statens vegvesen. Vurderingen stilles på bakgrunn av dokumentkvalitet og prosjektløsning. Det vektlegges også at all dokumentasjon blir levert i henhold til tidsfristen 20.05.2021, og at prosjektperioden avsluttes med en godt gjennomført presentasjon.

8.3 Krav til standarder og metoder

Prosjektgruppen skal bruke utvalgte verktøy i løpet av prosjektperioden:

- All formell dokumentasjon og rapportskriving skal skje i \LaTeX , i samskrivingsprogrammet Overleaf.
- All lagring av formell prosjektrelatert dokumentasjon og fagstoff skjer i felles Microsoft Teams-gruppe, der alle interessenter har tilgang. Her foregår også oppfølging av prosjektprosessen.
- All uformell dokumentasjon og kladder utarbeides i en felles Microsoft OneNote.
- Møteinnkalling skal skje via kalenderfunksjon i Outlook/Teams, da dette plasserer møtet direkte i mottakers kalender.

8.4 Endringshåndtering

Endringsønsker kan komme på vegne av bedrift, prosjektdeltakerne, brukere eller andre interessenter, og om det oppstår behov for endring skal dette diskuteres med og godkjennes av bedrift snarest. Endringsønsker skal behandles formelt og forretningsmessig, dette gjelder større endringer som prosjektets omfang og teknologivalg.

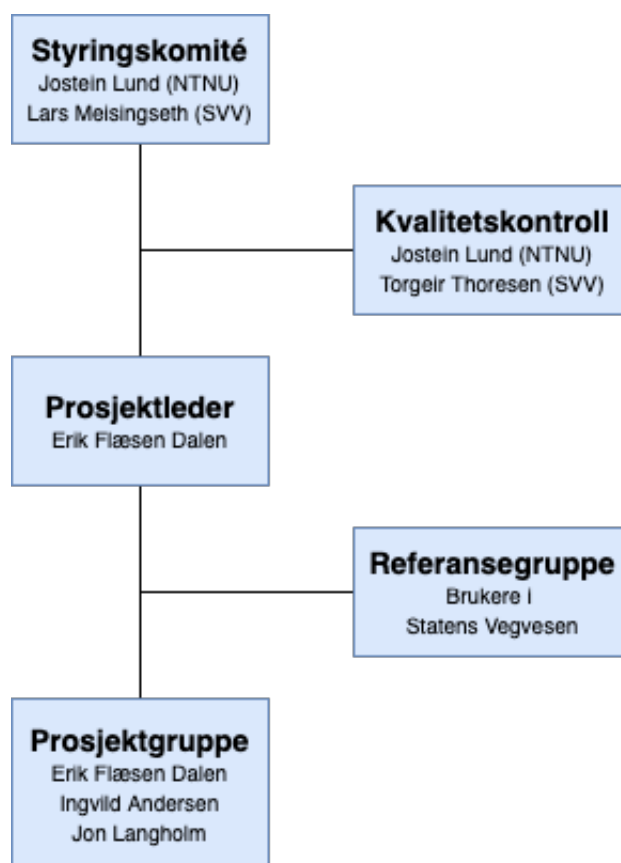
Framgangsmåten for håndtering av endringsønsker er:

1. Dokumenter endringens innhold
2. Analyser konsekvensene for prosjektet
3. Beregn eventuell kost/nytte
4. Godkjenning og aksept
5. Logg endringen
6. Juster planene
7. Informer interessentene
8. Gjennomfør endringen

Underveis i programvareutviklingen følger prosjektet Scrum for endringshåndtering, og ikke den overstående listen.

9 Prosjektorganisering

Prosjektet følger en klassisk prosjektorganisering, bestående av prosjektleder, styringskomité, kvalitetskontroll, prosjektgruppe og referansegruppe. Følgende organisering er valgt da det tyder på å passe best i dette prosjektet.



Figur 11: Prosjektorganisering

Prosjektleder

Erik Flæsen Dalen er utnevnt av prosjektgruppen til å inneha den formelle prosjektlederrollen gjennom prosjektet. Prosjektleders ansvar er i dette tilfellet å legge til rette og sørge for god kommunikasjon mot prosjektets interessenter og øvrige deltakere, løpende kvalitetskontroll av dokumentering og løsninger, overholdelse av tidsfrister og prosesskontroll.

Styringskomité

Definert til å være prosjektets oppdragsgiver, støttet av faglig kontakt, består

styringskomitéen av Lars Meisingseth v/Statens vegvesen og Jostein Lund v/NTNU. Lars Meisingseth fungerer som prosjektets eier og har beslutningsmyndighet for prosjektet.

Kvalitetskontroll

Som et bindeledd mellom prosjektleder og styringskomite fungerer kvalitetskontroll som kontrollør av prosjektets fremdrift, i form av dokumentering og prosess, sett fra NTNUs side. I tillegg skal også kvalitet på utført arbeid kontrolleres for å sørge for best mulig måloppnåelse.

Referansegruppe

Representanter for ansatte i Statens vegvesen, i hovedsak, men ikke begrenset til, Safurudin Mahic og Torgeir Thoresen, innehar rollen som referansegruppe. De representerer brukerne av systemet, og vil inneha en liknende rolle som kvalitetskontroll, men i dette tilfellet fra oppdragsgivers perspektiv. Teknisk kvalitetskontroll og beslutningsstøtte er typiske oppgaver.

Prosjektgruppe

Prosjektgruppen inneholder prosjektets utførende, nevnelig Erik Flæsen Dalen, Ingvild Andersen og Jon Langholm. Deres ansvar er å sørge for at prosjektet leverer et kvalitetsfylt resultat og dekker og oppnår de fastsatte krav og mål. Prosjektgruppen går inn i prosjektet med flytende arbeidsoppgaver, men forhåndsdefinerte fokusområder. Erik tiltrer rollen som prosjektleder, Ingvild fokuserer på dokumentering og rapporter, mens Jon fokuserer på utvikling og tekniske løsninger. Deltakerne bidrar likevel i de forskjellige delene etter behov.

10 Anbefaling om videre arbeid

Statens vegvesen ønsker seg altså en maskinlæringspipeline som kan brukes til forskjellige prosjekter i fremtiden. Bachelorgruppen har i forstudiet undersøkt mulige løsninger og teknologier som kan nyttes for å skape et 'Proof of Concept'. Det var vanskelig å tallfeste kostnad og nytteverdi, men den potensielle nytteverdien Vegvesenet kan hente i det lange løp overskrider det man antar ender opp som kostnad. Det er derfor vanskelig å argumentere for at prosjektet ikke bør gjennomføres. For deltakerne selv, vil kunnskapen og erfaringen som opparbeides under gjennomføringen av prosjektet være til høy verdi. Bachelorgruppen er høyst motiverte til å gå videre med prosjektet og dette anbefales på det sterkeste.

Referanser

[1] Regjeringen: *Statens vegvesen*, 2021.

Hentet fra:

<https://www.regjeringen.no/no/dep/sd/org/underliggende-etater/statens-vegvesen/id443412/>.

Lastet ned: 15. januar 2021

[2] J. A. Langholm, I. Andersen og E. F. Dalen, "Prosjekthåndbok", *NTNU*, Mai 2021. [Vedlegg].

Hentet fra: Vedlegg i innlevering av oppgave.

MLOps på Google Cloud Platform

Designrapport

Ingvild Andersen, Jon Akselberg Langholm, Erik Flæsen Dalen

19. mai 2021



Statens vegvesen

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
11.03.2021	1.0	Førsteutkast	Erik F.D., Ingvild A. og Jon L.
14.03.2021	1.1	Mindre revidering etter tilbakemeldinger fra veileder	Erik F.D., Ingvild A. og Jon A. L.
19.05.2021	1.2	Språkvask	Erik F.D., Ingvild A. og Jon A. L.

Innhold

1 Innledning	5
1.1 Dokumentets hensikt	5
1.2 Innhold	5
1.3 Avgrensning	6
1.4 Definisjoner og forkortelser	6
2 Kort om kunden og behov	7
3 Hvorfor valgt teknologi	8
4 Tekniske løsninger	9
4.1 Introduksjon til teknologi	10
4.1.1 MLOps	10
4.1.2 Plattform: Google Cloud Platform (GCP)	12
4.1.3 Rammeverk: TensorFlow Extended (TFX)	14
4.1.4 Orkestrering: Kubeflow Pipelines (KFP)	16
4.2 Verktøy og ressurser	18
4.2.1 Prosjekter i Google Cloud Platform	18
4.2.2 Utviklingsmiljø	20
4.2.3 Akseptanse- og testmiljø	20
4.2.4 Produksjonsmiljø	20
4.3 Detaljerte løsningsbeskrivelser	21
4.3.1 TensorFlow Extended (TFX)	21
4.3.2 Google Cloud Platform (GCP)	22
4.3.3 Kubeflow Pipelines (KFP)	23
4.3.4 Flyt i pipelinen	24
4.3.5 Forutsetninger og avhengigheter	26
4.3.6 Programvare og systemkrav	26
4.3.7 Oppdatert GANTT-diagram	26
5 Alternative flyter i GCP	27
Referanser	32

Figurer

2	Overordnet flyskjema for TFX-pipeline. Inspirert av TensorFlow-dokumentasjon. ⁵	16
3	Komponentspesifikk eksempelflyt i TFX-pipeline på GCP	16
4	Kubeflow Pipelines integrert med Google Cloud Platforms ML-verktøy	18
5	TFX-komponentene satt sammen i Kubeflow	24
6	Faktisk arbeidsprosess.	27
7	Planlagt arbeidsprosess.	27

1 Innledning

1.1 Dokumentets hensikt

Følgende dokument er en designrapport utarbeidet i sammenheng med bacheloroppgaven 'MLOps på Google Cloud Platform' på oppdrag fra Statens vegvesen. Dokumentets hensikt er å tilby leseren en konseptuell og tekstlig beskrivelse av løsningen oppgaven skal resultere i. Dokumentet er en del av prosjektprosessens kvalitetssikring, og omfatter derfor en innsikt i bedriftens krav og behov, samt en oversikt over foreslått løsningsdesign for å møte disse. Denne beskrivelsen vil sikre gjensidig kommunikasjon mellom oppdragsgiver og oppdragstaker, slik at alle parter sitter på samme mengde informasjon når dokumentet er ferdigstilt. Målet er at prosjektgruppen og Statens vegvesen skal bli samstemte i valg av løsningsdesign, tekniske detaljer og forventninger. Designrapporten bygger videre på beslutningene som ble tatt i forstudierapporten. Dokumentet begrenses til å beskrive forslag til løsning av prosjektoppgaven, men ikke hvordan man kan reproducere denne løsningen. En slik innføring vil finne sted i driftsrapporten.³⁴ Forslag til løsning vil også innebære en innføring i valgt teknologi og hvordan den samsvarer med oppgavekrav fra Statens vegvesen.

1.2 Innhold

Første kapittel i dokumentet beskriver dokumentets hensikt (1.1) og en oversikt over dets innhold (1.2). Videre presenteres en kort oversikt over dokumentet og oppgavens avgrensning (1.3). Definisjoner og forkortelser befinner seg i eksternt ordbok.³² Kapittel 2 går overordnet inn på kunden og behov, altså hvem kunden er og hva de ønsker ut av prosjektprosessen. Kapittel 3 og 4 går inn på bakgrunnen for valg av teknologi, samt en oversikt over disse. Helt konkret innebærer det overordnet innføring i MLOps (4.1.1), Google Cloud Platform (4.1.2), TensorFlow Extended (4.1.3) og Kubeflow Pipelines (4.1.4). Videre i kapittel 4 får leseren en innføring i prosjektets verktøy og ressurser, da spesielt en beskrivelse av prosjekter i Google Cloud Platform (4.2.1), samt en gjennomgang av utviklingsmiljø (4.2.2), akseptanse- og testmiljø (4.2.3) og produksjonsmiljø (4.2.4). I avsnitt 4.3 gjennomgås

detaljerte løsningsbeskrivelser, TFX (4.3.1), GCP (4.3.2), KFP (4.3.3), flyt i pipelinen (4.3.4) forutsetninger og avhengigheter (4.3.5), programvare og systemkrav (4.3.6) før oppdatert Gantt-diagram (4.3.7). Til slutt i rapporten finnes en overordnet oversikt over alternative flyter i GCP (5) før en liste over referanser og vedlegg.

1.3 Avgrensning

Som beskrevet i forstudierapporten²⁵ går prosjektets resultatmål ut på å ferdigstille et 'Proof of Concept' på en produksjonspipeline for en maskinlæringsmodell i Google Cloud Platform. Dette omfatter å kunne kjøre ut maskinlæringsmodeller i Google Cloud Platform, samt nye versjoner av disse. Det handler også om å kunne bruke maskinlæringsmodellene til å hente ut prediksjoner, overvåke deres prediksjonskvalitet, trening og skalering. Løsningen skal være kompatibel med Google Cloud Platform, men prosjektgruppen står fritt til å vurdere andelen selvskrevne kode opp i mot eksisterende grensesnitt i GCP. Det tas forbehold om at noe funksjonalitet kan sløyfes ettersom oppgavens omfang spesifiseres videre.

Prosjektgruppen skal ikke utvikle maskinlæringsmodellene, drifte eller implementere produksjonspipeline, drive opplæring, sette opp miljø i Google Cloud Platform, eller legge til funksjonalitet som ikke er etterspurt. Dette dokumentet skal presentere og begrunne et forslag til løsningsdesign, men omfatter ikke en gjennomgang av denne løsningen. Sistnevnte kommer i driftsrapport.³⁴ Det må også nevnes at løsningsforslaget ikke testes i sin helhet før arbeidet med driftsdokumentasjon, noe som kan føre til endringer i valgt løsning. Om dette skulle skje vil det kommuniseres tett med oppdragsgiver, og det vil beskrives i sluttrapporten.³⁵

1.4 Definisjoner og forkortelser

Det er utarbeidet en egen ordbok for prosjektet, som kan finnes i vedlagt ordbok.³²

2 Kort om kunden og behov

Oppdragsgiver er dataplattformen Saga, på vegne av Statens vegvesen. Organisasjonen har ansvaret for Norges riksveier, noe som innebærer forvaltning, utredning, planlegging, bygging, drift og vedlikehold av disse. Statens vegvesen har gjennom Saga en visjon om å gjøre organisasjonen mer datadrevet. Hensikten er å bruke data til å oppdage nye sammenhenger og muligheter, ta mer faktabaserte beslutninger, samt utnytte organisasjonens store datamengder på en mer hensiktsfull måte. Dagens løsning preges av uheldige formater og lite tilfredsstillende API-er. Deling av data og tversanalyser vektlegges også i liten grad.¹

Saga ønsker at prosjektgruppen skal utforske og utvikle et 'Proof of Concept', i form av en produksjonspipeline som kjører i Google Cloud Platform. Denne løsningen skal kunne ta inn og bytte ut maskinlæringsmodeller, mate disse med ny inndata, gjennomføre trening, være skalerbar, hente ut prediksjoner, samt overvåke kvaliteten på disse. Produksjonspipelinene skal være mest mulig automatisert, og skal kreve minst mulig interaksjon fra datavitere. En ønskesituasjon er at man kan sette maskinlæringsmodeller ut i produksjon, og at trening, testing, overvåking og serving av prediksjoner er tilnærmet selvgående. Dette vil dekke de behov SVV ønsker for å kunne ta i bruk dataen de sitter på så effektivt som mulig. Oppgaven skal løses ved bruk av Google Cloud Platform, og underliggende verktøy er til disposisjon. Prosjektgruppen står likevel fritt til å utforske andre alternativer, så lenge løsningen fungerer godt med plattformen. Oppdragsgiver er fortsatt i utforskningsfasen når det kommer til maskinlæring, og ønsker derfor at løsningen skal inneholde en sammenligning av løsninger, og en forklaring på hvorfor gjeldende produkt-teknologi-løsning ble valgt. Bakgrunnen for dette er at de vil finne teknologien som passer best for deres drift og bruk.

3 Hvorfor valgt teknologi

Valg av plattform ble som tidligere nevnt Google Cloud Platform. Dette valget ble innlysende da Statens vegvesen allerede bruker denne skyløsningen, og å samle alle tjenester på ett sted forenkler utvikling, drift og vedlikehold betraktelig. GCP er også en skyplattform som tilbyr flere verktøy for maskinlæring, noe som gir en stor mengde muligheter når man skal angripe maskinlæringsoppgaver. Plattformen er også langt framme når det kommer til maskinlæring, sammenlignet med lignende plattformer på markedet. Videre ble det besluttet å ta i bruk skalerbar teknologi som kunne integreres sømløst med GCP, og som spesielt kunne prosessere maskinlæringsmodeller basert på TensorFlow. Bakgrunnen for at prosjektgruppen har valgt å fokusere på kun maskinlæringsmodeller basert på TensorFlow, er at TensorFlow er det mest utbredte rammeverket for utvikling av maskinlæringsmodeller.³¹ Det ble besluttet å ta i bruk TensorFlow Extended (TFX) som rammeverk og Kubeflow Pipelines som orkestrator, da disse teknologiene er godt tilpasset slike modeller. TFX er de individuelle komponentene i pipelinen, og Kubeflow Pipelines passer på at disse kjøres i riktig rekkefølge. Disse tjenestene er også skalerbare ende-til-ende tjenester som kan utvides og egendefineres etter bedriftens ønske i senere tid. Blant annet kan Kubeflow Pipelines nyttes som orkestrator for maskinlæringsmodeller basert på andre teknologier enn TensorFlow.

Rammeverk og orkestrator er begge tjenester som er åpen for fri bruk og vil ikke påløpe kostnader ved bruk av disse. Pipelinen vil kjøre integrert med GCP, og det vil derav påløpe kostnader tilknyttet bruk av GCPs tjenester. Da prosjektet fortsatt er i en eksperimentell fase er det vanskelig å gi eksakte tall. For øvrig kan bedriften senere tilpasse selv hvor mye de vil integrere pipelinen med GCP. En tett integrasjon vil føre til høyere kostnader, men skyplattformen har også avanserte maskinlæringsverktøy som kan sikre kvalitet i trening, overvåking og levering av prediksjoner. Disse verktøyene er også svært avanserte, noe som gjør dem vanskelig å gjenskape på egen hånd utenfor plattformen.

4 Tekniske løsninger

Valg av løsning ble gjort på bakgrunn av krav fra Statens vegvesen og omfang av prosjektet. Det ble tidlig i planleggingen av løsningsdesign besluttet at teknologivalg skulle implementere prinsippene bak MLOps, som er maskinlæringens svar på DevOps.³² Bakgrunnen var at løsningen skulle være en automatisk trigget pipeline, der de som utvikler maskinlæringsmodeller, og de som har dem i produksjon, skal kunne samarbeide uten problemer. Dette gjelder helt fra modellene påbegynnes til de kjøres ut i produksjon og leverer prediksjoner. MLOps reflekterer dette ved å legge opp til automatisering hele veien fra man setter modeller ut i produksjon, til man leverer prediksjoner. Videre ble det besluttet å ta i bruk Google Cloud Platform for å lagre og kjøre løsningen og dens underliggende data, da dette er en skyplattform Statens vegvesen bruker i det daglige og er godt kjent med.

Når man skal opprette en automatisert maskinlæringspipeline behøver man ikke bare komponenter, men også et rammeverk og en orkestrator. Rammeverket konfigurerer pipelinen, og orkestratoren kontrollerer kjøringen av denne. Det vil si at orkestratoren styrer og dirigerer rekkefølgen på komponentene og arbeidet pipelinen består av. Det finnes flere rammeverk å velge i når man skal sette sammen maskinlæringspipeliner, og det ble i dette prosjektet besluttet å gå videre med TensorFlow Extended (TFX). TensorFlow er svært utbredt innen utvikling av modeller i dag, og det er derfor ønskelig med en pipeline som best mulig kan tilpasses disse. TFX er et ende-til-ende verktøy, og rammeverket støtter opp om alt fra førprosessering av data til trening og levering av produksjonsklare modeller. Rammeverket implementerer også prinsippene bak MLOps, og derav ønsket funksjonalitet og krav fra oppdragsgiver. Som orkestrator skal prosjektet ta i bruk Kubeflow Pipelines, som er en plattform for å bygge og kjøre ut portable og skalerbare maskinlæringspipeliner basert på Docker-konteinere. Løsningen tilbyr en ende-til-ende orkestrering som legger til rette for gjenbruk av kode, pipeliner og komponenter, noe som lønner seg for Statens vegvesen, som ønsker en mulighet for rask skalering.

4.1 Introduksjon til teknologi

Dette avsnittet skal overordnet beskrive de mest aktuelle verktøyene for å løse oppgavebeskrivelsen. Det er ikke en oversikt over valgt løsning, men heller en mulighet for leser å sette seg inn i de ulike verktøyenes funksjonalitet. Det er fordelaktig at leser kjenner til den viktigste funksjonaliteten som er aktuell, da det senere i rapporten skal argumenteres for valg av endelig løsning. Statens vegvesen er relativt ferske innen maskinlæring, og ønsker derfor en oversikt over mulige løsninger, i tillegg til en beskrivelse av og begrunnelse for valgt løsning. Sistnevnte gjennomgås i avsnittet detaljert løsningsbeskrivelse".

4.1.1 MLOps

Følgende informasjon som beskriver MLOps er i store deler basert på dokumentasjon fra Google Cloud Platform. Dette omfatter i hovedsak løsningsartiklene *Architecture for MLOps using TFX, Kubeflow Pipelines, and Cloud Build*⁵ og *MLOps: Continuous delivery and automation pipelines in machine learning*.²

MLOps og DevOps baserer seg begge på konseptene rundt kontinuerlig integrasjon (CI) og kontinuerlig levering (CD), men CI innen MLOps handler ikke kun om testing og validering av kode, men også av data, dataskjema og modeller. CD omfatter ikke kun levering av en enkelt programvarepakke, men hele systemer og pipeliner. MLOps presenterer også et nytt konsept kalt kontinuerlig trening (CT), som er unikt innen maskinlæringssystemer. Dette beskriver en automatisk og iterativ om-trening og levering av modeller. Metoden er mye mer omfattende enn DevOps innen testing, som omfatter datavalidering, samt en kontinuerlig evaluering av modellenes kvalitet. Utkjøring kan også være utfordrende, da modeller behøver en produksjonspipeline som trener og kjører ut nye versjoner. Dårlige ytelse i maskinlæringssystemer skyldes som regel endringer i dataprofiler, i motsetning til lite optimal kode hos programvaresystemer. Maskinlæringsmodeller må konstant overvåkes, og ha klare tiltak ved endringer i data eller en reduksjon i prediksjonskvalitet.²

MLOps handler altså om å forenkle samarbeidet mellom de som utvikler maskinlæringssystemer, og de som vedlikeholder dem. Praksisen vektlegger

en størst mulig grad av automatisering når modeller skal ut i produksjon. Hos bedrifter som planlegger få maskinlæringsmodeller med et lavt vedlikeholdsbehov, så kan en produksjonspipeline med manuelle triggere for trening, validering og utkjøring holde mål. Dette gjelder ikke Statens vegvesen, da det er en stor organisasjon som satser tungt på digitalisering, maskinlæring og automatisering av arbeidsoppgaver. I deres tilfelle vil det lønne seg å innføre MLOps først som sist, noe som vil omfatte at hele pipelinen består av automatiske triggere. Her skal alt fra innførsel av data til levering av prediksjoner skje så sømløst som mulig.

Leverer en modell til produksjon

Stegene fra utvikling til overvåkning av en maskinlæringsmodell kan overordnet beskrives som vist under. Når modellen utarbeides for første gang handler de første punktene også om å skape selve modellen, når modellen settes i produksjon kan disse tolkes som å hente inn ny data, analysere datatype og forberede disse for modellen.

1. Hente ut data

Velge relevant data fra diverse datakilder.

2. Analysere data

Undersøkende dataanalyse utført av dataviter. Handler om å forstå tilgjengelig data som behøves for å bygge maskinlæringsmodellen.

3. Forberede data

Her formatteres og renses data, samt splittes inn i sett for trening, validering og testing.

4. Trening av modellen

Her implementeres i første omgang forskjellige algoritmer som trenes på forberedt data, samt en innstilling av hyperparametere. Dette er parametere som kontrollerer maskinlæringsmodellens læringsprosess. De påvirker ikke læringen i seg selv, men kvaliteten på prosessen rundt. Disse varierer for eksempel ut ifra hvilken algoritme dataviteren velger å basere modellen på. Resultatet er en ferdigtrent modell.

5. Evaluering av modellen

Modellen evalueres på et sett av data den ikke har blitt eksponert for

tidligere, også kalt test-sett. Utdata fra denne fasen er metrikker som måler modellens prestasjon.

6. Validering av modellen

Om modellen bekreftes god nok for utkjøring, og dens prediksjoner er bedre enn grunnlinja (baseline), eller tidligere godkjente modell, blir den godkjent for levering.

7. Levering av modellen

Den validerte modellen kjøres ut til et miljø for å levere prediksjoner. Denne utkjøringen kan enten være som en mikrotjeneste med et REST API for å levere nettbaserte prediksjoner, en innebygget modell til en mobil enhet, eller som en del av et system som henter ut prediksjoner via terminal.

8. Overvåkning av modellen

Når modellen er kjørt ut i produksjonsmiljø, blir den kontinuerlig overvåket. Om evnen til korrekte prediksjoner reduseres, kan det igangsettes en ny iterasjon av hele prosessen.

Videre gjennomgang av eksisterende teknologiløsninger vil så nært som mulig gjenspeile denne prosessen.

4.1.2 Plattform: Google Cloud Platform (GCP)

Google Cloud Platform er en skytjeneste som leveres av Google. Plattformen tilbyr et stort antall datatjenester som blant annet datalagring, dataanalyse og maskinlæring. Den tilbyr både IaaS, PaaS og 'serverless computing'. Statens vegvesen er allerede kjent med GCP, noe som gjorde det naturlig at bedriften ønsket en løsning som er integrerbar med denne. Utviklings-, akseptansetest- og produksjonsmiljø er av den grunn opprettet her. Sett bort i fra bedriftens krav, så er Google Cloud Platform fortsatt et solid valg av maskinlæringsplattform. GCP tilbyr et stort antall maskinlæringsverktøy, og man skal teoretisk sett kunne trene, validere og levere modeller uten å bruke programmeringskode. Plattformen tilbyr lagring av data under alle steg av pipelinen, samt trening direkte på plattformen gjennom tjenesten AI Platform. Om utviklere ønsker å lage pipelinen selv, men med egen kode, så kan dette

også kjøres i GCP, og man har valget om hvor mange av plattformens egne tjenester som skal blandes inn.¹⁶ Dette legger opp til at pipelineer kan egendefineres, og tilpasses bedriften og maskinlæringsalgoritmene best mulig.

Google Cloud Platform tilbyr et stort antall tjenester for maskinlæring. I dette prosjektet skal man ikke nødvendigvis ta i bruk alle nevnte tjenester, men siden GCP er så tett integrerbart med denne løsningen, er det fordelaktig at bedriften har en overordnet oversikt over mulighetene.²⁰

- **Cloud AutoML**

En tjeneste som lar brukerne automatisk trene, evaluere, forbedre og kjøre ut egendefinerte maskinlæringsmodeller direkte i Google Cloud. Her benyttes kun et grafisk grensesnitt. Om brukeren ønsker det kan man også koble egendefinert kode opp mot AutoML.¹⁷

- **AI Platform**

Her kan utviklere av maskinlæringsmodeller ta deres modeller fra utvikling til produksjon og levering. AI Platform er integrert med blant annet BigQuery og Data Labeling Service (DLS). BigQuery hjelper med lagring av store mengder data, og DLS bistår med å merke og forberede data for trening og testing av modellen. AI Platform kan også integreres med pipelineer man lager selv, da spesielt gjennom Kubeflow Pipelines og TFX-komponenter.¹⁸ Dette gjennomgås nærmere i neste avsnitt.

- **AI Hub**

Google Cloud tilbyr hver enkelt bruker en egen katalog for å laste opp og eksperimentere med maskinlæringsmodeller, AI-komponenter og ende-til-ende pipelineer. Her kan man dele ML-pipelineer, notisbøker, modeller og annet ML-innhold.¹⁹

- **BigQuery ML**

Lar deg opprette og kjøre maskinlæringsmodeller i BigQuery gjennom bruk av standard SQL-spørringer. Lar altså SQL-brukere bygge modeller ved hjelp av eksisterende SQL-kunnskap. Nyttig verktøy spesielt for datavitere som vil evaluere modeller i BigQuery, uten at de har mye kunnskap om maskinlæringssammenheng i forkant.²¹

- **Cloud Storage Buckets**

Ikke et maskinlæringsverktøy i seg selv, men svært sentralt for å kunne sette modeller i produksjon. Når man skal behandle en maskinlæringsmodell må modellen, data og tilhørende artefakter ha en lagringsplass. Dette skjer via Cloud Storage Buckets. Her oppretter enten utvikleren en bucket manuelt, eller kjører kode som oppretter dem automatisk. Disse kan nås innad hvert enkelt prosjekt, om ikke man tildeler spesielle tillatelser. Cloud Storage behøves blant annet for å lagre modelldata som treningsapplikasjon, modell, artefakter, inndata (treningsdata) og utdata.²²

Dette er bare et lite utvalg av hva som finnes av maskinlæringsverktøy i GCP. Det finnes også andre tjenester som kan være nyttig, som Dataflow²³ og Dataproc,²⁴ men disse gjennomgås ikke i dybden per nå.

4.1.3 Rammeverk: TensorFlow Extended (TFX)

TensorFlow Extended (TFX) tilbyr et rammeverk for å opprette pipeliner bestående av TFX-komponenter. Dette inkluderer delte biblioteker for integrering, som behøves for å kjøre og overvåke TensorFlow maskinlæringsmodeller.³ Dette rammeverket er et ende-til-ende-verktøy som støtter opp under alt fra transformasjon av data til trening og levering av produksjonsklare modeller.⁴ TFX implementerer prinsippene bak MLOps, og de forskjellige komponenter tilbyr lagring, konfigurering og orkestrering av maskinlæringsmodeller. TFX-pipeliner er skalerbare og takler tunge maskinlæringsoppgaver, som modellering, trening, og validering. TFX-pipeliner takler også levering og administrering av utkjøringer. TFX-pipeliner kan orkestreres av Apache Airflow og Kubeflow Pipelines.

TensorFlow Extended har flere underliggende biblioteker som støtter opp under ønsket pipeline-funksjonalitet, slik som trening og utkjøring, samt integrasjon med GCP. Videre vil fasene i MLOps listes opp, sammen med TFX-komponenter som løser de forskjellige fasene, og tjenester disse integreres med i Google Cloud Platform.

Eksempelpipeline basert på TFX:

Informasjon er hentet fra Google Cloud Platforms dokumentasjon.⁵

1. Hente ut data

I første steg høstes data, og det kommer datafiler ut som brukes til trening og evaluering av modellen.

2. Validere data

Her kan man bruke TensorFlow-biblioteket *TF Data Validation (TFDV)*⁶ for å detektere dataanomaliteter. Her valideres ny data mot dataskjema, og om det oppdages avvik må enten skjema eller data oppdateres. TFDV fungerer godt på GCP gjennom Dataflow.

Kan integreres med Google Cloud: Dataflow.

3. Transformasjon av data (førprosessering)

Her kan biblioteket *TF Transform (TFT)*⁷ tas i bruk. Når inndata er ferdig validert, splittes det som kjent opp i test- og treningssett for maskinlæringsmodellen, noe TFT kan bidra med. Systemet sitter igjen med filer for å trene og evaluere modellen, vanligvis i *TFRecords* format. Det opprettes også transformasjonsartifakter som bidrar med å eksportere den lagrede modellen etter trening.

Kan integreres med Google Cloud: Dataflow.

4. Trening og tilpassing av modell

For å implementere og trene maskinlæringsmodellen kan *TF Estimators* og *Keras*⁸ tas i bruk. Disse tilbyr API-er som kan ta i bruk utdata fra punkt 3 (transformasjon). *Keras tuner* kan tilpasse hyperparametere, eller man kan bruke andre tjenester som *Katib*.³³ Dette steget ender med en lagret modell som brukes til evaluering, og en annen som brukes til levering av modellen og prediksjoner til et grensesnitt.

Kan integreres med Google Cloud: AI Platform: Jobs.

5. Evaluering og validering av modell

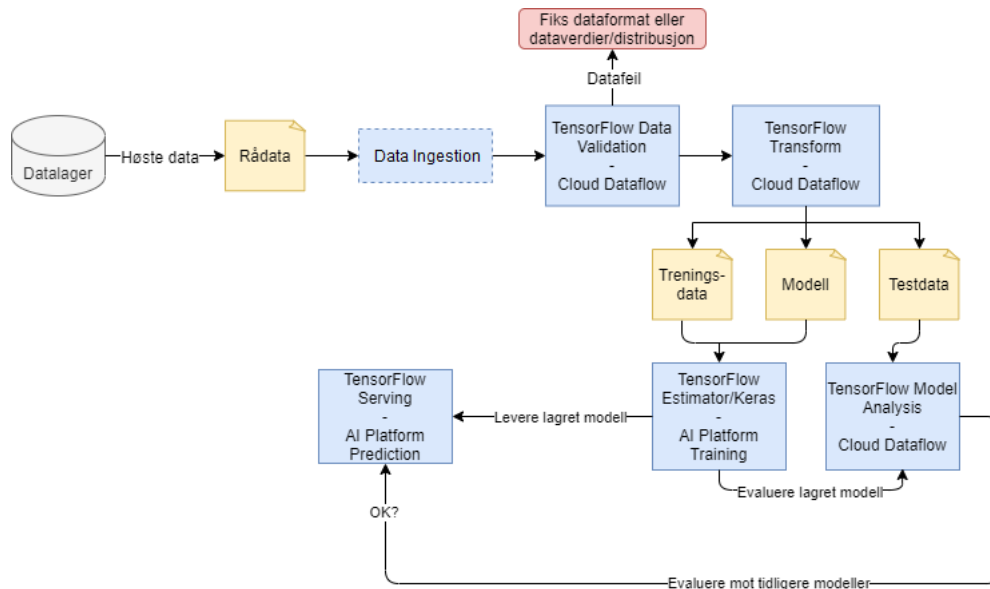
Modellen evalueres på et test-datasett for å måle dens prediksjonskvalitet. Her kan man bruke *TF Model Analysis (TFMA)*,⁹ som evaluerer modellkvaliteten i sin helhet, og identifiserer hvilke deler av modellen som yter dårlig. Denne delen bestemmer om modellen skal settes ut i produksjon eller ikke.

Kan integreres med Google Cloud: Dataflow.

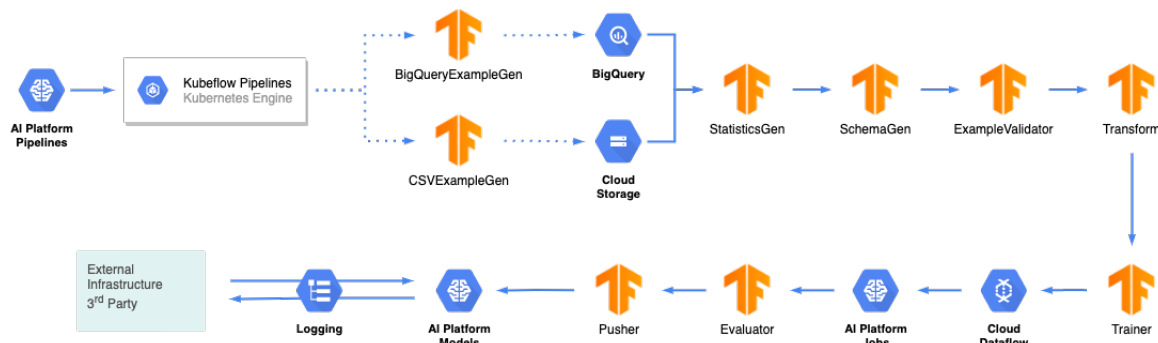
6. Levering av modell for prediksjoner

Når modellen er validert, kan den leveres som en mikrotjeneste for å gi nettbaserte prediksjoner via biblioteket *TF Serving (TFS)*.¹⁰ Utdata blir en prediksjon fra den trenete maskinlæringsmodellen. Eventuelt kan man lagre den trenete modellen i et modellregister.

Kan integreres med Google Cloud: AI Platform: Models.



Figur 2: Overordnet flyskjema for TFX-pipeline. Inspirert av TensorFlow-dokumentasjon.⁵



Figur 3: Komponentspesifikk eksempelflyt i TFX-pipeline på GCP

4.1.4 Orkestrering: Kubeflow Pipelines (KFP)

En orkestrator behøves for å koble sammen de ulike komponentene i produksjonspipelen. Den sørger for at pipelen går automatisk i ønsket sekvens, etter planlagte triggerer¹¹ slik at neste komponent i pipelen får

nødvendig inn-data fra foregående komponent. Kubeflow er et rammeverk som stammer fra konteiner-orkestratoren Kubernetes, og det består av flere tjenester for maskinlæring. En av disse tjenestene kalles for Kubeflow Pipelines (KFP). KFP lar deg sette sammen, orkestrere og automatisere maskinlæringssystemer. Hver komponent kan kjøre på Kubeflow eller Google Cloud Platform. Målet er å skape en ende-til-ende orkestrering, samt legge til rette for å enkelt kunne gjenbruke kode og pipeliner. KFP består av en 'engine' for planlegging av hvordan pipelinene skal kjøres. Her brukes Argo Workflows¹² for å orkestrere Kubernetes-ressursene. Videre innehar KFP en Python SDK¹³ for å definere og manipulere komponentene. Orkestrereren kan også samhandle med notebooks som Jupyter, og den har en metadata-lagring for informasjon om kjøring, modeller, datasett og andre artefakter. Kubeflow Pipelines tilbyr også flere definerte komponenter som kan kjøre tjenester hos Google Cloud Platform. Disse komponentene bidrar til at man kan utføre oppgaver gjennom tjenester som BigQuery, Dataflow, Dataproc og AI Platform.

Oppbygging

- **Komponenter**

Kode pakket som Docker-avbildninger.¹⁴ En enkelt komponent tar inndata-argumenter, produserer utdata-filer og utfører en fase av pipelinen.

- **Bestemme sekvens**

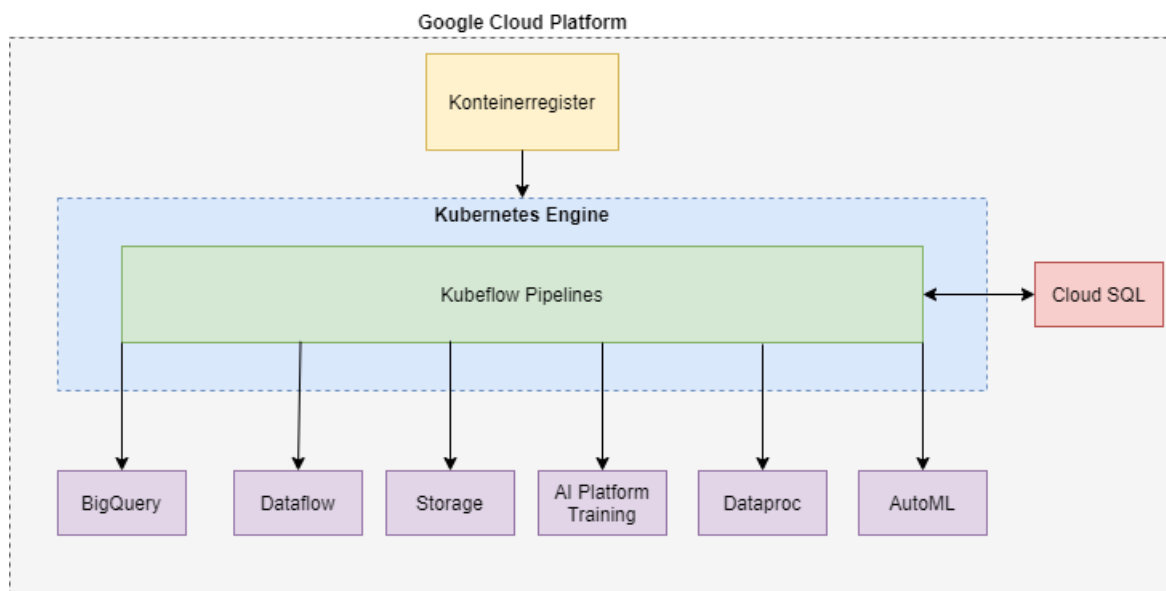
Her spesifiseres sekvesen pipelinen skal kjøre i. Dette defineres gjennom et Python *domain-specific language (DSL)*.¹⁵ Et steg i pipelinens definisjon vekker en komponent i pipelinen. Om den er svært kompleks kan komponenter også kjøre iterativt, eller kun kjøres etter planlagte hendelser.

- **Inndata-parametere**

En pipeline behøver et sett av inndata-parametere. Disse verdiene sendes til alle pipelinens komponenter, og innehar blant annet kriterier for hvordan data skal filtreres, og hvor man skal lagre artefaktene som pipelinen produserer.

Om man velger å kjøre Kubeflow Pipelines integrert med Google Cloud

Platform kan det se ut som i figur 4. Dette prosjektets løsning er ikke like tett integrert med tjenestene i GCP, men det er en fordel å ha oversikt over mulig integrasjon om bedriften ønsker en større sammenkobling i fremtiden. Flytskjema er inspirert av figur i Google Cloud Platforms dokumentasjon.⁵



Figur 4: Kubeflow Pipelines integrert med Google Cloud Platforms ML-verktøy

4.2 Verktøy og ressurser

Som det tidligere har blitt nevnt så utvikles løsningen i Google Cloud Platform, og oppdragsgiver har valgt å dele dette inn i tre forskjellige GCP-prosjekt, videre omtalt som *miljø*. Disse består av utvikling-, akseptansetest- og produksjonsmiljø. Dette avsnittet vil først beskrive hva et prosjekt i Google Cloud Platform er, før det går inn på de tre miljøene denne utviklingsoppgaven er delt inn i. Implementasjon og drift av løsningen er som beskrevet i forstudierapport²⁵ ikke inkludert i oppgavens omfang, og vil derfor ikke tas hensyn til.

4.2.1 Prosjekter i Google Cloud Platform

Et prosjekt i Google Cloud Platform er en måte å organisere bedriftens ressurser på. Det består av et gitt antall brukere, API-er med innstillinger for autentisering og monitoring, samt en oversikt over hvor mye penger

prosjektet har brukt. Brukere har altså valget mellom å sette alle sine ressurser i et enkelt prosjekt, eller å fordele og sortere dem utover flere. Hos en stor organisasjon som Statens vegvesen har sistnevnte vært den beste metoden. Dette forenkler prosessen med å sikre at brukere kun har tilgang til de prosjektene de er involverte i, slik at man ikke ødelegger viktige eller urelaterte bedriftsressurser om man gjør en feil innad eget prosjekt. GCP har noe som kalles Identity and Access Management (IAM). Dette er prosjektes og organisasjonens «tilgangskontroll». Tilgang kan gis i forskjellige grader til ansattes Google-kontoer, eller til såkalte Service Accounts. Disse kontoene tillater applikasjoner å autentisere seg på tvers over Google Cloud-ressurser og tjenester.²⁸ Eksempelvis om man ønsker å flytte data fra en applikasjon til en 'bucket', så kan en Service Account være nyttig. I dette prosjektet er det som nevnt innledningsvis tildelt tre prosjekter, disse følger fremgangsmåten bak 'Utvikling, testing, akseptanse og produksjon (DTAP)²⁹', som består av følgende steg:

- **Utvikling**

Løsningen utvikles i et utviklingssystem, dette behøver ikke å være innad prosjektet. Dette systemet behøver ikke å ha noen testmuligheter.

- **Testing**

Når utvikleren mener at løsningen er klar, leveres den til testmiljøet for å verifisere at den fungerer som forventet. I denne løsningen har det vært nødvendig å teste såpass iterativt, at utviklings- og testmiljø har gått noe om hverandre.

- **Akseptanse**

Når testingen er suksessfull leveres løsningen til et akseptansemiljø. Her vil bedriften sikre at løsningen møter deres krav.

- **Produksjon**

Når bedriften er fornøyd med løsningen, kan den implementeres eller tas i bruk. Dette skjer i produksjonsmiljøet.

4.2.2 Utviklingsmiljø

Prosjektgruppen har fått tildelt tre miljøer i Google Cloud Platform fra Statens vegvesen. Det første fungerer som et utviklingsmiljø, der forslag til løsning blir skapt. Selv om et utviklingsmiljø i teorien ikke behøver testmuligheter, så har dette vært et behov i denne sammenhengen. Dette har skjedd på bakgrunn av at flere verktøy for utvikling av pipeliner ligger inne i GCP, og det blir svært vanskelig å komme frem til en løsning kun basert på lokale utviklingsplattformer.

Jupyter Notebook

Det mest brukte lokale verktøyet for utvikling av løsningen har vært Jupyter Notebooks. Dette verktøyet kan kjøres gjennom AI Hub og virtuelle maskiner (instanser) på GCP, eller på lokal datamaskin. Jupyter Notebooks er en nettbasert applikasjon med åpen kildekode, som lar brukeren opprette og dele dokumenter som inneholder blant annet kode, likninger og virtualiseringer. Dette egner seg godt til å utvikle og trene maskinlæringsmodeller lokalt, og brukes derfor mye av datavitere.²⁷ Teknologien er godt integrert med Google Cloud Platform, noe som gjør det lett for datavitere å enten bruke notisbøker rett i GCP, eller å overføre ferdige notisbøker til egen AI Hub når de skulle ønske. I dette prosjektet har Jupyter Notebooks blitt brukt til å blant annet skrive og kompilere pipelinene.

4.2.3 Akseptanse- og testmiljø

Når løsningen er ferdig utviklet skal den inn i akseptanse- og testmiljø. Disse miljøene er vanligvis separerte, men i denne sammenhengen er de slått sammen. Her skal man resprodusere løsningen, altså pipelinen, med minst mulig komponenter. Målet er å se om man kan resprodusere løsningen enkelt, og bruke flere modeller og/eller pipeliner i samme prosjekt. Løsningen er altså klar, men man må oppdage hvordan den best mulig kan implementeres.

4.2.4 Produksjonsmiljø

Da oppgaven er avgrenset til utvikling og testing, men ikke implementasjon, utgår dette miljøet i denne sammenhengen. Ved videre arbeid med løsningen vil det involveres, men dette gjennomgås ikke i denne rapporten.

4.3 Detaljerte løsningsbeskrivelser

For å tilby en kontekst for løsningsbeskrivelser vil en overordnet liste over deloppgaver som må løses presenteres, som videre peker på tekniske løsninger på deloppgavene. Videre presenteres de tekniske løsningene som omfattes i detalj.

1. Nye versjoner av modell

Det er ønskelig å kunne deploye nye versjoner av maskinlæringsmodeller

Løsning: TFX Evaluator, TFX Pusher

2. Eksponere maskinlæringsmodell

Maskinlæringsmodellen må kunne eksponeres for input og brukes fra andre applikasjoner

Løsning: TFX Pusher, AI Platform: Models

3. Overvåke maskinlæringsmodell

Det er ønskelig å kunne overvåke maskinlæringsmodellen og dens evne til å predikere riktig

Løsning: GCP logging og monitoring

4. Skalere pipeline

Pipeline må kunne skaleres med tanke på kapasitet og ytelse

Løsning: Kubernetes Engine i GCP

5. Trene maskinlæringsmodeller effektivt

Maskinlæringsmodeller må kunne trenes i GCP

Løsning: TFX Trainer, AI Platform: Jobs

6. Lagre maskinlæringsmodeller

Maskinlæringsmodeller må kunne lagres på en hensiktsmessig måte

Løsning: TFX Pusher, AI Platform: Models

4.3.1 TensorFlow Extended (TFX)

Her kommer detaljert beskrivelse av de TFX-komponenter som er direkte relevant for deloppgavene i kapittel 4.3. En gjennomgang av hvordan TFX-komponentene henger sammen, samt en forklaring av de øvrige komponentene er å finne i kapittel 4.3.4.

4.3.1.1 TFX Evaluator Evaluator analyserer modellen som har blitt trent i pipelinen og ser hvordan den yter på evalueringssettet. Denne kan man konfigurere for å se nærmere på ytelse i spesifikke tilfeller. I forstudiet så vi på muligheten for å velge den modellen som yter best i det spesifikke tilfellet den skal brukes til, og beslutningsgrunnlaget for dette gis fra TFX Evaluator. Evaluator kan også brukes til å sammenligne den nye modellen med den gamle modellen, og automatisk velge å deploye den nye modellen eller skrote den.

4.3.1.2 TFX Pusher TFX Pusher deployer modellen fra pipelinen til et endepunkt. I dette tilfellet deploys modellen til AI Platform: Models. Modellen som deploys er den som blir godkjent av TFX Evaluator.

4.3.1.3 TFX Trainer Trainer utfører selve treningen av modellen. Komponenter tar inn data og konfigurasjon, og trener en modell ut i fra dette. Denne kan konfigureres til å kjøre treningen i AI Platform: Jobs, og er ønskelig å gjennomføre i dette prosjektet. Konfigurasjon av treningen må gjøres av en dataviter, så bachelorgruppen må lage en løsning for hvordan dataviteren skal få lagt inn konfigurasjonen.

4.3.2 Google Cloud Platform (GCP)

4.3.2.1 AI Platform AI Platform i GCP tilbyr en rekke funksjonalitet for maskinlæring på skyen. På AI Platform finner man alt man trenger fra å skape modeller, trene dem og eksponere dem. Funksjonalitet herfra skal integreres i pipelinen for å integrere funksjonalitet med resten av GCP.

Models Her lagres trente maskinlæringsmodeller. Disse kan man eksponere for tredjepartsapplikasjoner med API-kall som er ferdig laget i GCP. Dette gjør at man slipper å lage API-kallene selv. Pipelinen skal deploye modeller hit på slutten av pipelinen. Versjoner grupperes etter modell, og man kan spesifisere en standardversjon for hver modell.

Pipelines Under pipelines ligger instanser av Kubeflow Pipelines. Hver instans kan ha flere pipelineer og modeller, men det kan være hensiktsmessig

å skille disse i forskjellige instanser. Mer om dette i kapittel 4.3.3.

Jobs Her skjer selve treningen av modeller i AI Platform. Pipeline skal kjøre treningen her fremfor å gjøre det lokalt på orkestratoren. All konfigurasjon av treningen vil komme fra TFX Trainer.

Notebooks Her ligger Jupyter Notebooks. Dette skal brukes som utviklingsmiljø for å utvikle pipeline og deploye dem til orkestratoren. Når man oppretter en notebook, har den allerede alt man trenger av verktøy for å drive med maskinlæring installert, og gir på den måten et miljø som er klart til bruk og enkelt å reprodusere.

4.3.2.2 Logging og monitorering En fordel med å integrere GCP i så mange ledd som mulig i pipeline, er at logger samles på ett sted. Logging i GCP samles på ett sted og har et eget spørrespråk for å finne frem til logger. Ved hjelp av dette kan logger fra de forskjellige stegene i pipeline leses av.

4.3.2.3 Kubernetes Engine Her tilbys et kontainerbasert miljø for deploying av applikasjoner. I dette prosjektet er selve pipeline som kjører i Kubeflow Pipelines å anse som en applikasjon som kjører i Kubernetes Engine. Applikasjoner nytter clustere for porsjonering av kapasitet.

4.3.3 Kubeflow Pipelines (KFP)

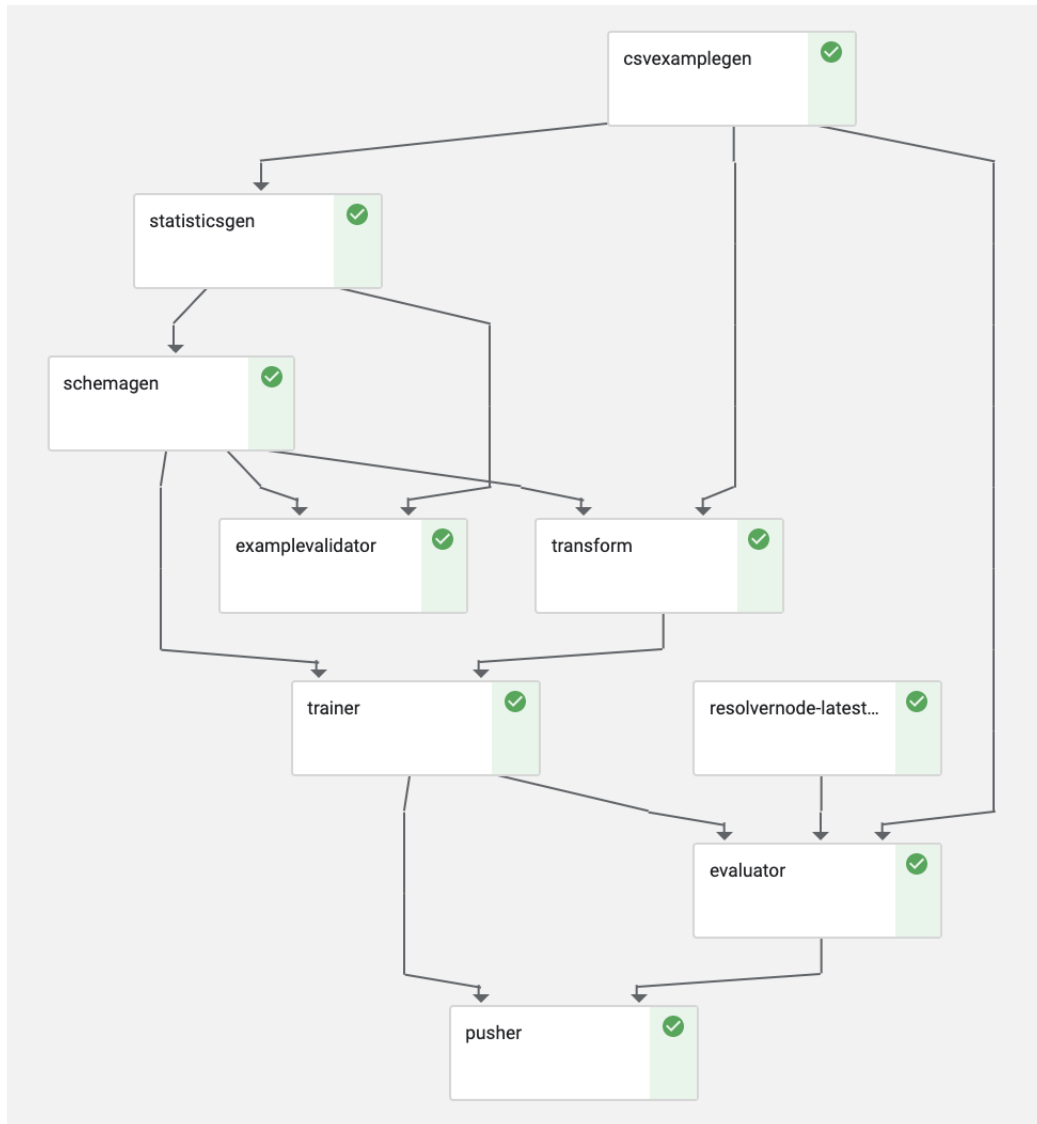
Rent teknisk konfigureres pipeline i TFX, hvor den blir satt opp til å bruke komponenter og funksjoner i GCP. Videre kompiles TFX-koden til en konfigurasjonsfil som er lesbar for KFP. Deretter opprettes pipeline i KFP, som kjører på et cluster i Kubernetes Engine på GCP. Dette betyr at GCP står for alt av nødvendig datakraft og at alt kan logges og monitoreres i GCP. Dette er svært nyttig da SVV allerede kjenner skyplattformen og benytter den i stor grad.

Valget om KFP som orkestrator er i hovedsak begrunnet i at det er god integrasjon mot GCP, men også at KFP er kompatibelt med andre ML-rammeverk enn TensorFlow. Dette legger til rette for å utvide støtte for

andre maskinlæringsmodeller i fremtiden. Øvrige argumenter omhandler skalerbarhet, noe Kubernetes Engine har god støtte for.

4.3.4 Flyt i pipelinen

Dette kapittelet skal beskrive flyten i pipelinen slik den blir konfigurert med TFX-komponentene beskrevet i kapittel 4.3.1.



Figur 5: TFX-komponentene satt sammen i Kubeflow

Examplegen er første stopp i pipelinen og er ansvarlig for inndataen. Dataen blir delt opp i trenings- og evalueringssett som blir brukt av Transform

og StatisticsGen. ExampleGen kan konfigureres til å hente data fra csv-filer eller direkte fra BigQuery.

Resolvernoder er den andre inngangsnoden i pipelineen. Denne henter inn gjeldende versjon av modellen for å sammenligne den nye i Evaluator.

StatisticsGen genererer statistikk fra dataen fra examplegen.

Schemagen lager skjemaer av dataen fra examplegen. Skjemaet beskriver inndataen med typer, kategorier og rammer. TFX-komponenter for datavalidering, transformering og modellanalyse bruker skjemaet som kommer herfra. Skjemaet er automatisk generert og stemmer ikke nødvendigvis 100%, så dataviter bør se over og eventuelt modifisere. Skjemaet kan skrives helt manuelt, men det er gjerne greiest å modifisere et som kommer fra denne komponenten. Skjemaet skrives i klartekst, noe som gjør det enkelt å redigere av datavitere.

Examplevalidator validerer inndataen fra examplegen mot skjemaet i schemagen og ser etter avvik. Resultatet fra denne komponenten brukes ikke videre i pipelineen.

Transform tar imot skjemaet fra Schemagen og data fra Examplegen og forprossesserer dataen ut ifra en funksjon som må forsynes av dataviter.³⁰ Denne funksjonen er spesifikk for hver modell og krever teknisk kompetanse innen maskinlæring, så prosjektgruppen skal ikke utvikle disse funksjonene, men legge til rette for at dataviteren kan legge inn sin.

Trainer utfører treningen, og tar inn utdataen fra Transform og SchemaGen, såvel som en konfigurasjonsfil for treningen som forsynes av dataviter. Etter Trainer har kjørt, kommer det ut en ferdigtrent modell.

Evaluator sammenligner den nylig trente modellen med en baseline eller en tidligere versjon for å vurdere ytelsen mot hverandre. Denne kan konfigureres

til å se på mindre segmenter av dataen for å sammenligne ytelsen i spesielle tilfeller. Evaluator returnerer en vurdering av beste modell.

Pusher tar modellen fra Trainer og vurderingen til Evaluator og deployer modellen til endepunktet. Dette er siste stopp i pipelinen.

4.3.5 Forutsetninger og avhengigheter

- Google Cloud Platform
- Brukerkonto eller Service Account tilknyttet prosjekt i GCP
- Maskinlæringsmodell skrevet i TensorFlow
- Konfigurerte moduler for TFX (trainer og transform)

4.3.6 Programvare og systemkrav

- TensorFlow $\geq 2.3.2$
- TFX $\geq 0.26.1$
- KFP $\geq 1.4.0$
- Google Cloud SDK $\geq 327.0.0$
- Python $\geq 3.7 < 3.9$
- Kommandolinjegrensesnitt; CMD (Win), Bash (Lin), Bash/Terminal (Mac)

4.3.7 Oppdatert GANTT-diagram

Oppdatert GANTT-diagram ligger på prosjektsiden i Teams. Om leser ikke har tilgang til denne kan kopi forespørres. Figur 6 representerer den faktiske arbeidsprosessen, og figur 7 viser det som var planlagt fra start. Som man ser har designrapporten vært mer omfattende enn forventet, noe som er naturlig i et slikt innovativt prosjekt.

Uke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Forstudie																				
Systemkrav-/designrapport																				
Driftsdokument/driftsrapport																				
Sluttrapport																				
Vurdering av gruppesamarbeid																				
Presentasjon																				

Figur 6: Faktisk arbeidsprosess.

Uke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Forstudie																				
Systemkrav-/designrapport																				
Driftsdokument/driftsrapport																				
Sluttrapport																				
Vurdering av gruppesamarbeid																				
Presentasjon																				

Figur 7: Planlagt arbeidsprosess.

5 Alternative flyter i GCP

Ut over valgt teknologi og løsning har prosjektgruppen utarbeidet en oversikt over alternative flyter for pipeline i Google Cloud Platform som kan være til nytte for vurdering av andre løsninger. Dette er kun en overordnet oversikt og vil ikke beskrives ytterligere i dybden.

ML-ARKITEKTUR

INNFORING AV DATA

STREAMING

Pub/Sub
Apache Kafka

BATCH/STRUCTURE

BigQuery

TRANSFORMING

Cloud
Dataflow

ANALYSE OG VALIDERING AV DATA

Datalab
Data Studio
Monitoring

PREPPING AV DATA

Dataprep
Dataproc
Dataflow

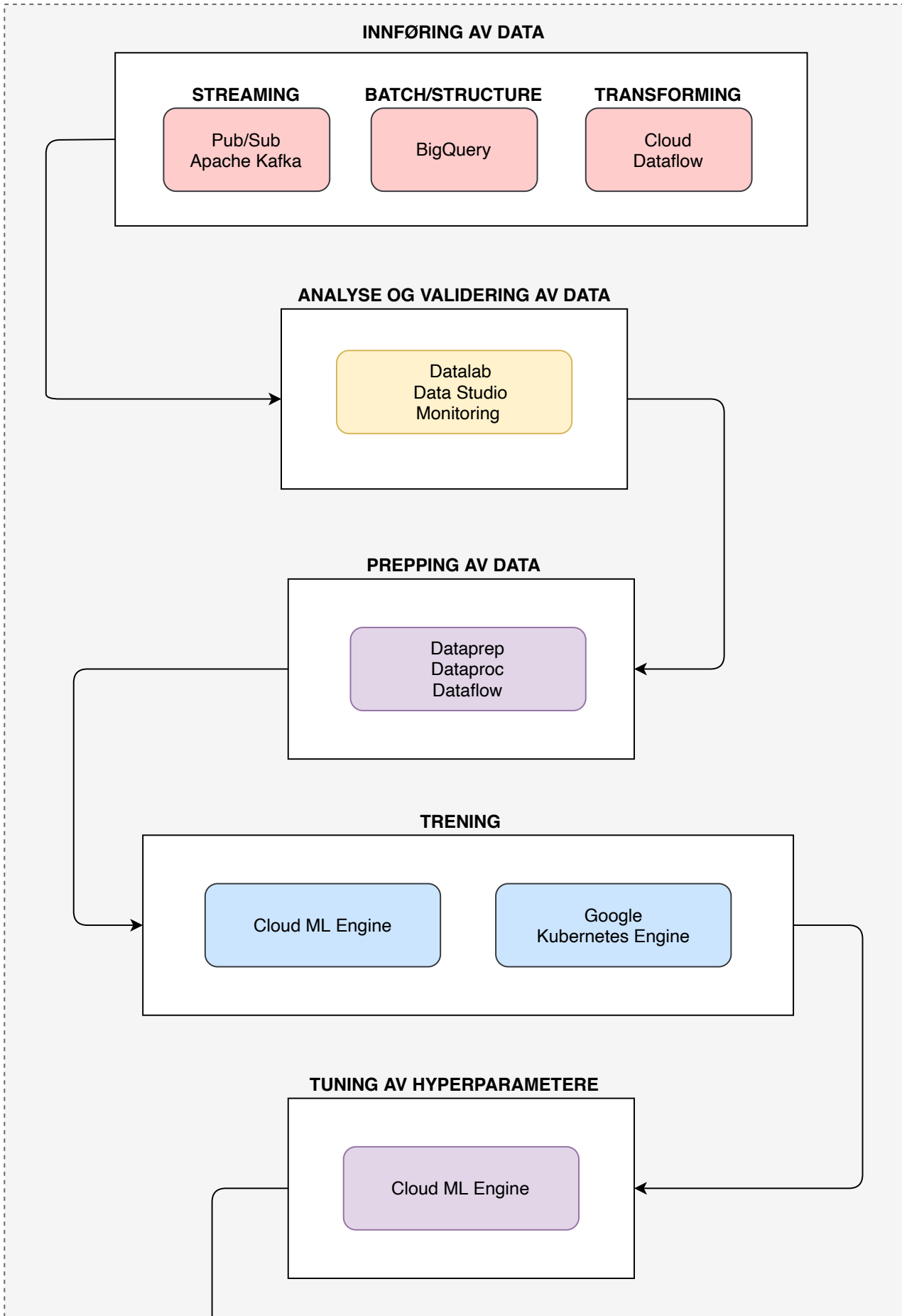
TRENING

Cloud ML Engine

Google
Kubernetes Engine

TUNING AV HYPERPARAMETERE

Cloud ML Engine



EVALUERING OG VALIDERING AV MODELL

TensorFlow
Extended
(TFX)

SERVING: INTERAKSJON MED BRUKERE

Cloud ML Engine

TensorFlow Serving
Kubernetes Engine

LOGGING

GCP Logging

RAMMEVERK OG ORKESTRERING

Google Cloud
Composer
Apache Airflow

Argo
Google Kubernetes

Kubflow

VALG AV TRENINGSMODUS

STATISK

Hente data

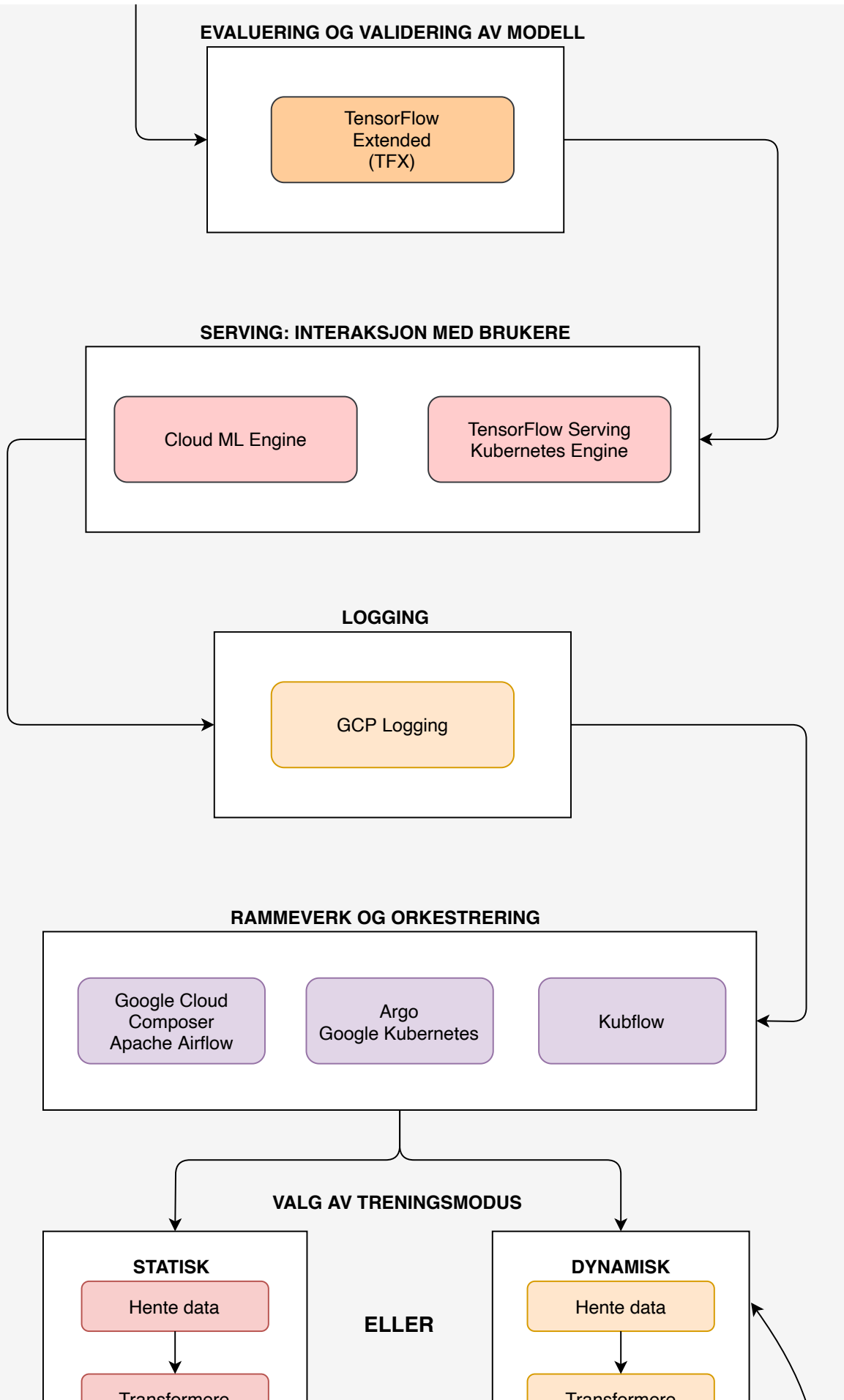
Transformere

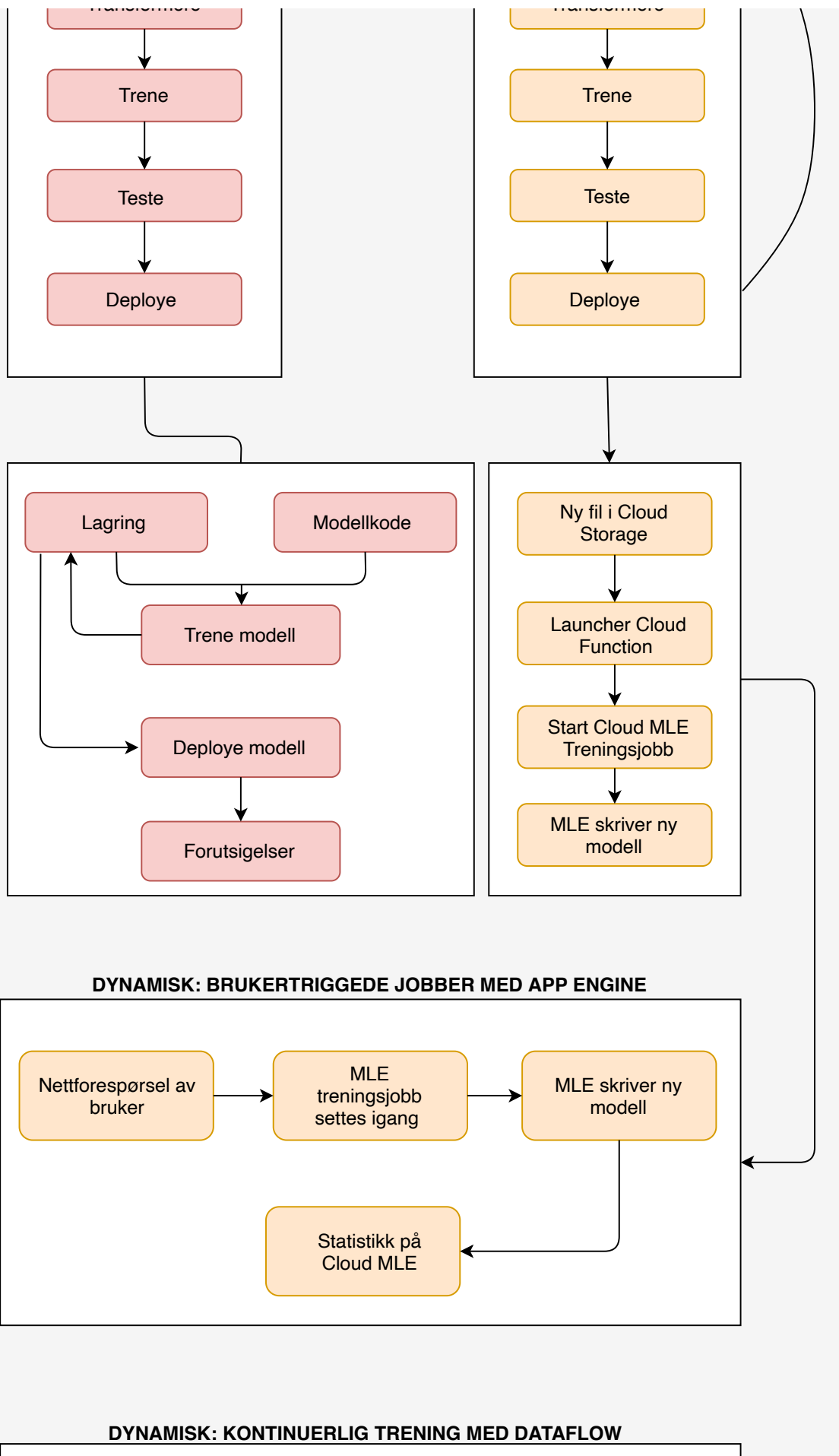
ELLER

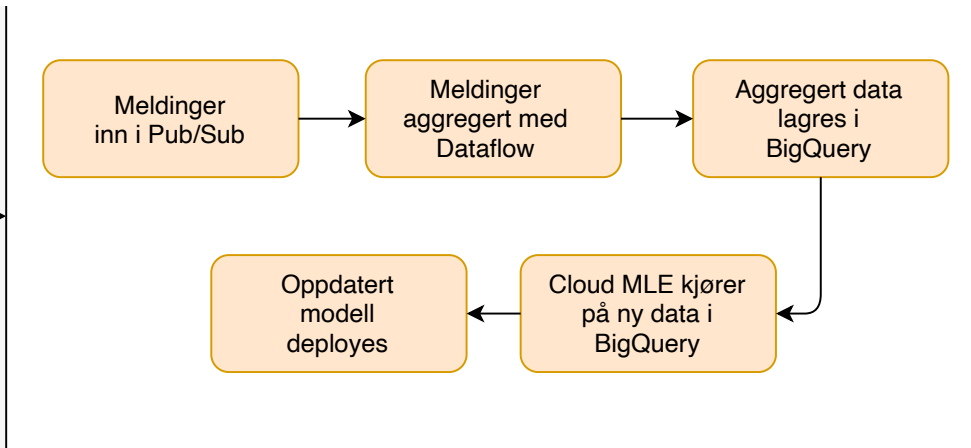
DYNAMISK

Hente data

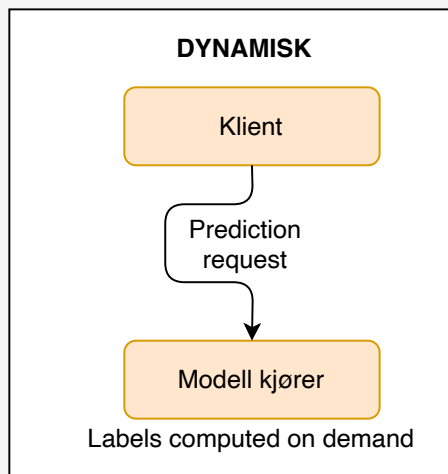
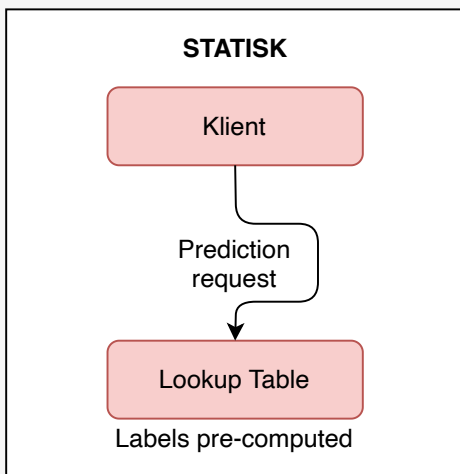
Transformere







INTERFERENS



Referanser

- [1] L. Meisingseth *Presentasjon om Saga*, 2019.

Hentet fra:

https://www.vegvesen.no/_attachment/2849432/binary/1349698?fast_title=Saga-Stordataplatform+for+transportdata.pdf.

Lastet ned: 15.02.2021

- [2] Google Cloud, *MLOps: Continuous delivery and automation pipelines in machine learning*, 2020.

Hentet fra:

<https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>.

Lastet ned: 22.02.2021

- [3] TensorFlow, *The TFX User Guide*, 2021.

Hentet fra:

<https://www.tensorflow.org/tfx/guide>.

Lastet ned: 23.02.2021

- [4] V. Tatan, "Intro to ML Ops: Tensorflow Extended (TFX)", *towards data science*, Apr. 2020. [Online].

Hentet fra:

<https://towardsdatascience.com/intro-to-ml-ops-tensorflow-extended-tfx-39b6ab1c7>

Lastet ned: 23.02.2021

- [5] Google Cloud, *Architecture for MLOps using TFX, Kubeflow Pipelines, and Cloud Build*, 2021.

Hentet fra:

<https://cloud.google.com/solutions/machine-learning/architecture-for-mlops-using-tfx-kubeflow-pipelines-and-cloud-build>.

Lastet ned: 23.02.2021

- [6] GitHub: TensorFlow, *TensorFlow Data Validation*, 2021.

Hentet fra:

<https://github.com/tensorflow/data-validation>.

Lastet ned: 23.02.2021

- [7] TensorFlow, *Get Started with Tensorflow Transform*, 2021.
Hentet fra:
https://www.tensorflow.org/tfx/transform/get_started.
Lastet ned: 23.02.2021
- [8] TensorFlow, *Estimators*, 2021.
Hentet fra:
<https://www.tensorflow.org/guide/estimator>.
Lastet ned: 23.02.2021
- [9] TensorFlow, *Getting Started with TensorFlow Model Analysis*, 2021.
Hentet fra:
https://www.tensorflow.org/tfx/model_analysis/get_started.
Lastet ned: 23.02.2021
- [10] TensorFlow, *Serving Models*, 2021.
Hentet fra:
<https://www.tensorflow.org/tfx/guide/serving>.
Lastet ned: 23.02.2021
- [11] Kubeflow, *Overview of Kubeflow Pipelines*, 2020.
Hentet fra:
<https://www.kubeflow.org/docs/pipelines/overview/pipelines-overview/>.
Lastet ned: 27.02.2021
- [12] Argo Project, GitHub, *Argo Documentation*, 2021.
Hentet fra:
<https://argoproj.github.io/argo-workflows/>.
Lastet ned: 01.03.2021
- [13] Kubeflow, *Building Pipelines with the SDK*, 2020.
Hentet fra:
<https://www.kubeflow.org/docs/pipelines/sdk/>.
Lastet ned: 01.03.2021
- [14] Docker Docs, *docker image*, 2021.
Hentet fra:

<https://docs.docker.com/engine/reference/commandline/image/>.

Lastet ned: 01.03.2021

- [15] Wikipedia, *Domain-specific language*, 2021.

Hentet fra:

https://en.wikipedia.org/wiki/Domain-specific_language.

Lastet ned: 01.03.2021

- [16] Google Cloud, *Google Cloud Overview*, 2020.

Hentet fra:

<https://cloud.google.com/docs/overview>.

Lastet ned: 05.03.2021

- [17] Google Cloud, *AutoML*, 2020.

Hentet fra:

<https://cloud.google.com/automl/docs>.

Lastet ned: 05.03.2021

- [18] Google Cloud, *AI Platform*, 2020.

Hentet fra:

<https://cloud.google.com/ai-platform>.

Lastet ned: 05.03.2021

- [19] Google Cloud, *AI Hub documentation*, 2020.

Hentet fra:

<https://cloud.google.com/ai-hub/docs>.

Lastet ned: 05.03.2021

- [20] crossML engineering, "Google Cloud Platform (GCP) for Machine Learning AI", *Medium*, Jul. 2020. [Online].

Hentet fra:

<https://medium.com/crossml/google-cloud-platform-gcp-for-machine-learning-ai-361>

Lastet ned: 05.03.2021

- [21] Google Cloud, *What is BigQuery ML?*, 2020.

Hentet fra:

<https://cloud.google.com/bigquery-ml/docs/introductions>.

Lastet ned: 05.03.2021

[22] Google Cloud, *Working with Cloud Storage*, 2020.

Hentet fra:

<https://cloud.google.com/ai-platform/training/docs/working-with-cloud-storage>.

Lastet ned: 05.03.2021

[23] Google Cloud, *Dataflow*, 2020.

Hentet fra:

<https://cloud.google.com/dataflow>.

Lastet ned: 05.03.2021

[24] Google Cloud, *Dataproc*, 2020.

Hentet fra:

<https://cloud.google.com/dataproc>.

Lastet ned: 05.03.2021

[25] J. A. Langholm, I. Andersen og E. F. Dalen, "Forstudierapport", *NTNU*, Feb. 2021. [Vedlegg].

Hentet fra:

Vedlegg i innlevering av oppgave.

[26] J. A. Langholm, I. Andersen og E. F. Dalen, "Ordbok", *NTNU*, Feb. 2021. [Vedlegg].

Hentet fra:

Vedlegg i innlevering av oppgave.

[27] Jupyter, *The Jupyter Notebook*, 2020.

Hentet fra:

<https://jupyter.org/>.

Lastet ned: 05.03.2021

[28] Google Cloud, *Projects*, 2021.

Hentet fra:

<https://cloud.google.com/storage/docs/projects>.

Lastet ned: 05.03.2021

[29] Wikipedia, *Development, testing, acceptance and production*, 2009.

Hentet fra:

https://en.wikipedia.org/wiki/Development,_testing,_acceptance_

and_production.

Lastet ned: 05.03.2021

- [30] Microsoft, *Feature engineering in machine learning*, 2020.

Hentet fra:

<https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/create-features>

- [31] towards data science, *Which deep learning framework is the best?*, 2020.

Hentet fra:

<https://towardsdatascience.com/which-deep-learning-framework-is-the-best-eb51431>

- [32] J. A. Langholm, I. Andersen og E. F. Dalen, "Ordbok", NTNU, Mai 2021.

[Vedlegg].

Hentet fra:

Vedlegg i innlevering av oppgave.

- [33] Kubeflow, *Kubeflow Katib: Scalable, Portable and Cloud Native System for AutoML*, 2021

Hentet fra:

<https://blog.kubeflow.org/katib/>

Lastet ned: 07.04.2021

- [34] J. A. Langholm, I. Andersen og E. F. Dalen, "Driftsrapport", NTNU, Mai 2021.

[Vedlegg].

Hentet fra:

Vedlegg i innlevering av oppgave.

- [35] J. A. Langholm, I. Andersen og E. F. Dalen, "Sluttrapport", NTNU, Mai 2021.

[Vedlegg].

Hentet fra:

Vedlegg i innlevering av oppgave.