

# WeatherVisualizer - Rapport

## Medlemmer:

Trym Haugan Berger

Lars Aasen

Brage Kongshaug Johannessen

---

Målet med prosjektet er å skape en løsning for værmelding der en kan få mer detaljert informasjon ved hjelp av AI. I dag kan man få værdata fra flere store nettsider, men for mange kan det være vanskelig å forutse hvordan dataen reflekteres i den virkelige verden. Mange har også opplevd å kle seg feil selv om de har sett på værmeldingen. For eksempel kan sterk vind føre til mye kaldere forhold enn det man tolket av temperaturen. Vi bruker derfor AI/ML til å løse disse utfordringene.

## SCOPE

Prosjektet går ut på å velge en lokasjon og få et AI-generert bilde av hvordan været ser ut der. I tillegg, ved hjelp av en «text completion» AI modell får man skriftlig data om været og generelle tips om hva man bør kle seg i. I dag kan man få værinformasjon gjennom nettsider som yr.no, vær.no, osv. Gjennom dette prosjektet får du et mer direkte bilde av hvordan været er der. Vi bruker OpenAI sin Dall-e-3 og GTP-4o. Værdata hentes frem MET Norway sin API. Prosjektet er skrevet i Python med Gradio som plattform.

Antall besøkende av siden per dag og hvor mye inntekt prosjektet skaper vil være vår «business metric». Måtene prosjektet kan få inntekt er enten reklameplassering eller betalte abonnenter for ekstra funksjoner. Fremtidige milepæler i prosjektet vil derfor være:

1. Innføre inntektskilder om prosjektet får interesse
2. Utvikle mer funksjonalitet, hente værmelding for ulike tider, skape flere bilder på en gang og implementere kartet bedre.
3. Utvide siden videre til flere typer varsler

Stakeholders i prosjektet er:

1. Utviklerne
2. OpenAI
3. Potensielle abonnenter

MET Norway er teknisk sett en stakeholder, men har deklarerert at de ikke ønsker noen formell affiliasjon med brukere av sin API, så dermed lister jeg dem ikke.

## **METRIKKER**

For at prosjektet skal anses som en suksess må brukere ha lyst til å bruke produktet regelmessig som en erstatter eller alternativ til ordinære værmeldingssider. For å nå dette må svarene fra AI-modellene gi tilfredsstillende nok respons konsistent. For å måle dette kan vi generere mange svar fra modellene for å så måle «accuracy» ved å se hvor mange svar var riktig i forhold til totale svar generert. På denne måten kan vi følge med på prosjektets ytelse ved forandringer.

Per dags dato koster det oss 8 cent per henting av værdata på grunn av OpenAI kostnadene. Om vi ser vekk fra andre kostnader som lønn som ikke er relevant i prosjektet vårt i dag, så må vi tjene mer enn 8 cent per forespørsel fra brukere i gjennomsnitt for at prosjektet skal være lønnsomt.

## DATA

Vi henter all nødvendig værdata fra met.no sin API. Vi bruker data om temperatur, vindhastighet, fuktighet, sky-område, og nedbør i området. MET sin API tar en lengdegrad og breddegrad parameter, og sender tilbake et JSON Objekt med data for området. Dataen er allerede godt formatert så all behandling vi trenger å gjøre er å lese den ut av JSON format til variabler.

Siden vi henter data fra MET Norway må vi være forsiktig med antall forespørsler, så vi ikke blir blokkert. Om prosjektet blir større blir vi nødt til å iverksette et mer robust cache system for data for å redusere forespørsler om data vi allerede har hentet.

Brukerdata (latitude og longitude) blir samlet inn fra manuell inntasting, men kartet på nettsiden kan bli anvendt av brukeren til å samle inn koordinatene til ønsket sted. Når brukeren klikker på kartet vil koordinatene for stedet bli vist.

## MODELLERING

Som nevnt bruker vi OpenAI sin Dall-E 3 image generation modell og GPT-4o text completion modell.

Fordi vi bruker ferdig trente modeller, vil det som avgjør ytelsen være hvor godt vi utnytter modellene, og hvor godt vi tilpasser modellene til vårt prosjekt.

For å tilpasse modellene gir vi de skreddersydde forespørsler med innsatte variabler. Her er et eksempel fra forespørsel til Dall-E 3:

```
"The weather is {weatherCode}, it is {temp} degrees celsius, the humidity is {humid} percent, the wind speed is {windspeed}, the amount of downfall is {precip} and the sky has a {cloudspercentage} percentage cloud coverage."
```

For å utforske hvordan vi kan oppnå bedre resultater må vi finne ut hvilken fraser som øker konsistensen av resultatene. Visse ord, rekkefølgen av presentasjon av data, og ekstra instruksjoner kan føre til store variasjoner i svarene som blir mottatt. Vi har opplevd at tilleggsinstruksjoner kan hjelpe mot å få uønskede resultater.

Følgende tilleggsinstruksjon har vi opplevd har minsket forekomst av bilder som ikke er relatert til været:

```
"Create a picture of landscape with focus on the weather.  
The weather and landscape should fit the given location."
```

Samtidig må vi være forsiktig med å ikke gi unødvendig mange instruksjoner siden dette og kan føre til lavere ytelse.

En god måte å utvikle tilpasninger vår av modellen vil være å prøve ulike instruksjoner, registrere instruksene i et skjema, og rangere ytelsesutfallet til instruksjonen for å finne de beste kombinasjonene.

## DEPLOYMENT

Vi hoster nå prosjektet gjennom Gradio. For å forbedre systemet må vi monitorere resultater som blir produsert under drift. Vi kan lette bruke metodene beskrevet i modellerings-delen av rapporten til å justere modellene sine ytelser under drift.

Etter hvert som OpenAI gir ut nye modeller kan vi bytte til disse for å få økt ytelse. Vi må aktivt monitorere forandringer i OpenAI og MET Norway sine API's for å holde prosjektet i drift. Om vi bytter til nyere modeller må vi vurdere om vi er nødt til å starte prosessen med tilpasning på nytt, ettersom de kan oppføre seg annerledes med de samme instruksene.

Gradio er for det meste bare en plattform for å deploye demo av enkle ML prosjekter. Dermed om systemet skal bli bedre i fremtiden vil det måtte bli bygget en nettside mer fra bunnen av. For eksempel har gradio begrensninger på kjøring av custom javascript, og derfor kan ikke kartet overføre lengde og breddegrad automatisk til Gradio sin input når kartet trykkes på.

## REFERANSER

**LinkedIn**, u.d. How do you evaluate the feasibility of AI projects?. [Internett]

Tilgjengelig på:

<https://www.linkedin.com/advice/3/how-do-you-evaluate-feasibility-ai-projects>

[Funnet 2024].

**Zapier**, u.d. How to make a custom GPT. [Internett]

Tilgjengelig på: <https://zapier.com/blog/custom-chatgpt/>

[Funnet 2024].

**MET**, u.d. MET Weather API. [Internett]

Tilgjengelig på: <https://api.met.no/>

[Funnet 2024].

**Gradio**, u.d. Build & share delightful machine learning apps. [Internett]

Tilgjengelig på: <https://www.gradio.app/>

[Funnet 2024].

**OpenAI image**, u.d. Image generation. [Internett]

Tilgjengelig på: <https://platform.openai.com/docs/guides/images/image-generation>

[Funnet 2024].

**OpenAI api**, u.d. Introduction. [Internett]

Tilgjengelig på: <https://platform.openai.com/docs/api-reference/introduction>

[Funnet 2024].