[Lettere Livs Møter (LLM)]

[Shridar Patil], [12.11.2024]

# **BESKRIV PROBLEMET**

## **SCOPE**

Målet med tjenesten er å tilby en Kunstig intelligens (KI)-assistent som planlegger dates basert på brukerens prompt der brukeren står fritt til å spesifisere ønsker og krav til planleggingen. Modellen vil dermed gi en skreddersydd date plan som brukeren kan gjennomføre eller endre ved ny prompts. Tjenesten retter seg både til single eller godt etablerte partnerskap som kan bruke tjenesten for å planlegge dates. Den har derfor et stort omfang av potensielle brukere.

Large Language Modellen (LLM) er Chat GPT som tilhører Open AI (1) via deres API-løsning (2). APIen gir en LLM som kan integreres i et program slik at modellen kan kommunisere med programmet. Videre kan programmet dermed også brukes til å spesialisere KI modellen til å bedre kunne løse oppgaven.

Per i dag er den manuelle løsningen at en eller begge personer som skal på date planlegger daten uten hjelp av KI. Videre finnes deg også andre som også har laget mer fullstendige KI dating løsninger som <https://dateplanner.ai/>. Dette er en fullstendig løsning som sorterer og organiserer flere dates, forslår planer og replikker til meldinger på datingapper. Løsningen er derfor større enn tjenesten beskrevet her og retter seg derfor kun til personer som er single. Vår løsning har derfor en bredere appell grunnet at det også egner seg til brukere som ikke kun er single og videre fokuserer kun på planlegging av dates over mange tilleggs funksjoner.

**Vurdering av tjenestens prestasjon og ytelse**

Tjenestens suksess vil måles i brukertilbakemelding og brukerstatestikk. Brukerstatestikken vil inneholde, antall besøk på nettsiden, samtaler og lignende. Dette vil dermed bedømme populariteten og videre hvor intensivt tjenesten brukes på en objektiv måte. En grov regel med brukerstatestikk er at høyt forbruk av tjenesten tilsvarer et godt produkt.

Brukeropplevelsen vil være subjektive tilbakemeldinger gitt av brukere som utnytter tjenesten. Dette vil være kommentarer som er sendt inn av brukeren der de kan skrive fritt. Disse må dermed bedømmes av utvikleren av tjenesten slik at tilbakemeldinger kan tas i etterretning. I tillegg kan det også stilles spørsmål der brukere blir bedt om kort skåre eller anmelde appen. For eksempel om de vil anbefale tjenesten til andre eller vil bruke den videre. Dette vil også være en god indikator på om tjenesten har god ytelse.

**Beskrivelse av løsningen og dets komponenter**

Tjenesten består av et hoved skript som har flere under funksjoner. Før tjenesten blir lansert har er relevante data innsamlet om relevante aktiviteter og plasser som restauranter, aktiviteter og utesteder, men er ikke begrenset til disse. Tekst filen med beskrivelse av disse forslagene blir prosessert i Open Ai sin embedding modell. Dette omformer tekstfilen til en stor datatabell som representerer teksten numerisk og modellen kan da utføre matching mellom datatabellen og brukerens prompt. I tillegg henter programmet også nyeste 7 dagers varslet fra YR sin API *Location Forecast* (2) for den relevante plassen slik at modellen også tilpasser svaret etter været. LLM modellen blir også startet med et system melding som gir modellen en overordnet beskrivelse av dens spesifikke oppgave. System meldingen er hvor værdataene blir inkorporert i modellen slik at værdataene kun hentes en gang for hver ny instans en bruker kjører.

For å utføre sammenligningssøk mellom brukerens prompt og datatabellen brukes «cosine\_similarity» fra scikit-learn biblioteket (3). Dette gjør at vi kan finne rader i datatabellen som matcher brukerens prompt og dermed gi disse til LLM modellen som relevante forslag den kan komme med når den formulerer svaret. Dette gjøres ved at denne informasjonen legges til brukerens melding før det sendes til LLM modellen.

Til slutt, for at brukeren har en grafisk fremstilling av dette produktet brukes *Gradio* som det aktuelle biblioteket for å gjøre tjenesten offentlig tilgjengelig. Her brukes et *chatboks-objekt* sammen som tar imot brukerens input og et *textbox-objekt* til å lagre samtalen med modellen. Dette er integrert med resten av modell funksjonaliteten slik at LLM modellen henter ut brukerens prompt fra chatboksen og dermed avgir svaret i tekstboksen.

**Stakeholders i prosjektet**

På **utvikler** siden:

* + Forfatteren(e) av denne rapporten

Tjenester brukt av produktutvikleren (indirekte stakeholders)

* Open Ai ved bruk av sin API (utvikler betaler Open Ai for tokens)
  + Chat GPT modellen
  + Embedding modellen
* YR sin API for henting av værdata
  + Location forecast
* Gradio som hoster nettsiden når den er offentlig

**Tidslinje for prosjektet**

Delmål 1:

* Etablere en enkel GPT modell som fungerer på terminal som har blitt spesialisert for oppgaven den skal løse.

Delmål 2:

* Koble GPT modellen til en visuell nettside bygd via Gradio. Det skal være en chatbox for å oppgi prompts og videre en tekstboks som viser meldingshistorikken.

Delmål 3:

* Samle og dermed embedde en database. Ved hjelp av en matching funksjon implementere dette inn i GPT modellen for å forbedre svarene.

Delmål 4:

* Anvende YR sin *location forecast* inn i modellen sitt svar slik at den kan ta forbehold om været for dataen slik at svaret blir enda bedre tilpasset.

Delmål 5:

* Kvalitets forbedringer rettet mot brukergrensesnittet. For eksempel bruke andre farger og redigere hvordan brukeren opplever Gradio nettsiden.
  + Fikk derimot ikke dette til å funke dessverre og fokuset ble mot slutten på å ferdigstille og levere rapporten/repositoret.

**Ressurser som anvendes i prosjektet**

Prosjektet behøver følgende ressurser

* Utvikler med kyndighet til å skape programmet som er beskrevet over. De vil være ansvarlig for å skrive koden som kommuniserer med APIene og videre skaper funksjonene som gjør at dette fungerer korrekt.
* API tjenester som er tilgjengelig for utvikleren å kalle på for å oppnå ønskede funksjonaliteter.
  + Open AI som tilbyr sin Chat GPT modell som fungerer som en LLM. Denne brukes til å generere et tilpasset svar til brukeren
  + Open AI sin embedding modell. Denne skaper en numerisk database med forslag som programmet søker i for å gi LLM modellen mer spesialiserte forslag enn den vil ha tilgjengelig alene til sitt svar til brukeren.
* Hosting tjeneste: For at nettsiden skal være tilgjengelig døgnet rundt kreves det en datamaskin som kjører koden nødvendig for å holde nettsiden oppe. Dette kan derfor ikke gjøres på utviklerens private datamaskin og blir derfor gjort vi Gradio sin hosting plattform.

## **METRIKKER**

**Krav for løsningen**

Gitt løsningens nåværende tilstand vil tilbakemeldingen fra testere/brukere om at produktet fungerer som ønsket være tilstrekkelig. Det er her ikke lagt vekt på at svarene må være så bra som mulig, men at modellen heller er «godt på vei» til å utføre oppgaven.

Hvis løsningen skulle undersøkes for kommersiell potensiale burde metrikkene utvides. For eksempel ved at løsningen kun er tilgjengelig ved et abonnement vil det være gunstig å se på både inntekten og brukertilfredsheten tjenesten gir som indikasjoner på hvor god løsningen er. Inntekten gitt ved prisen og antall brukere beskriver hvor mange av brukerne som er villig til å betale for produktet i en gitt tid. Sammen med disse dataene bør muligheten til å gi tilbakemeldinger på produktet i form av skåring og fri tekst være viktig for å få en indikasjon på hva brukeren deskriptivt og mer numerisk tenker om løsningen. Det vil også være mulig med ytterligere metrikker for å vurdere prestasjonen til løsningen videre, men det er verdt å merke at alle kun gir en indikasjon og ikke kan gi et fullstendig bilde.

Ettersom LLM modellen produserer et fritekstsvar i dette tilfellet er det ikke like lett å bruke standard maskinlærings metrikker for å avgjøre dette. Istedenfor vil det trolig være brukerens anmeldelse av tjenesten som vil gi gode metrikker. For eksempel brukertilfredshet som nevnt over. I tillegg kan brukeren gi tilbakemelding både med numeriske og deskriptive svar om hvor relevant modellens svar er ifølge brukeren og videre om de føler svarene er modellen gir er relevante for dem. Dette kan dermed gi en indikasjon på systemets ytelse.

# **DATA**

**Beskrivelse av data løsningen anvender**

Modellen som blir brukt i denne løsningen er en LLM og har derfor blitt trent på store mengder data jeg ikke kan beskrive i god detalj. For å spesialisere KI modellen bruker jeg derimot egen genererte data. Det består av et system melding til modellen som er skrevet av utvikleren om hvilken oppgave den skal løse. Den andre formen for data som kommer fra utvikleren og er oppgitt i systemmeldingen er vær data. Detter er i form av en tabell hvor hver rad representerer værvarslet for Bergen hver time 7 dager frem i tid. Kolonnene inneholder ulike forespeilede vær parametere som blant annet: temperatur, vindstyrke, sannsynlighet for nedbør og sannsynlighet for skyer.

Vi har også samlet informasjon om aktiviteter, spisesteder og utesteder i Bergens område for å kunne gi modellen forslag. Disse er lagret som tekstfiler (tilgjengelig i GitHub repoen i *data* mappen). Disse dataene har blitt embedded med Open Ai sin model og er i en json fil som er på størrelse 4 megabyte.

Til slutt kommer brukerens egne data. Brukerens input data er fritekst der de beskriver ønskene sine rundt dateplanen som skal gis til LLM modellen. Før inputet blir prosessert av modellen blir det embedded av samme modell som tekstfilene nevnt over for å dermed utføre matematisk matching der samsvar mellom brukerinputet og json filen blir oppgitt som forslag til modellen for hva den kan foreslå av aktiviteter. Etter at dette er sendt som input til modellen vil LLM modellen dermed gi et svar som output som vises i chatboksen.

**Personvern og etiske hensyn**

Tjenesten utviklet av oss lagrer ingen data når den blir brukt. Videre har Open Ai fastslått av data som generes fra APIen kun skal lagres midlertidig for eventuelle feilsøk men ikke blir lagret som datapunkt for å trene fremtidige modeller eller for å samle brukerdata. Videre krever det ingen innlogging og det er derfor ingen krav om å oppgi brukerspesifikke opplysninger ved bruk av tjenesten, brukeren står derfor fritt til å oppgi så lite informasjon om seg selv som de vil og ingen personsensitiv informasjon er nødvendig for bruk av tjenesten.

Det er derimot en geografisk begrensing til Bergen som forteller noe om personens lokalisasjon, derimot er Bergen en by og ingen ytterligere informasjon oppgis for å spore brukeren til et enkeltindivid.

Et mulig etiskdilemma kan være at modellen ved nok informasjon kan skreddersy for mye og derfor skade selvstendigheten til parten som skal planlegge daten. Ved at personen slipper å bruke energi på å planlegge vil terskelen for dates kunne bli lavere. Fjerneringen av denne barrieren til å gå ut på date kan blant single gjøre at terskelen for dates med flere mulige partnere blir lavere og dermed blir dateing mer uformelt og at hver date mister noe av sin høytidelighet. Derimot er dette kun en antatt problemstilling og det vil være andre tjenester utover KI som tilbyr samme hjelpen med å planlegge dater, for eksempel Youtube videoer, blogger eller andre nettsider. Vi tror derfor denne problemstillingen ikke er en hindring for vår tjeneste.

**Data format**

For å slippe å bruke alt innholdet i tekstfilene med forslag for hver gang GPT modellen skal svare brukes embedded data av brukerens input og tekstfilene. Dette gjør at vi kan finne likheten mellom dataene etter embedding ettersom dataene blir transformert fra fritekst til en tabell struktur som kan analyseres matematisk.

Utover dette utføres ingen rensing av dataene. Resultatet av matchingen blir lagt til meldingen til modellen med strenger som presiser av hva som er brukerens input og hva som er forslag fra databasen. Selve responsen er også friteksten som er hentet ut fra GPT modellens responsvariable. For at GPT modellen skal funke best mulig og vite hva brukeren og modellen selv har svart tidligere oppgis dataene i et samtaleobjekt med hele samtalehistorikken. Samtalehistorikkobjektet inneholder meldinger om hvem som er opprinnelsen til utsaget (KI-modellen, brukeren eller utvikleren), hva slags type data det er (denne løsningen bruker kun fritekst) sammen med selve meldingene.

# **MODELLERING**

Prosjektet brukte en LLM modell og det ble det ikke utforsket andre typer KI-modeller. Valg av LLM modell ble gjort ved å søke av etter tilgjengelige LLMer på nettet, hvor Open Ai APIen fremstod lett og hadde både LLM modellen sammen med en embedding modell i samme API noe som ga muligheten til fritekst genering og videre sammenligning av tekst data ved embedding og ble derfor valgt.

Baseline ytelse for en slik tjeneste føles derfor naturlig å være standard GPT modellen tilgjengelig på Open Ai sine nettsider hvor svaret til samme prompt kan sammenlignes. Merk at Chat GPT på Open Ai sine sider burde ha ytterligere kontekst for å løse date-planlegningen siden den ikke har blitt fortalt av systemet på forhånd at det er dette den skal utføre.

# **DEPLOYMENT**

Modellens brukergrensesnitt er bygget ved Gradio biblioteket (4). Det er her en innebygd mulighet for å gjøre nettsiden offentlig ved en lenke når applikasjonen startes. Derimot fører dette til at pcen som utfører programmet alltid må stå på og ha nett tilkobling. Dette er ugunstig ettersom det da alltid vil stå på utviklerens pc. For å løse dette problemet brukes *Hugging faces spaces* (5). Hugging face tilbyr en gratis hosting plattform som hoster prosjektet med liten kapasitet for demo testing. Siden målet med dette prosjektet er kun å vise til en fungerende demo passer dette ypperlig og ble derfor valgt til hosting tjeneste. Nettsiden er derfor tilgjengelig ved en lenke oppgitt i GitHub reposistoriet. Dette prosjektet vil derfor være oppe til enten utvikleren eller hugging face tar det ned.

**Eventuelle utbedringer:**

Etter lansering og bevis på at nettsiden fungerer er det ikke planlagt videre overvåkning eller vedlikehold av tjenesten. Det finnes samtlige utbedringer som kan utføres for å gjøre tjenesten, men det er ingen planer om å implementere disse per nå. Noen eksempler på mulige utbedringer.

* Hvert individ får en personlig bruker. Det blir dermed mulig å lagre meldingshistorikk for hver bruker for å forbedre tjenesten.
  + Brukere kan dermed gi modellen et voksende kontekstvindu slik at den svarer bedre over tid og kan justere seg bedre over å starte på nytt for hver innlogging.
  + Kan være spesielt nyttig hvis samme par skal på date nummer to eller tre og brukeren vil gi tilbakemelding fra første date.
* La tjenesten utnytte web crawling eller utvide databasen som modellen har tilgjengelig. Dette vil gi KI-modellen mer presis informasjon som dermed lar den produsere et bedre og mer spisset svar for brukerne. Samtidig vil dette også la tjenesten fungere utover Bergen og derfor anvendes av flere brukere.
  + Dette vil derimot øke driftskostnadene ettersom det vil gi mer data som må prosessers av programmet og LLM modellen.
  + Vil behøve avklaringer rundt tillatelser og begrensinger på hva slags informasjon man kan samle med web crawling.
* Lage en tilbakemeldings tjeneste for brukeren slik at utvikleren lettere kan få rapporter om feil og tilbakemeldinger fra brukerne. Begge disse rapportene kan dermed brukes for å utbedre og videreutvikle applikasjonen til å bli et bedre produkt over tid.

# **REFERANSER**

1. OpenAI, "ChatGPT Overview," [Online]. Available: <https://openai.com/chatgpt/overview/>. [Accessed: Nov. 12, 2024].
2. OpenAI, "OpenAI API," [Online]. Available: <https://openai.com/index/openai-api/>. [Accessed: Nov. 12, 2024].
3. Yr, "Weather Forecast API," [Online]. Available: <https://developer.yr.no/featured-products/forecast/>. [Accessed: Nov. 12, 2024].
4. Scikit-learn, "Cosine Similarity (sklearn.metrics.pairwise.cosine\_similarity)," [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html>.

[Accessed: Nov. 12, 2024].

1. Gradio, "Gradio - Build Machine Learning Demos and Web Apps," [Online]. Available: <https://www.gradio.app/>. [Accessed: Nov. 12, 2024].
2. Hugging Face, "Hugging Face Spaces Documentation," [Online]. Available: <https://huggingface.co/docs/hub/spaces>. [Accessed: Nov. 12, 2024].