

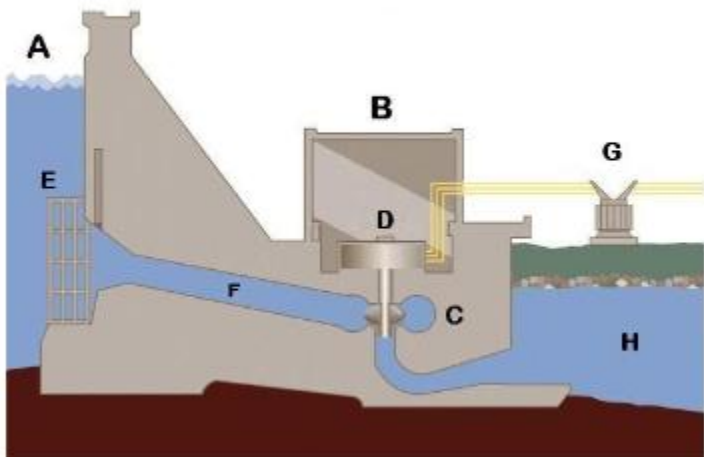
Caseoppgave sommerjobb, Digitalisering & IKT

Caseoppgaven består av 2 oppgaver. Oppgave 1 er en programmeringsoppgave hvor du skal bruke et utdelt datasett til å lage 2 maskinlæringsmodeller. På intervjuet skal du presentere koden din og hva du har gjort for oss, for eksempel i en Jupyter Notebook eller lignende. Oppgave 2 er en «konseptoppgave» hvor du skal skissere en mulig måte å bruke LLM på våre meldingslogger. Her skal du ikke skrive kode, men heller tenke i større trekk om hvordan oppgaven kan løses og hvilke hoveddeler løsningen vil bestå av.

Dersom du får knapt med tid, prioriter oppgave 1. Det er bare å sende mail dersom du har spørsmål eller noe ved oppgaven er uklart.

Programmeringsoppgave: Modellbasert overvåking

Å Energi Vannkraft ønsker å maksimere oppetiden på våre vannkraftverk. Som en del av dette overvåker vi sentrale komponenter, for å kunne varsle om potensielle feil *før* noe alvorlig har skjedd. Én av disse komponentene er generatoren (komponent D på figuren), som består av en rotor og en stator.



Statoren har 3 elektriske viklinger som dytter elektroner via en transformator ut på nettet. Jo mer effekt som produseres i generatoren, jo mer strøm går det i viklingene og desto mer varmetap blir det i statorviklingene. Varmetap arter seg som en temperaturøkning i statorviklingene, og dette varmetapet kan vi måle. Hvis statorviklingene blir for varme kan vi få skader i isolasjonen, og generatoren kan måtte repareres. Dette kan ta mange måneder å reparere. Det er derfor viktig å overvåke om temperaturene er for høye. Generatoren kjøles ved å bruke vann fra elva til å kjøle ned kjøleluft som igjen brukes i radiatoren til å kjøle generatoren. Merk at ellevannets temperatur varierer fra sommer til vinter med ca. 20 grader, slik at kjøleeffekten også varierer. Mer kjøling på vinter, mindre på sommer.

Vi ønsker å overvåke temperaturen til statorviklingene i generatoren med en metode kalt *modellbasert overvåking*. Metoden går ut på å ha både en sensor og en modell for komponenten, som i dette tilfellet er statorviklingene. Sensoren forteller hva som *faktisk skjer* i generatoren, mens modellen forteller hva som *burde skje* i generatoren. Ved å sammenligne avviket mellom sensoren og modellen kan vi varsle dersom avviket blir stort. Avviket defineres som sensormåling minus modellprediksjon.

Oppgave: Bruk datasettet du har fått til å lage 2 ulike modeller av temperaturen i statorvikling L1. Modellene skal beregne temperaturen, slik at vi kan bruke den til å varsle

dersom målt temperatur fra sensoren er mye høyere enn beregnet (predikert) temperatur fra modell. Den ene modellen kan gjerne være helt enkel (for eksempel en lineær modell), men du står fritt til å velge modelltyper selv. Du har fått utdelt et datasett (se forklaring litt lenger nede), gjør den analysen og preprosesseringen på det du mener er nødvendig. Du velger selv hvilke features fra datasettet du vil bruke som input til modellene (men ikke selve viklingstemperaturen), og du kan også legge til flere features (for eksempel værdata eller tid/sesonginformasjon) om ønskelig.

Presenter og visualiser resultatene for begge modellene slik du synes det er hensiktsmessig. Drøft litt rundt hver modell. Hvor nøyaktig er modellen? Er det noen spesielle forhold hvor den er særlig nøyaktig eller unøyaktig? Hvordan kan modellen forbedres? Hvilke datakilder har du brukt/ikke brukt og hvorfor?

Merk: Hensikten med oppgaven er å vise forståelsen din for data og databehandling, metoder i Data Science og maskinlæring, og visualisering av og drøfting rundt resultatene dine, ikke å få en absolutt best mulig modell. Så ikke bruk mye tid på for eksempel tuning av hyperparametere. Hopp over eventuelle (i dine øyne) nødvendige detaljer og forklar heller til oss hva du ville ha gjort.

Datasett: Du har fått utlevert et datasett med historiske data fra en generator fra et av våre vannkraftverk. Datasettet inneholder gjennomsnittlige målinger for hver time for en rekke målinger:

Viklingstemperatur L1 [°C]: Temperaturen i statorvikling L1, målt på et spesifikt punkt på viklingen.

Strøm [A]: Strømmen gjennom viklingene.

Spenning [kV]: Spenningen i viklingene.

Aktiv effekt [MW]: Produsert aktiv effekt i generatoren. Ved 0 eller manglende verdi kan du anta at generatoren ikke har produsert strøm

Reaktiv effekt [MVar]: Produsert reaktiv effekt i generatoren.

Kjøling kaldluft inn [°C]: Den kalde kjølelufta som sendes fra radiatoren og inn til statoren for å kjøle den ned.

Kjøling varmluft ut [°C]: Den varme kjølelufta som kommer ut etter den har kjølt ned statoren.

Vannføring [m³/s]: Vannmengden som sendes gjennom turbinen for å produsere strøm.

Turbinlager [°C]: Temperaturen i turbinlageret.

Konseptoppgave: LLM

Vi får inn meldingslogger fra våre kraftverk (se eksempel under). Disse inneholder dato/tid, hendelsestekst, meldingstype og lokasjon. Meldingstypene er for eksempel: feilmelding, forvarsel, vernmelding osv. Hendelsesteksten inneholder informasjon om hva som har skjedd.

Tid	Hendelse	Meldingstype	Stasjon
2024-12-31 12:36:14:120	Færåsen Aggr. 1 Startklar På	Driftsmelding	Færåsen
2024-12-31 12:36:13:621	Færåsen Aggr. 1 Startklar Av	Driftsmelding	Færåsen
2024-12-31 12:09:00:140	Færåsen Aggr. 1 Stillstand På	Driftsmelding	Færåsen
2024-12-31 12:02:47:746	Færåsen Aggr. 1 Startklar På	Driftsmelding	Færåsen
2024-12-31 12:02:47:608	Færåsen Turbin 1 Bypassventil Lukket	Driftsmelding	Færåsen
2024-12-31 12:02:16:682	Færåsen Turbin 1 Hovedventil Lukket	Driftsmelding	Færåsen
2024-12-31 12:02:04:574	Færåsen Turbin 1 Bypassventil Åpen	Driftsmelding	Færåsen
2024-12-31 12:01:35:053	Færåsen AGG1 AGC Trigger stopp Av	Annet	Færåsen
2024-12-31 12:01:35:052	Færåsen G1E Bryter agg offline	Driftsmelding	Færåsen
2024-12-31 12:01:35:052	Færåsen G1E Bryter spl_stop sekvens ferdig	Driftsmelding	Færåsen
2024-12-31 12:01:34:983	Færåsen G1 Spenningsregulator startet Av	Driftsmelding	Færåsen
2024-12-31 12:01:34:833	Færåsen G1E Bryter Ute	Driftsmelding	Færåsen
2024-12-31 12:00:12:888	Færåsen Aggr. 1 Styring Stopp -Kommando	Kommando	Færåsen
2024-12-31 12:00:01:045	Færåsen AGG1 AGC Trigger stopp På	Annet	Færåsen
2024-12-31 07:56:22:305	Færåsen Aggr. 1 Start *Pågå	Driftsmelding	Færåsen
2024-12-31 07:55:59:406	Færåsen Aggr. 1 Synkronisering *Pågå	Driftsmelding	Færåsen
2024-12-31 07:55:59:375	Færåsen G1E Bryter agg offline	Driftsmelding	Færåsen

Disse loggene kan være uoversiktlige og variere mye fra kraftverk til kraftverk. **Vi ønsker å bruke en LLM, for eksempel Llama, GPT-4 eller en mer «lettvekt» modell, for å spørre spørsmål på disse dataene.** Eks: «lag en liste over vernmeldinger vi har fått fra alle kraftverk de siste 24t». «Hva har skjedd på Kraftverk XXX de siste 3 timene?». Bruk RAG (Retrieval-Augmented Generation) eller evt. en annen lignende metode for å kunne finne sanntidsdata i forhold til den trente LLM-en, slik at den kan finne de nyeste meldingene som har kommet inn. Gjør nødvendige antakelser eller forenklinger om database, meldingslogg e.l. for å løse oppgaven.

1) Gjør research på LLM og RAG, AI chatbots ++

2) Basert på din research, forklar overordnet hvordan du ville implementert / løst oppgaven

Tips: Hvilken LLM kan være egnet? Er valg av språkmodell viktig? Hvordan kan du sende spørsmål via API? Hvordan kan du bruke egne data fra en database som inneholder meldingslogger via RAG?

Merk: du skal ikke skrive kode. Beskriv med tekst, diagrammer og illustrasjoner i en powerpoint eller liknende