

# Refleksjonsnotat – Miljødataanalyseprosjekt

Dette prosjektet har gitt oss en bred og praktisk innføring i anvendt programmering, spesielt innen miljødataanalyse. Vi har jobbet med alt fra datainnsamling og analyse til visualisering, og ikke minst lært verdien av versjonskontroll og samarbeid.

## Læring

Et av de mest sentrale læringspunktene har vært innføringen i Git. Selv om vi slet med å bruke Git aktivt og gjøre jevnlige commits, har det gitt oss en bratt, men verdifull læringskurve. Da vi flere ganger slet med git ble det til at gruppen jobbet mye i egne branches, og store endringer ble så commitet etter flere ukers arbeid. Dette er ikke ideelt og vi forstår nå bedre hvordan Git fungerer som et verktøy for både samarbeid og historikk i utviklingsprosjekter, og vi ser tydelig behovet for å bruke det mer strukturert i fremtiden.

Vi har også fått solid erfaring med Pandas og NumPy, spesielt i analysen av temperaturdata der vi har brukt statistiske mål som gjennomsnitt, median og standardavvik. Dette ble tydelig demonstrert gjennom vår visuelle fremstilling med Plotly, der vi brukte interaktive grafer for å illustrere klimaendringer over tid – for eksempel ved å vise temperaturutviklingen i Trondheim de siste ti årene. Vi fant det veldig verdifullt å bruke de åpne Plotly-bibliotekene som finnes på nett til å hjelpe oss med visualiseringen av grafene.

En del av prosjektet har også vært å bli kjent med enhetstesting i Python, noe vi i starten hadde lite erfaring med. Vi implementerte noen tester ved hjelp av unittest-rammeverket, men vi ser i etterkant at deknningen kunne vært bedre. Det ble raskt tydelig at tester er mer enn bare «noe man legger til på slutten», det handler om å sikre at logikken er robust og tåler endringer senere i utviklingsprosessen.

## Utfordringer og løsninger

En utfordring var forståelsen og strukturen av src/-mappen og bruk av Jupyter Notebooks. Vi endte opp med å ikke bruke notebooks, selv om det ville vært nyttig til utforskende analyse og utvikling av enkeltstående funksjoner. I stedet fokuserte vi tungt på å bygge en helhetlig GUI-applikasjon i Tkinter, noe som ble vårt hovedfokus og kanskje dro noe av fokuset bort fra dataanalyse i starten. Vi løste dette ved å strukturere koden vår i større funksjoner, men i etterkant ser vi at det kunne vært bedre å utvikle mer gjenbrukbare og modulære funksjoner underveis.

En annen utfordring var mangel på planlegging rundt hvilke data vi ønsket å hente og analysere. Vi startet noe for raskt med kode uten å ha satt tydelige mål for analysen. Dette har lært oss viktigheten av å forstå både datagrunnlaget og mål for visualiseringen før utviklingen starter.

En stor utfordring i starten var å finne gode API-er som ga den informasjonen vi var ute etter. Gruppen fokuserte på å finne pålitelige kilder, som selvsagt er bra. Utfordringen her var at det var få eller ingen utliggere. Gruppen valgte likevel å lage en funksjon som kunne teste dette på temperaturdata, slik som vi har lært i faget. I tillegg ble en rensing funksjon laget som interpolerer om det var mangelfull data i filen vi hentet. I denne filen var det en slik mangel, og interpoleringen fungerte bra. Dette ga verdifull læring i datasensitiv behandling og hvordan valg påvirker analysens pålitelighet.

### **Samarbeid**

Samarbeidet i gruppa fungerte stort sett godt. Kommunikasjonen var det største problemet på grunn av vår manglende erfaring med Git og branches. Dette gjorde at vi i starten ikke fikk utnyttet felles utvikling fullt ut. Som nevnt tidligere førte dette til at gruppen jobbet i flere dager på en ting uten å oppdatere i main. Etter hvert som vi ble tryggere på Git, forbedret dette seg, og vi klarte å jobbe mer parallelt.

### **Sluttresultat**

Resultatmessig har vi levert en fungerende applikasjon med interaktiv meny og ulike analysefunksjoner. Visualiseringene våre ble spesielt gode med hjelp fra åpne biblioteker som Plotly og Matplotlib, og vi fikk erfaring med både statiske og dynamiske visualiseringer. Koden vår er godt strukturert, men vi ser forbedringspotensial når det gjelder dokumentasjon og gjenbruk av funksjoner.

### **Forbedringspotensialer og fremtiden**

For fremtidige prosjekter vil vi:

- Flere enhetstester
- Bruke notebooks mer aktivt i starten. Src for mer generelle funksjoner.
- Planlegge strukturen bedre.
- Lage generelle funksjoner og gjenbruke mer kode.
- Bruke Git jevnlig og strukturert

Vi ser også hvordan disse ferdighetene er relevante i arbeidslivet. Det å kunne grunnleggende programmering, og ikke minst kunne forstå og kommunisere med versjonskontroll og åpne biblioteker, vil være nyttig i mange bransjer, også utenfor rene IT-jobber.

### **Oppsummering**

Dette prosjektet har gitt oss innsikt i større programmeringsprosjekter: fra innsamling og behandling til visualisering og distribusjon. Vi har lært tekniske ferdigheter, samarbeidsrutiner og lært verktøy som Git og Plotly. Viktigst av alt har vi opplevd hvordan programmering og datavitenskap kan brukes for å forstå og formidle viktige miljødata. Dette tar vi med oss videre i både studier og yrkesliv.