

Refleksjonsnotat

Datainnsamling og databehandling

Vi benyttet eksempelkode i Frost sin API-dokumentasjon til å hente inn dataene i dataframes, og det samme gjorde vi for SSB sine data. Deretter trimmet vi bort kolonner vi anså som ikke-relevante. I tillegg sorterte vi ut data som var over en gitt kvalitet fra kolonnen qualityCode, basert på informasjon om kvalitetsparameteren fra Frost.

Dataanalyse og visualisering

Vi bearbeidet datasettet vårt ved hjelp av Pandas og Numpy slik at det ble lettere å håndtere. For statistiske beregninger brukte vi Numpy og regnet ut gjennomsnittet, median og standardavvik. Vi fremstilte noe av statistikken ved bruk av Matplotlib, og Seaborn. I regresjonsanalysen brukte vi Sklearn til en modell som trener på 100 % av testdatasettet på værddata og for utslipp brukte vi en modell som trener 50 % og en på 100% på testdatasettet. Vi beregnet også fremtids prediksjon, ved å trene regresjonsmodellen på historisk data og predikere fremtiden. Til slutt håndterte vi manglende verdier ved to ulike metoder. Mer informasjon om dette står i datakvaliteten og manglende verdier. Regresjonen ble fremstilt ved bruk av Matplotlib, Seaborn, Widgets og Plotly.

2. Tilegnede ferdigheter

Nye tekniske ferdigheter

Tidligere hadde ingen av oss noe særlig erfaring med Github, men etter dette prosjektet ser vi verdien av å bruke litt ekstra tid på oppsettet i starten i bytte mot å kunne jobbe sammen så sømløst som Github åpner for. I tillegg brukte vi ofte kommandolinja til å installere pakker, pushe og pulle fra Github og oppdatere filer. Kommandolinja er et utrolig nyttig verktøy, og dette ble en fin introduksjon.

Programmeringsforståelse

I datainnhentingsdelen av prosjektet lærte vi om databehandling i Pandas, vi lærte å bruke list comprehension, noen avanserte pandasfunksjoner og litt om bitwise-operatorer. Spesielt å kunne Pandas virker viktig, da vi etter hvert har tilgang på enormt store mengder data gjennom internett, og det å faktisk kunne lese og bruke dataene er vanskelig uten en viss mengde kunnskap om verktøyene som trengs for å håndtere store mengder data. I tillegg lærte vi litt om iteratorer i form av iterrows, som virker veldig nyttig når man skal jobbe seg gjennom store mengder data uten at det skal koste for mye av minnet til datamaskinen.

Datakvalitet og manglende verdier

I prediksjonsanalysen bruker vi nærliggende data for å tilnærme manglende data på værdaten og i utslippsdataen var det ikke manglende verdier så vi lagde funksjoner som genererer Nan verdier. Vi brukte så lineær regresjon til å legge inn manglende verdier og fremstille dette.

Tekniske og praktiske utfordringer

I starten slet vi med Github fordi vi aldri hadde brukt det før, men med litt feiling og prøving kom vi omsider i gang.

4. Samarbeid og prosjektledelse

Gruppeorganisering og kommunikasjon

Vi delte oppgavene jevnt med hverandre. Vi distribuerte det slik at alle på gruppa fikk både jobbet med å behandle og fremstillet data.

Lærdom fra gruppedynamikk

Vi lærte mye om viktigheten av god kommunikasjon. I tidligere gruppeprosjekter har vi delt inn arbeidet sånn at man aldri egentlig jobber sammen, man jobber bare mot samme felles mål. Her, ved hjelp av versjonshåndtering, jobbet vi ofte i samme kodefil. Dette kan bli noe trøblete tidvis, og vi merket tydelig forskjell på når vi snakket sammen om hvordan vi så for oss strukturen på koden før vi faktisk skrev den. I begynnelsen av prosjektet jobbet vi uten noe klar plan, og resultatet ble som ventet. Når vi senere snakket sammen og planla, merket vi at det gikk både raskere, og koden ble bedre.

5. Evaluering av resultatene

Alt i alt er vi fornøyde med resultatene. Vi mener selv at vi henter inn dataene på en effektiv måte, og behandler de på en sånn måte at vi får vist fram det vi ønsket å vise.

6. Forbedringsforslag

I fremstilling av utslipp støtte vi på et problem. Når vi kaller på en funksjon for å hente inn data til en annen funksjon, blir også den første funksjonen plotet. Dette problemet oppstår på grunn av at Jupiter notebok automatisk viser grafiske objekter for funksjoner som blir kalt på.

I tillegg skulle vi vært flinkere til å bruke branching underveis. Vi endte opp med å kun branche en gang. I tillegg skulle vi brukt mer tags.

Vi ønsket også å gjøre Utslipp mer interaktiv ved bruk av widgets slik at vi for eksempel kan se verdier når vi holder musa over.

7. Videre utvikling

Mulige utvidelser

En mulighet vil være å hente inn flere datakilder, og da synes vi det hadde vært spennende å se på luftkvalitet. Og da gjerne luftkvalitet i forhold til temperatur eller CO2-utslipp.

8. Oppsummering

Vi kan nå bruke github på en relativt god måte, som gjør det lettere å samarbeide i fremtidige programmeringsprosjekter som er veldig relevant som studier og senere jobb.