Prognotisering av Bergen BySykkel

Av Sverre Kristian Thune

Innhald

1 Innleiing

Bysyklar er eit miljøvenleg og effektivt alternativ til bil og kollektivtransport, og gjer det mogleg for innbyggjarar å kome seg rask rundt i byen. I Bergen vart Noregs første faste, heilårs bysykkelordning til i juli 2018 med 76 stasjonar og 400 syklar. «Bergen Bysykkel har vist seg å vere enormt populært med over 1 000 000 turar i året fordelt på rundt 40 000 bysyklistar» (UIP, u.d.).

Med ein slik etterspørsel for bysyklar, med avgrensa tal syklar og stasjonar, oppstår utfordringar knytt til tilgjengelegheit og balanse i systemet. Trafikk varierer for kvar stasjon og enkelte stasjonar går tomme, medan andre blir fulle. Dette skapar uforutsigbarheit for brukarar, som gjerne bruker syklar under rushtid og ved høgt trafikkerte stasjonar.

Målet med dette prosjektet er å utvikle ein prediksjonsmodell som kan estimere kor mange ledige syklar som vil vere tilgjengeleg på relevante stasjonar éin time fram i tid, ved hjelp av historiske data av sykkelbruk, vêr og tidspunkt. Løysinga av eit slikt system bidreg til betre brukaroppleving for bysyklistane og ein meir effektiv drift av systemet.

1.1 Prosjektets hovuddelar

1. **Datainnsamling og klargjering**

   Data frå stasjonar, turar og vêr blir vurdert, samanstilt og rydda. Målet er å gjere om rådata til eit samla datasett, der kvar rad oppfyller eit tidspunkt der det er gjort ein observasjon av turar og vêr for kvar stasjon.

   Dette omfattar blant anna avrunding av tidsstempel til heile timar, og bruk av Last Observation Carried Forward (LOCF) for å fylle tidsrommet mellom observasjonar. One hot encoding vil bli brukt f.eks på stasjonar for å gjere data meir lesbart for modellar.

2. **Modellering**

   Det blir trena fem ulike modellar for å predikere talet på ledige syklar éin time fram i tid. Desse modellane er DummyRegressor, lasso, LightGBM, CatBoos og XGBoost, og blir vurdert og samanlikna. Modellen som oppnår lågast Root Mean Square Error (RMSE) på valideringsdata blir valt som den beste og endelege modellen.

3. **Evaluering og implementering**

   Den endelege modellen blir implementert i ein pipeline som kan køyrast automatisk på nye data. Pipelinen kan gi predikasjonar for neste heile time for alle mål-stasjonar, og skrive ut resultat i eit lesebart format.

1.2 Avgrensingar

Det vil førekome nokre avgrensingar i prosjektet:

- Predikasjonane gjeld berre dei stasjonane som kunden har spesifisert som relevante.
- Modellen predikerer berre sykkeltilgjenge éin time fram i tid, og ikkje tal på turar eller individuelle brukarar.
- Modellen er trena på tidlegare historiske data, som gjer at det er forventa at framtidige mønster i data liknar tidlegare observasjonar.

2 Datagrunnlag og klargjering

For å trena modellane må all rådata som er tildelt først ryddast og samanstillast, slik at dei ulike mønstera i data kjem tydeleg fram. Dette inneber å fjerne unødvendig informasjon, fylle ut manglande verdiar, og slå saman alle datasett til treningsdata.

2.1 Datakjelder

Rådata som er nytta i prosjektet består av tre separate datasett henta frå ulike kjelder: stations.csv, trips.csv og weather.csv. Desse datasetta inneheld til saman informasjon om sykkeltilgjenge, turaktivitet og vêrforhold i Bergen, og dannar grunnlaget for treningsdata:

1. stations.csv
   Datasettet er henta frå Bergen Bysykkel sine opne data (Bysykkel, u.d.), og inneheld sanntidsinformasjon om talet på ledige syklar ved kvar stasjon. Kvar rad er ein observasjon på eit gitt tidspunkt, med kolonnar som stasjonsnamn og talet på ledige syklar og tomme plassar. Ein kolonne å legge merke til er «skipped_updates», som teller talet på uendra observasjonar fram til ein ny observasjon.
2. trips.csv
   Datasettet er henta frå eit offentleg API utvikla av Max Halford, og som er tilgjengeleg på GitHub (Max, u.d.). Datasettet inneheld registrerte sykkelturar, med tidspunkt for avgang og ankomst frå start-stasjon til ende-stasjon.

3. weather.csv
   Datasettet er henta frå Open-Meteo, som tilbyr ein fri API med historiske vêrobservasjonar i verda. Datasettet inneheld timesdata for Bergen, med kolonnar som temperatur, nedbør og vindstyrke. Med dette kan ein undersøke korleis vêr påverkar sykkelbruk.

2.2 Prosessering

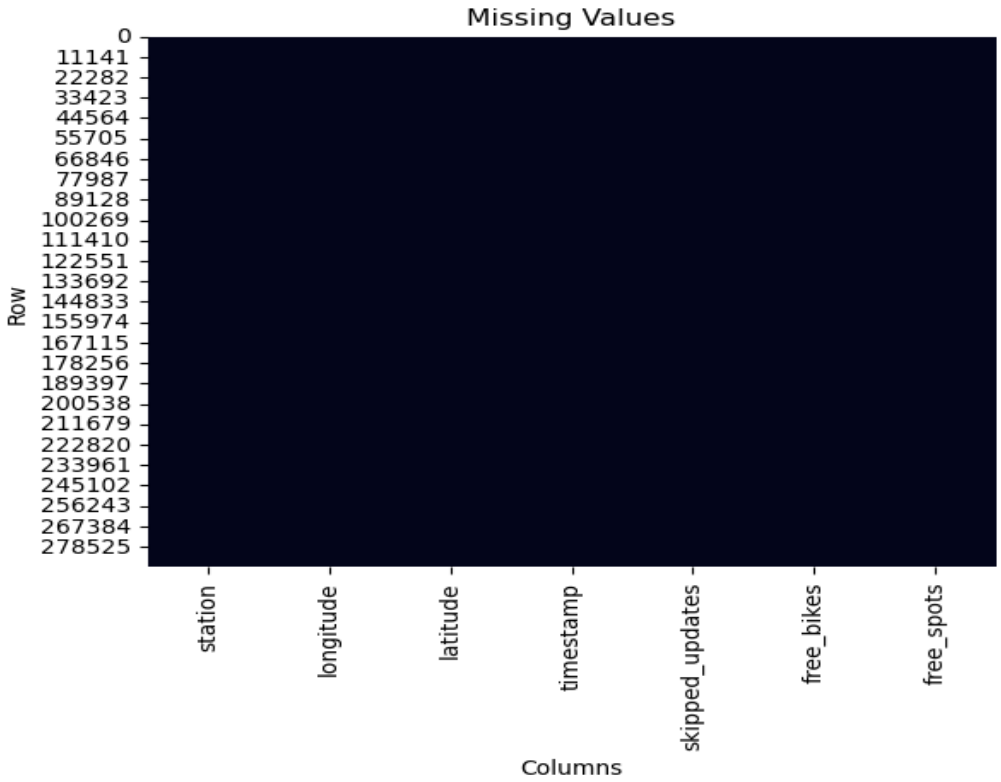
Som nemnt tidlegare vil dataprosesseringa gå føre seg slik:

1. Fjerne unødvendige features
2. Kontroll og utfylling av manglande verdiar
3. Samanslåing av datasett

Nedanfor blir det vist korleis dei tre datakjeldene er presenterte og ein kort oversikt over kolonnar, eksempelverdiar, eventuelle manglande data og korleis desse vart behandla i klargjeringa.

2.2.1 Oversikt over stations.csv

Datasettet stations.csv inneheld sanntidsinformasjon om talet på ledige syklar og ledige plassar ved kvar stasjon i Bergen. Kvar rad representerer éi observasjon frå ei stasjon på eit gitt tidspunkt. Varmekartet og tabellen under viser at datasettet er komplett, og korleis det vart rydda.



Figur 1. Varmekart over manglande verdiar i stations.csv. Figuren viser samt kolonnane og tal observasjonar i datasettet, og at datasettet er komplett

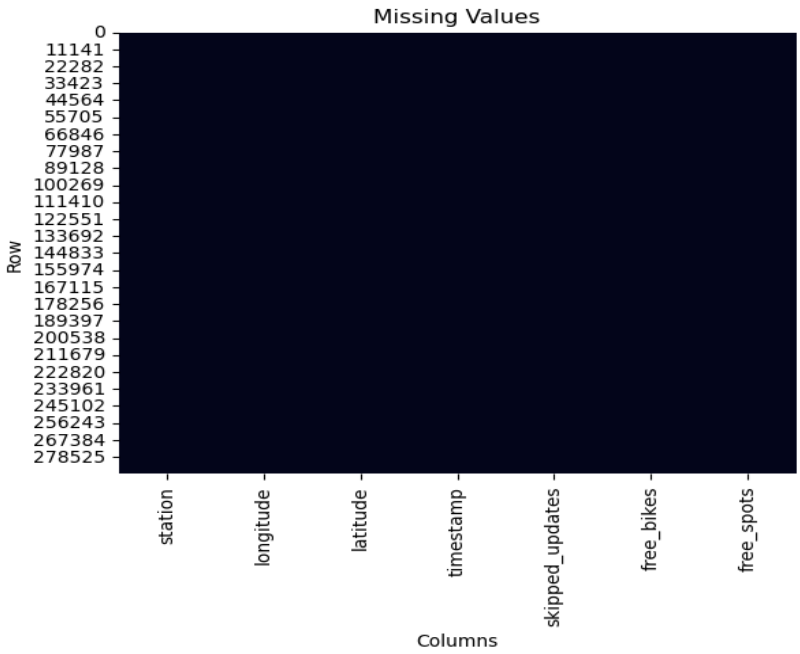
Kolonne	Eksempelverdi	Manglande verdiar	Handtering
station	Torget	ingen	Fjerna urelevante stasjonar
longitude	5.325284	ingen	Kolonne fjerna (ikkje relevant for modellering)
latitude	60.395878	ingen	Kolonne fjerna (ikkje relevant for modellering)
timestamp	2025-01-16 08:07:36+00:00	ingen	Runda ned til næraste heile time
skipped_updates	0	ingen	Kolonne fjerna
free_bikes	1	ingen	Behald kolonne
free_spots	23	ingen	Behald kolonne

Tabell 1. Tabellen viser kolonnane med eksempelverdiar, tal manglande verdiar og handtering av kolonnane.

Etter rydding vart kolonnane station, timestamp, free_bikes og free_spots vidareførte til neste steg i prosesseringa. Desse kolonnane dannar grunnlaget for å predikere kor mange syklar som er tilgjengeleg ved kvar stasjon til kvar heile time.

2.2.2 Oversikt over trips.csv

Datasettet trips.csv inneheld observasjonar av individuelle sykkelturar med start-/slutt tidspunkt og lokasjonar. Kvar rad representera éin tur frå ein startstasjon til ein endestasjon. Varmekartet og tabellen under viser at datasettet var komplett, og korleis det vart rydda.



Figur 2. Varmekart over manglande verdiar i trips.csv. Figuren viser samt kolonnane og tal observasjonar i datasettet, og at datasettet komplett.

Kolonne	Eksempelverdi	Manglande verdiar	Handtering
started_at	2023-01-01 04:22:50.614000+00:00	ingen	Runda ned til næraste heile time
ended_at	2023-01-01 04:33:19.884000+00:00	ingen	Runda ned til næraste heile time
start_station_name	Torget	ingen	Fjerna urelevante stasjonar
start_station_latitude	60.39587808663882	ingen	Kolonne fjerna (ikkje relevant for modellering)
start_station_longitude	5.325283812313046	ingen	Kolonne fjerna (ikkje relevant for modellering)
end_station_name	Takhagen på Nordnes	ingen	Fjerna urelevante stasjonar
end_station_latitude	60.39886453454994	ingen	Kolonne fjerna (ikkje relevant for modellering)
end_station_longitude	5.306410871328268	ingen	Kolonne fjerna (ikkje relevant for modellering)

Tabell 2. Tabellen viser kolonnane med eksempelverdiar, tal manglande verdiar og handtering av kolonnane.

Etter rydding vart kolonnane started_at, ended_at, start_station_name, og end_stations_name vidareført. Desse kolonnane vert seinare brukt til å få oversikt over avgangar og ankomstar per stasjonstime, som gir eit mål på trafikkflyten i systemet.

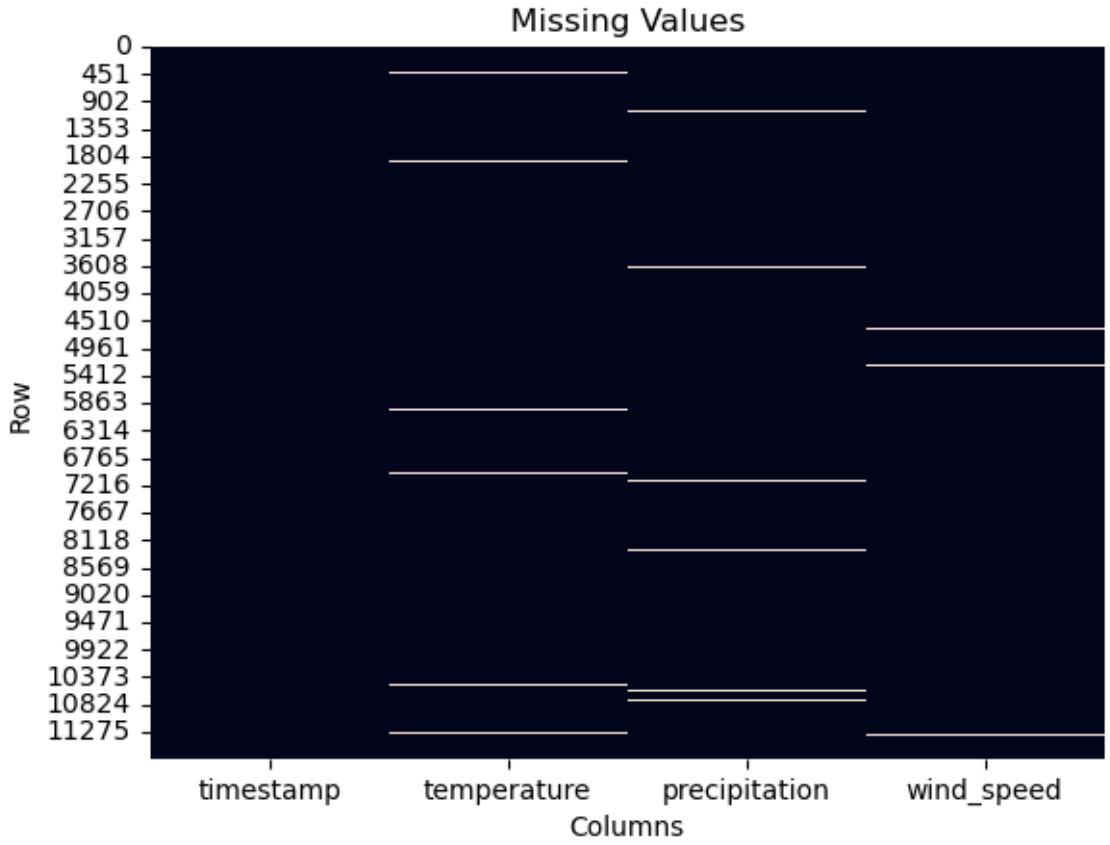
For å oppnå ein slik oversikt vart data frå trips.csv omforma ved hjelp av gruppering etter stasjonsnamn og avrunda tidspunkt. Grupperinga av start_station_name og started_at gav ein oversikt over tal avgangar, medan ankomstar vart berekna ut frå end_station_name og ended_at. Desse resultata vart deretter slått saman til eitt samla datasett med kolonnane station, timestamp, departures og arrivals. Det blir også lagt til ein ekstra kolonne «net_change», som er differansen av avgangar og ankomstar den gitte timen, altså: arrivals – departures = net_change. Datasettet trips.csv vil nå sjå slik ut:

Kolonne	Eksempelverdi	Manglande verdiar
station	Akvariet	ingen
timestamp	2024-04-11 06:00+00:00	ingen
departures	2	ingen
arrivals	1	ingen
net_change	1	ingen

Tabell 3. Tabellen viser dei nye kolonnane med eksempelverdiar og tal manglande verdiar

2.2.3 Oversikt over weather.csv

Datasettet weather.csv inneheld vêrmålingar for Bergen-området, med kolonnar som tid, temperatur, nedbør og vindstyrke. Kvart datapunkt er ei måling per heile time, med temperatur gitt i grader Celsius, regn og sludd i millimeter og vindhastigheit i m/s. Varmekartet og tabellen under viser at datasettet inneheld manglande verdiar, og korleis datasettet vart rydda og verdiane behandla.



Figur 3. Varmekart over manglande verdiar i trips.csv. Figuren viser samt kolonnane og tal observasjonar i datasettet, og at datasettet inneheld manglande verdiar.

Kolonne	Eksempelverdi	Manglande verdiar	Handtering
timestamp	2023-12-31 23:00:00+00:00	ingen	Timesoppløysing – beholdt uendra
temperature	3.0	132	Fylt manglande verdiar med gjennomsnitt (Simpleimputer, mean-strategi)
precipitation	0.0	108	Fylt manglande verdiar med gjennomsnitt
wind_speed	17.9	110	Fylt manglande verdiar med gjennomsnitt

Tabell 4. Tabellen viser kolonnane med eksempelverdiar, tal manglande verdiar og handtering av kolonnane.

Etter rydding i datasettet vart alle kolonnar rekna som relevante og vidareførte til neste steg i prosesseringa. Sidan observasjonane er timesbaserte og manglande verdiar vart fylt med middelverdiar frå nærliggande timar, er det rimeleg å anta at dei imputerte verdiane er realistiske i kontekst av datamaterialet.

2.2.4 Last Observation Carried Forward (LOCF)

For å sikre at vi har observasjonar for sykkeldata for kvar heile time, vart metoden Last Observation Carried Forward (LOCF) nytta på datasettet stations.csv. Denne metoden fører vidare den siste kjende observasjonen fram til neste registrerte måling, slik at eventuelle tidsrom utan oppdateringar blir fylt med realistiske verdiar.

Tabellane under viser eit eksempel frå stasjonen Akvariet før og etter LOCF vart nytta. Kolonnen skipped_updates viser at enkelte oppdateringar manglar mellom tidsstempel, noko som gjer at fleire timar ikkje har registrerte verdiar.

Før LOCF:

station	timestmap	skipped_updates	free_bikes	free_spots
Akvariet	2025-04-25 16:00:00+00:00	0	1	17
Akvariet	2025-04-25 19:00:00+00:00	2	3	15

Etter LOCF:

station	timestmap	skipped_updates	free_bikes	free_spots
Akvariet	2025-04-25 16:00:00+00:00	0	1	17
Akvariet	2025-04-25 17:00:00+00:00	0	1	17
Akvariet	2025-04-25 18:00:00+00:00	0	1	17
Akvariet	2025-04-25 19:00:00+00:00	2	3	15

Det finnes no komplette observasjonar for kvar time, der verdiane mellom to faktiske målingar er fylt med den sist kjende observasjonen.

2.2.5 Oppsummering av datasett

Resultatet av kontrollen viste manglande verdiar i weather.csv og ufullstendige observasjonar i stations.csv, dokumentert av kolonnen skipped_updates. Metoden Last Observation Carried Forward (LOCF) vart nytta for å sikre komplette timesvis observasjonar i staitons.csv, medan gjennomsnittsimputering vart brukt i weather.csv for å fylle manglande verdiar. Etter rydding og utfylling av manglande data er alle dei tre datasetta heile og klare til vidare samanslåing, som kan gi eit fullstendig bilete av korleis sykkelbruk påverkast av vêr og tid ved kvar stasjon.

2.3 Samanslåing av datasett

For å trene maskinlæringsmodellar er det avgjerande å ha eit fullstendig og strukturert datasett som samlar all relevant informasjon. Dette inneber å setje saman tildelte datasett og rydde bort eller dele opp kolonnar med for mykje generell informasjon. Ei slik prosess gjer det enklare for modellane å identifisere mønster i data, og legg samtidig til rette for meir presis analyse og tolking av samanhengar mellom variablar.

2.3.1 Samanslåing av stations.csv, trips.csv og weather.csv

Ein viktig observasjon er at stations.csv og trips.csv har observasjonar med ulik tidsperiode. Første registrering i trips.csv er frå 01.01.2023, medan stations.csv først startar 10.04.2024, om lag eitt år seinare. For å gjere datasetta samanliknbare, vart samanslåinga gjort med omsyn til tidsrommet som er dekt av stations.csv, slik at berre overlappinga i denne tidsperioden vart brukt vidare.

Sidan trips.csv ikkje har registrerte turar for kvar einaste time, oppstod det manglande verdiar i kolonnane arrivals og departures etter samanslåinga. Dette vart erstatta med verdien 0, ettersom eit manglande tal på turar naturleg kan tolkast som ingen registrerte avgangar eller ankomstar på den aktuelle timen.

Etter samanslåinga av stations.csv og trips.csv vart det oppretta eit samla datasett, df_model_ready, som inneheld kolonnane station, timestamp, free_bikes, free_spots, arrivals og departures. Deretter vart weather.csv lagt til ved å slå saman datasetta på kolonnen timestamp, slik at kvar observasjon fekk tilhøyrande vêrdata. Det samla datasettet, df_model_ready, har nå radar som representerer observasjonar av sykkeldata, turaktivitet og vêrforhold for kvar time, og ved kvar stasjon.

Ein siste ulempe er at enkelte timar ikkje hadde tilhøyrande vêrobservasjonar, så det oppstod nokre manglande verdiar i kolonnane temperature, precipitation og wind_speed. Desse vart utfylt ved hjelp av lineær interpolering, som estimerar mellomliggjande verdiar baser på nærliggjande observasjonar, og vidare supplert med framfylling og bakfylling (forwardfill og backwardfill). Denne metoden sørgjer for at manglande verdiar heilt i starten og slutten av datasettet, som ikkje kan bli interpolerte, også får utfylt verdiar dersom desse manglar.

Gjennom denne prosessen er df_model_ready sikra kontinuitet i data og risikoen for datalekkasje mellom timar er minimalisert. Datasettet er no fullstendig og modellklart og inneheld alle nødvendige observasjonar og viktige variablar. Men datasettet kan framleis gjerast meir presist.

2.3.2 Oppdeling av timestamp i df_model_ready

Kolonnen timestamp i df_model_ready inneheld fleire typar informasjon, som år, månad, dag og klokkeslett. For å gjere data meir presise og enklare å tolke for modellane, vart denne kolonnen delt opp i fleire separate variablar. Dette gjer det mogleg for modellane å fange opp tidsbaserte mønster som endringar over døgnet, gjennom veka og mellom ulike månader.

Frå «timestamp» vart følgjande kolonnar oppretta:

- year – året observasjonen er registrert
- month – månaden (1 – 12)
- weekday – dag i veka (0 = måndag, 6 = søndag)
- hour – klokkeslett (0 – 23)

I tillegg vart det oppretta seks binære variablar basert på desse tidsverdiane:

- is_winter– 1 dersom observasjonen fell i vintersesongen (des – feb), 0 elles
- is_spring– 1 dersom observasjonen fell i vårsesongen (mars – mai), 0 elles
  is_summer– 1 dersom observasjonen fell i sommarsesongen (jun – aug), 0 elles
- is_autumn – 1 dersom observasjonen fell i høstsesongen (sep – nov), 0 elles
- is_freeday – 1 dersom observasjonen fell på helg eller norsk heilagdag, 0 elles
- is_rush_hour – 1 dersom observasjonen fell innanfor definerte rushtider (07 – 09 og 15 – 17), 0 elles

Desse featursa gjer informasjonen meir presis og gir modellen sjansen til å forstå korleis sykkeltilgjenge varierer mellom kvardagar og fridagar, samt korleis trafikkmønster endrar seg i rushtida.

2.3.3 Oppretting av målvariabel

For at modellen skal vite kva den skal predikere, vart det oppretta ei ny kolonne med namn «free_bikes_next». Denne kolonnen representerer talet på ledige syklar ved kvar stasjon éin time fram i tid, og fungerer dermed som målvariabel i modelleringa.

Kolonnen «free_Bikes_next» vert brukt både til å trene modellen i å predikere framtidige sykkeltilgjenge, og til å evaluere kor godt modellen klarar å treffe faktiske observasjonar. På denne måten kan ein samanlikne modellens predikasjonar med reelle data for å berekne ut nøyaktigheit og feilrate (RMSE), som blir nytta til å velje beste modell.

Måten kolonnen vart oppretta på, var ved å lage ein forskyvd kopi av kolonnen «free_bikes» innanfor kvar stasjon. Ved å flytte verdiane éin time opp i datasettet, vil kvar rad representere både observasjonen av sykkeltilgjenge på det aktuelle tidspunktet, og den faktiske verdien éin time fram i tid for same stasjon.

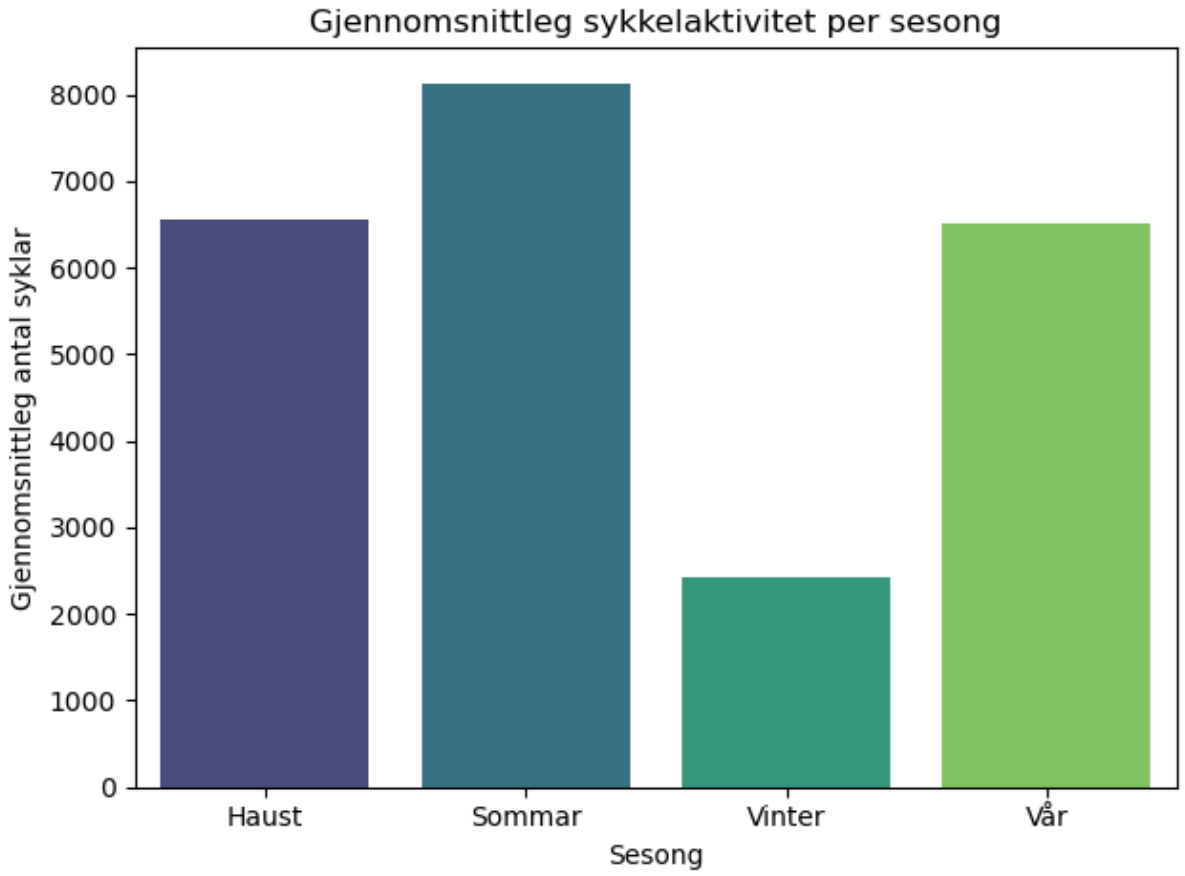
For å få til dette vart det nytta tidsforskyving per stasjon med funksjonane groupby og shift. Sida det blir gruppert etter stasjonar vil funksjonen shift, altså tidsforskyvinga, berre hende innanfor kvar stasjon si eiga tidsrekkje. Ein konsekvens av å gjere «shift» på data direkte, er at slutten på éin stasjon sin tidsrekkje ville fått neste rad frå ein annan stasjon som framtidsverdi. For eksempel hadde siste registrering av stasjonen «Florida Bybanestopp» fått første observasjon av ledige syklar frå stasjonen «Torget» som sin framtidsverdi. Dette fører til datalekkasje.

Ei ulempe med denne metoden er at siste observasjon i tidsrekkja til kvar stasjon ikkje vil ha framtidsverdi. Dette blir løyst med forward fill (ffill), der den siste gyldige verdien blir vidareført slik at datasettet held seg komplett.

3 Utforskande dataanalyse (EDA)

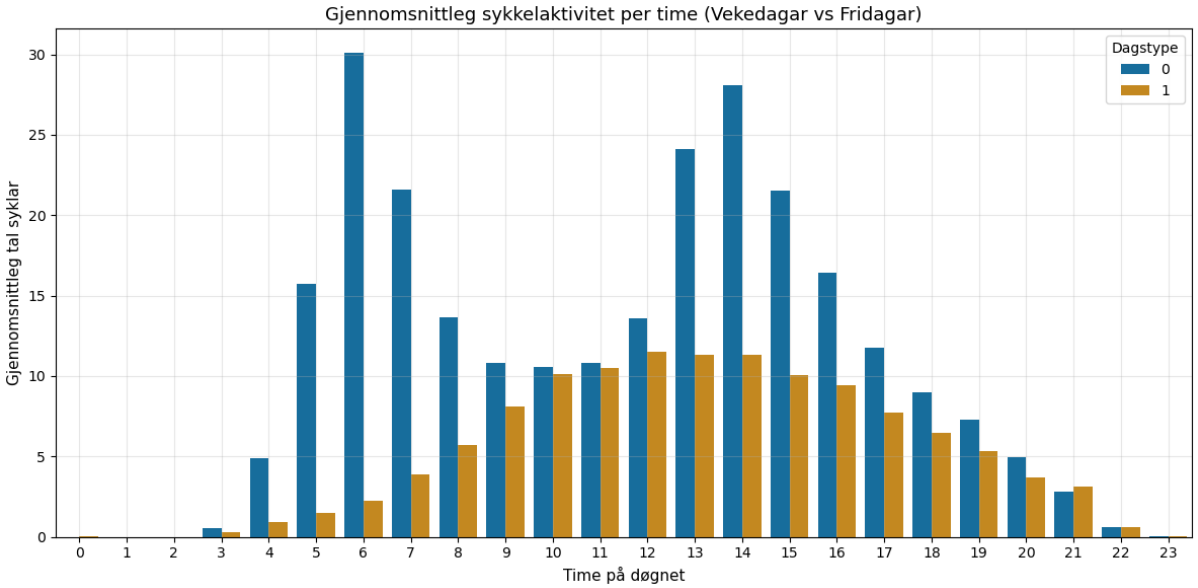
For å forstå datasettet er det nødvendig med ein utforskande dataanalyse (Exploratory Data Analysis). Dette hjelper med å forstå strukturen, oppdage potensielle samanhengar mellom variablar samt, tydlegare sjå uteliggjarar. Analysen kan hjelpe med å identifisera problem i data før vi begynner å modellere (Utforskende dataanalyse, 2025).

3.1.1 Sykkelaktivitet vs sesong



Figur 4. Figuren viser gjennomsnittleg sykkelaktivitet per sesong. Frå figuren kan ein sjå at sykkelbruken er mest populær i sommar sesongen (juni – august) og lik i haust og vår sesongen.

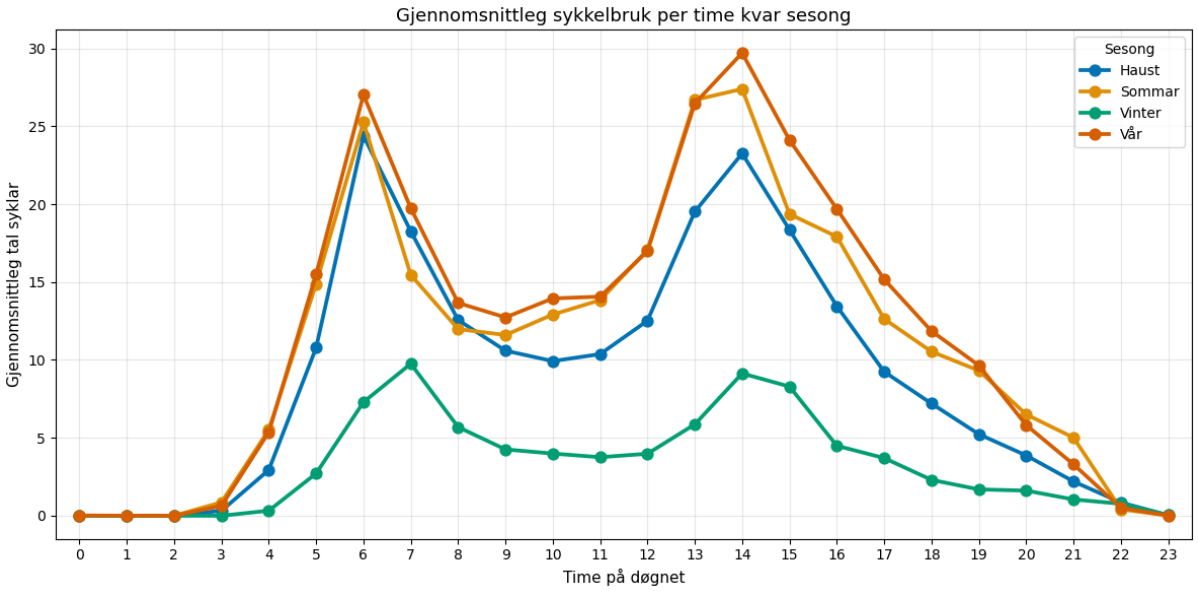
3.1.2 Sykkelaktivitet kvar time (vekedag vs fridag)



Figur 5. Figuren viser gjennomsnittleg sykkelaktivitet kvar time i vekedagar og helge- og heilagdagar. Frå figuren ser ein eit tydleg rushtidmønster, der aktiviteten er størst ved klokkesletta 05:00-08:00 og 13:00-17:00. I fridagar er det eit

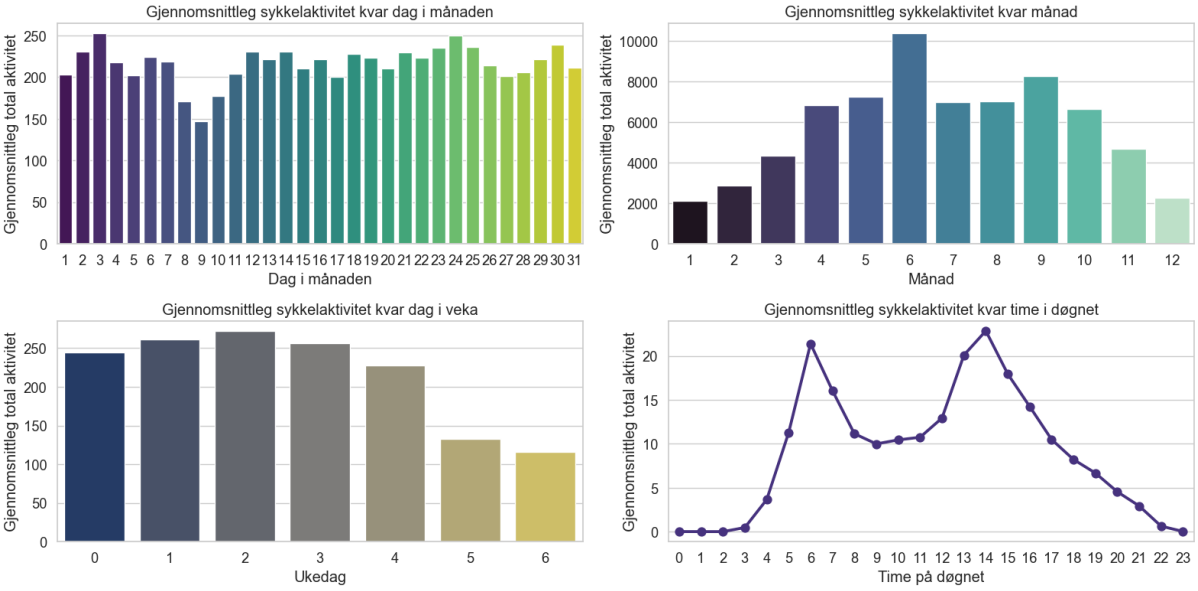
jamnare mønster der aktiviteten aukar om morgonen, er på sitt største om ettermiddagen og går jamt ned mot kvelden.

3.1.3 Sykkelaktivitet kvar time (sesong)



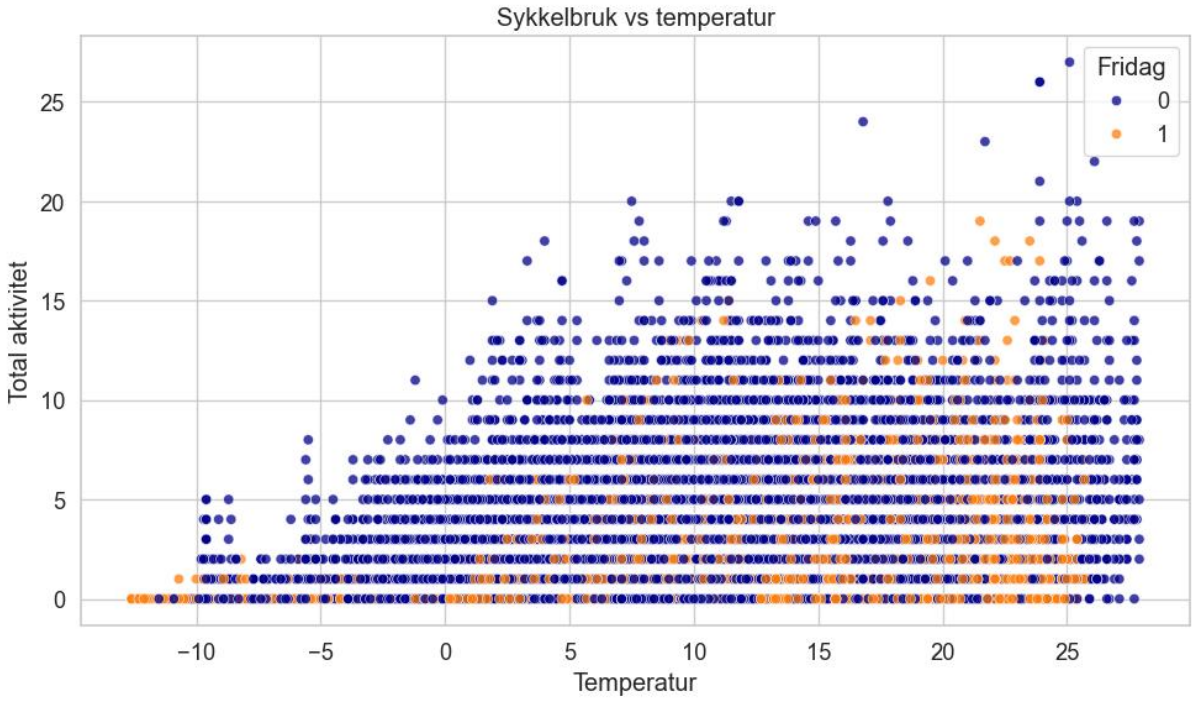
Figur 6. Figuren viser gjennomsnittleg sykkelaktivitet kvar time kvar dag i kvar sesong. Frå figuren ser ein det klassiske rushtidmønsteret i alle sesongar, samt ein liknande gjennomsnittleg sykkelbruk om vår, sommar og haust, og ein tydeleg mindre bruk om vinteren.

3.1.4 Sykkelaktivitet gjennomsnittleg månad, veke, time

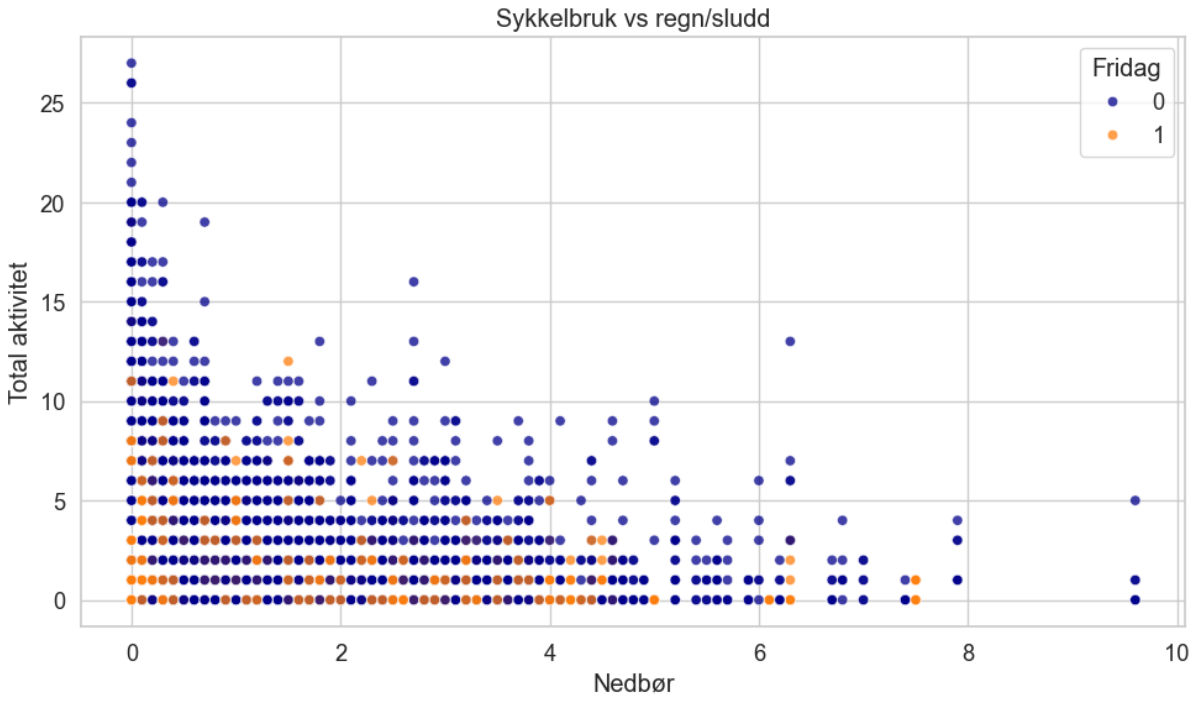


Figur 7. Figuren viser fire diagram: Øverst til venstre viser gjennomsnittleg sykkelaktivitet kvar dag i ein månad, og ved sida av, gjennomsnittleg sykkelaktivitet ved kvar månad i året. Under til venstre viser gjennomsnittleg sykkelaktivitet kvar dag i veka, og ved sida av, gjennomsnittleg sykkelaktivitet kvar time i døgnet.

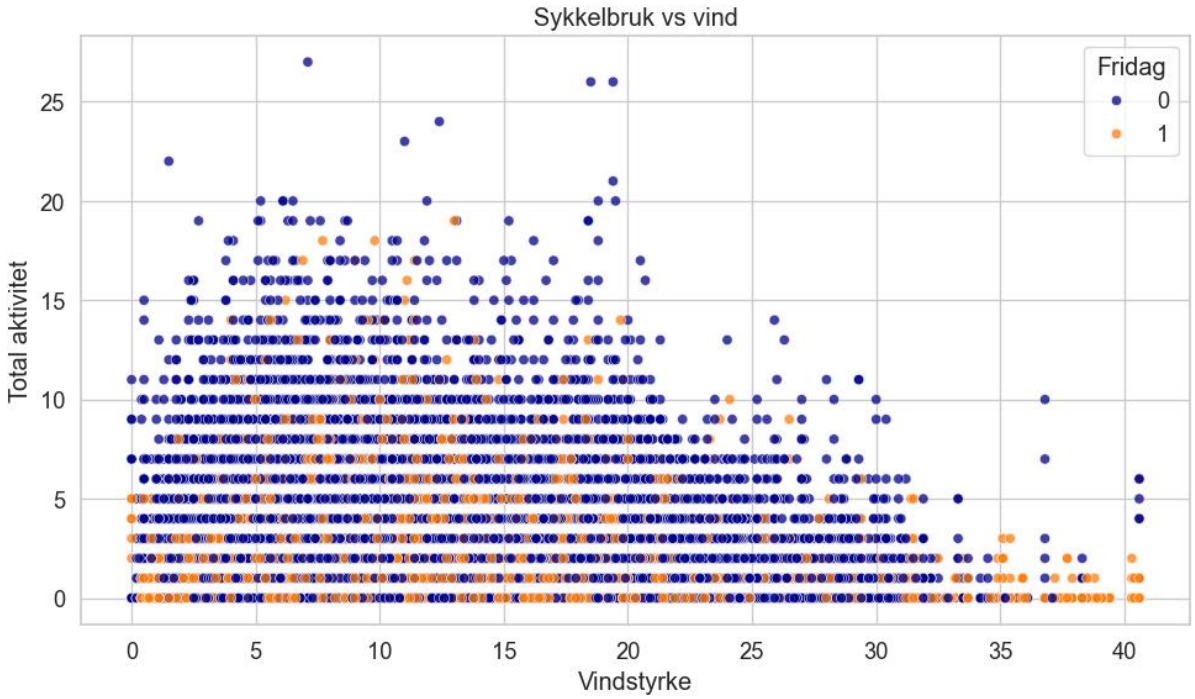
3.1.5 Sykkelaktivitet vs temperatur, regn/sludd, vind



Figur 8. Figuren viser sykkelaktiviteten på vekedagar og fridagar basert på lufttemperaturen. Frå figuren ser ein at aktiviteten aukar med stigande temperatur.

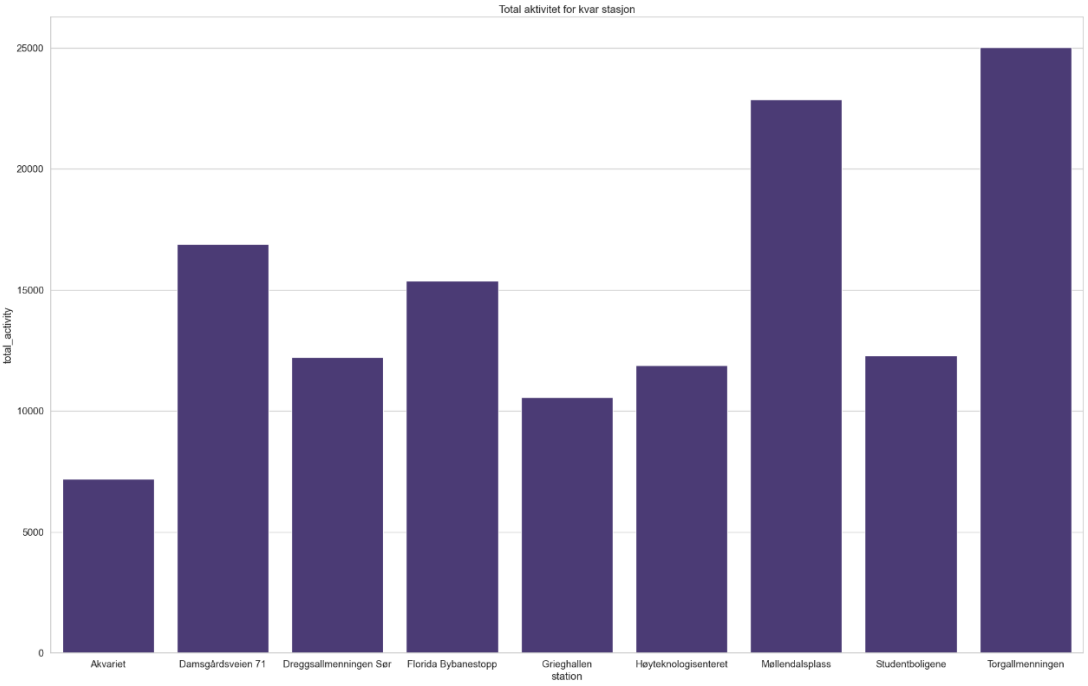


Figur 9. Figuren viser sykkelaktiviteten på vekedagar og fridagar basert på nedbør. Frå figuren ser ein at aktiviteten minkar med aukande nedbør.



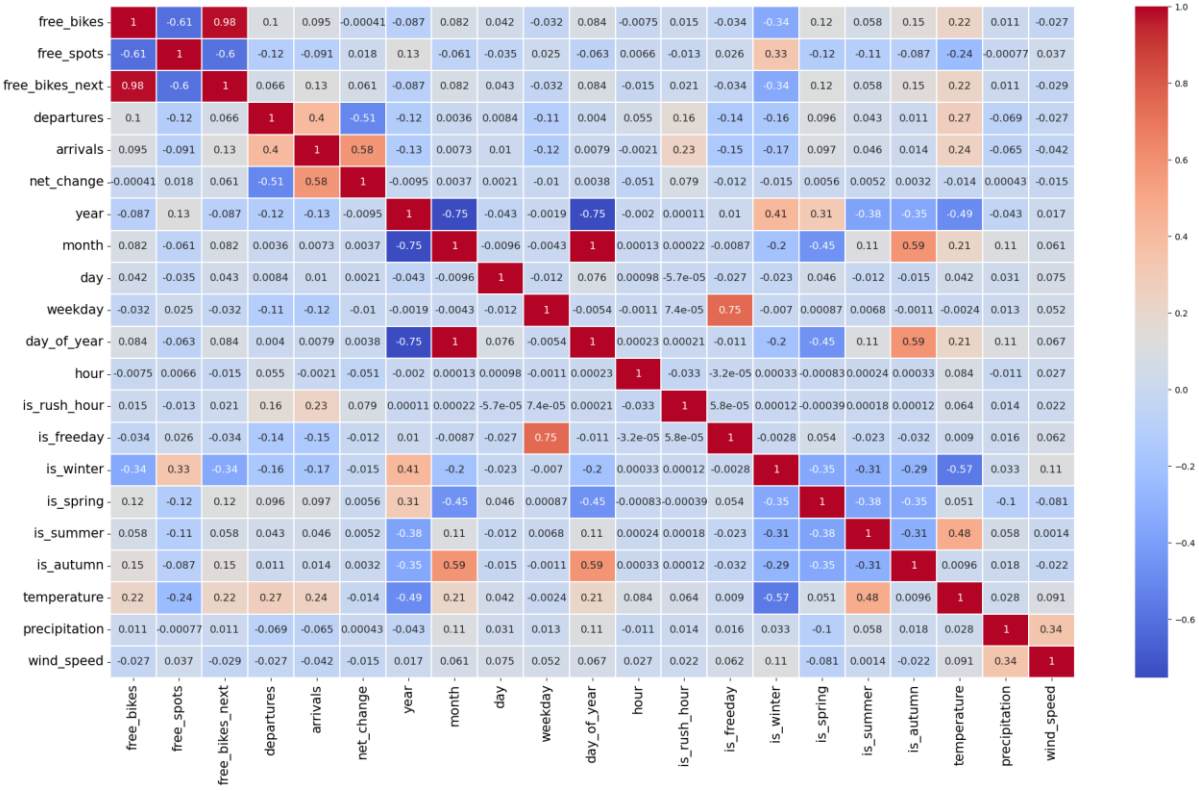
Figur 10. Figuren viser sykkelaktiviteten basert på vindstyrke. Frå figuren ser ein at aktiviteten minkar sakte med aukande vindhastigheit. Noko viktig å leggje merke til er dei store verdiane for vindhastigheit, noko som kan vere feil i målingane.

3.1.6 Travlaste stasjon



Figur 11. Figuren viser total sykkelaktivitet for kvar stasjon i analysen. Frå figuren ser ein at stasjonen "Torgallmenningen" er den som opplever mest trafikk.

3.1.7 Korrelasjonar mellom variablar



Figur 12. Figuren viser ein oversikt over korrelasjonar mellom dei numeriske variablane. Den sterkaste samanhengen er mellom free_bikes og free_spots, som gir meining då dei summert er kapasiteten til kvar stasjon. Temperatur har ein moderat positiv korrelasjon til sykkelaktivitet, sett nedst i venstre hjørne. Nedbør og vind viser ein negativ samanheng, noko som vi også har sett i tidlegare figurar.

3.2 Oppsumering EDA

Frå den utforskande dataanalysen eksisterer det tydelege mønster ein kan ta med vidare i feature engineeringa. Dette er typiske mønster som sesongvariasjonar, døgnrytme, rushtid, og aktivitet ved ulike vêrforhold. Frå analysen veit vi kva variablar som er viktig og kven av dei vi eventuelt kan omforme eller fjerne.

4 Feature Engineering og Modellering

Feature engineering er viktig for å gjere klar data til modellering. Dette inneber å konstruere nye variablar frå eksisterande informasjon for å gjere det lettare for modellen å oppdage mønster og samanhengar i data. I df_model_ready vart det allereie gjort ein del feature engineering då kolonnen «timestamp» vart dele opp i mindre einingar. Dette var for å gjere det lettare å visualisere data i EDA delen. Her går det framleis an å opprette fleire lag-features som hjelper modellane å fange opp endringar over tid.

Når denne feature engineeringen er gjort blir ulike maskinlæringsmodallar trena for å predikere talet på ledige syklar éin time fram i tid. Målet er å finne den modellen som gjev lågast feilrate målt ved Root Mean Square Error (RMSE).

4.1 Feature engineering

4.1.1 Sinus og Cosinus verdiar for tidsdata

Sidan dataen er kontinuerleg 24/7, er det viktig at modellen forstår dei sykliske mønstra i tiden. For å oppnå dette vart tidsvariablar som time, dag og månad omgjorde til sinus og cosinusverdiar. Denne representasjonen av tid gjer det mogleg for modellen å oppfatte at for eksempel tidspunktet 23:59 ligg ganske nær 00:00 (FRANCO12, 2019).

Den same logikken gjeld for dagar og månader gjennom året. Modellen kan lettare fange opp sesongvariasjonar og overgangar mellom årstider og veker, som sett i den utforskande dataanalysen, påverkar sykkelbruken og sykkeltilgjenge. Tidsdata blir på denne måten meir realistisk for modellen.

Det blir oppretta sinus og cosinus verdiar for følgane variablar:

- «hour» -> «hour_sin» og «hour_cos»
- «weekday» -> «day_sin» og «day_cos»
- «day_of_year» -> «day_of_year_sin» og «day_of_year_cos»

Dermed blir de tidlegare variablane fjerna for å minske støy for modellane. I tillegg vert kolonnane, «day», «month», «year» og «date» fjerna. Sinus og cosinus variablane er bra nok for å sjå desse sykliske mønstra og reduserer dermed også støy. Desse variablane vil også gi ein veldig lik korrelasjonen med de tidlegare variablane, så då er det best å berre ekskludere de.

4.1.2 Dummy variablar for stations

Den einaste ikkje-numeriske variabelen i df_model_ready er «station». For å kunne bruke denne informasjonen i numeriske modellar vart variabelen omgjort til dummyvariablar. Dette gjer at modellen kan fange opp korleis talet på ledige syklar varierer mellom dei ulike stasjonane, basert på samanhengen mellom dei andre variablane.

Sidan datasettet består av ni utvalde stasjonar, vart det oppretta ni nye kolonnar i datasettet, altså éin for kvar stasjon. Desse fekk namn som «station_Akvariet», «station_Torget», «station_Florida_Bybanestopp» osv. Observasjonen får verdien 1 dersom dataen høyrer til den aktuelle stasjonen, elles 0. Med desse dummy variablane kan modellen lære stasjonsspesifikke mønster, som for eksempel kva stasjonar som blir mest påverka av rushtid, eller kva stasjonar som står fulle eller tomme om morgonen. (FRANCO12, 2019)

4.2 Val av modellar

I analysen blir det trena opp seks maskinlæringsmodellar. Desse modellane dekker både lineære og ikkje-linære samanhengar:

1. DummyRegressor:
   Baseline modell som alltid predikerer gjennomsnitt av treningsdata
2. Lasso:
   Ein lineær modell som straffar uteliggjarar for å hindre overfitting, og velje ut dei mest relevante variablane.
3. LightgbmRegressor:
   Ein tre-modell som er rask og effektiv på store datasett og som handterer ikkje-lineære samanhengar bra.
4. CatBoostRegressor:
   Ein modell optimalisert for kategoriske data og krev derfor lite feature engineering. Den er svært robust mot overfitting.
5. XGBRegressor:
   Ein modell også optimalisert for kategoriske data, og gir svært høg predikasjonspresisjon.

4.3 Trening og validering

For å vurdere kor godt modellane tilpassar seg ny data, vart datasettet delt i tre delar: 70% trening, 15% validering og 15% testing. Denne fordelinga gjer at det er mogleg å trene modellen på mesteparten av dataen. Samtidig har vi to uavhengige datasett til å justere hyperparametrar og teste feilrate på beste og endelege modell.

Etter at modellen er tilpassa treningsdataen og modellen med de beste hyperparametrane er funne, blir valideringsdatasette nytta til å evaluere predikasjonsevna. Modellen som får lågast feilrate (RMSE) på valideringsdata vert nytta som den endelege modellen.

4.4 Justering av hyperparametrar

Målet med å justere hyperparametrar til maskinlæringsmodellar er å finne dei mest optimale verdiane som effektiviserer og påverkar modellens evne til å prestere betre. I denne analyse er det valt å nytte GridSearchCV til dette. GridSearchCV er ein brute-force teknikk for justering av hyperparametre, som trener modellen med alle gitte kombinasjonar av spesifiserte hyperparametrar for å finne beste modell. For kvar kombinasjon av parametrar vart modellen trena og evaluert ved hjelp av 5-fold kryssvalidering, og kvar modell vart evaluert med feilrate (RMSE). Modellen med lågast RMSE vart den endelege modellen, som blir vidare evaluert med valideringsdatasettet (Hyperparameter Tuning, 2025).

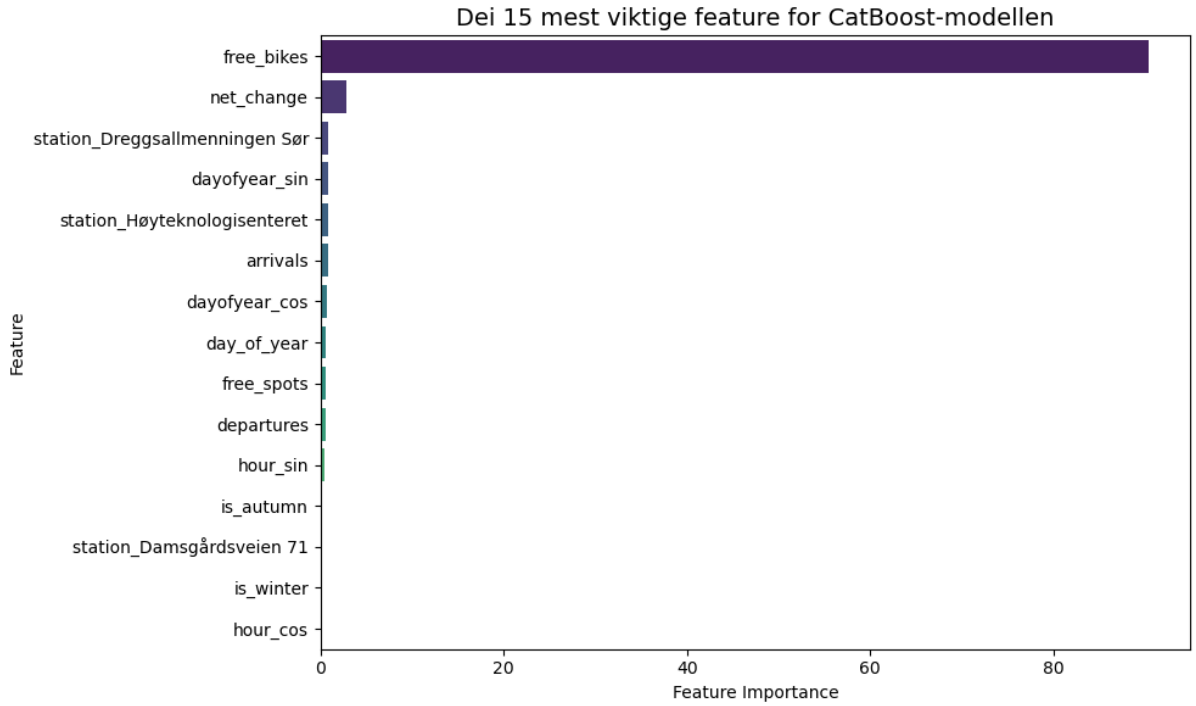
5 Resultat

Modell	Trening RMSE	Validering RMSE	Test RMSE
DummyRegressor	6.585	6.097	-
Lasso	1.130	1.294	-
LightgbmRegressor	0.935	1.290	-
CatBoostRegressor	0.953	1.218	1.254
XGBRegressor	1.045	1.268	-

Tabell 5. Tabellen viser de ulike maskinlæringsmodellane som vart brukt og deira feilrate (RMSE) på trening og valideringsdata. Berre den modellen med lågast feilrate på valideringsdata vart evaluert med testdata.

Resultatet viser at modellane presterte langt betre enn baseline modellen. Feilraten modellane hadde på valideringsdatasettet var ganske likt, og låg alle mellom 1.2 og 1.3 RMSE. Modellen med best predikasjonsevne var CatBoostRegressor med ein feilrate på 1.218 på valideringsdata og 1.254 for testdata. Ein feilrate på 1.25 betyr at modellen i snitt bommar med om lag éin sykkel per stasjon per time, noko som er godt innanfor eit praktisk bruksområde.

5.1 Viktigeste features for CatBoostRegressor



Figur 13. Figuren viser de featursa som den beste modellen leggje mest vekt på. Frå figuren ser ein at dette er variabelen «free_bikes». Ein ser dermed at predikasjonsverdiane til modellen er mest basert på nåverande syklar ved kvar stasjon kvar time. Det viser framleis at modellen tar variablar som år, dag og tid i betraktning men veldig lite vêrdata.

5.2 Predikasjonstest på sist registrert data

I filen predict.py blir den beste modellen testa på å predikere ledige syklar éin time fram i tid. Den tar den siste observasjonen for kvar stasjon i df_model_ready, som er 02.05.2025 klokka 15:49 i standard UTC, og skal predikere for timen 17:00. I Bergen lokaltid blir siste observasjon 17:49, og dette blir tatt hand om etter predikasjonen. Når predict.py blir køyrt, får vi dette i terminalen:

```
siste tidsstempel i data 2025-05-02 17:49:25+02:00
Neste hele time: 2025-05-02 18:00:00+02:00
Time å predikere 2025-05-02 19:00:00+02:00
Stasjon Akvariet : Nåværende 1.0 sykler, Predikert 1.0 sykler
Stasjon Damsgårdsveien 71 : Nåværende 15.0 sykler, Predikert 15.0 sykler
Stasjon Dreggsallmenningen Sør : Nåværende 21.0 sykler, Predikert 21.0 sykler
Stasjon Florida Bybanestopp : Nåværende 14.0 sykler, Predikert 14.0 sykler
Stasjon Grieghallen : Nåværende 14.0 sykler, Predikert 14.0 sykler
Stasjon Høyteknologisenteret : Nåværende 21.0 sykler, Predikert 19.0 sykler
Stasjon Møllendalsplass : Nåværende 4.0 sykler, Predikert 4.0 sykler
Stasjon Studentboligene : Nåværende 15.0 sykler, Predikert 14.0 sykler
Stasjon Torgallmenningen : Nåværende 20.0 sykler, Predikert 19.0 sykler
```

6 Refleksjon

Arbeidet med prosjektet viste tydleg kor viktig det er å forstå data før ein begynner med modellering. Når ein får tre ulike og uavhengige datasett er det viktig og ta med seg det viktigaste og gjere eit bra forarbeid før samanslåing av datasetta. Med ein feilrate (RMSE) på 1.218 med beste modell, er det bevist at modellen har lært seg mønster frå det samanslåtte datasettet og ved hjelp av features engineeringa. Eg har erfart frå denne type data, at sykliske tidsmønster er heilt avgjerande for å få låg feilrate, samt behandlinga av manglande verdiar, for å få ein realistisk og komplett data.

Prosjektet var utfordrande med tanke på ujamne tidsrekkjer i stations.csv og trips.csv, som kravde ein del handtering for å unngå datalekkasje. For modellen, så kunne den kanskje blitt forbetra med meir kontinuerleg historikk frå fleire år, eller fleire vêrfaktorar. Eg kunne ha valt å lage fleire kategoriske bins for vêrdata, som kunne kanskje forbetra RMSE for modellar som handterer dette, men valde og ha alt numerisk.

Noko å leggje merke til også er den urealistiske vind hastighetane i vêrdata. Sjølv om den framleis viser samanheng av minkande turar ved høgare hastighetar, så viser den framleis, det eg vil kalle, urealistiske verdiar. For eksempel er det urealistisk at nokre tusen turar er registrert mellom 30m/s og 40m/s når 23m/s blir rekna som orkan. Dette tok eg ikkje i betraktning under data prosesseringen, og gjekk utifrå at modellane handterer slik data sjølv.

Ein anna forbetring eg kunne ha gjort er å ekskludere for høgt korrelerte features. For eksempel free_bikes og free_spots hadde begge ganske høg negativ og positiv korrelasjon med free_bikes_next. Eg kunne for eksempel ha testa modellar utan desse for å sjekke kor mykje predikasjonen blir påverka av andre faktorar som vêret og tiden. Ein kan sjå dette på figur 13, der det viser at predikasjonsverdiane til CatBoost er veldig avhengig av free_bikes.

7 Konklusjon

Prosjektet oppnådde målet om å lage ein modell som kan predikere talet på ledige syklar ved dei relevante stasjonane éin time fram i tid. Den beste modellen vart CatBoostRegressor, så gav ein feilrate på berre 1.218 RMSE på testdata. Dette viser at modellen fangar opp mønstre i data som sesongvariasjonar, døgnrytme, rushtid og generell sykkelaktivitet, sjølv om den legger mykje vekt på free_bikes variabelen.

Ein gjennomsnittleg feilmargin på om lag ein sykkel per stasjon per time er ganske presist, og er optimalt for praktisk bruk og drift av Bergen Bysykkel. Modellen er derfor godt eigna til vidare bruk på sanntidsdata, og kan bidra til balansering av syklar mellom dei gitte stasjonane.

8 Bibliografi

Bysykkel, B. (n.d.). *Open data*. Retrieved from bergenbysykkel: https://bergenbysykkel.no/en/open-data

FRANCO12. (2019, Februar 23). *Timestamp with Sin and Cosine*. Retrieved from Kaggle: https://www.kaggle.com/code/franco12/timestamp-with-sin-and-cosine

Hyperparameter Tuning. (2025, August 2). Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/machine-learning/hyperparameter-tuning/

Max, H. (n.d.). *MaxHalford/bike-sharing-history*. Retrieved from github: https://github.com/MaxHalford/bike-sharing-history

Open-Meteo. (n.d.). Retrieved from Open-Meteo: https://open-meteo.com/

Sevec, J. (n.d.). *What are Dummy Variables?* Retrieved from mtab: https://mtab.com/blog/what-are-dummy-variables

UIP, B. M. (n.d.). *Bergen Bysykkel*. Retrieved from urbaninfrastructure.no: https://urbaninfrastructure.no/bergen-bysykkel-as/

Utforskende dataanalyse. (2025, September 9). Retrieved from wikipedia: https://no.wikipedia.org/wiki/Utforskende_dataanalyse