INF161 – innføring i data science

Sykehusopphold

Rapport om maskinlæringsmodell som prediker sykehusoppholdet per pasient

Petter Alexander Hoff
03.11.2023

Innholdsfortegnelse

1. Introduksjon

For å effektivisere ressursbruk og forbedre pasientbehandlingen er det viktig å kunne forutsi lengden på sykehusopphold. Målet med dette prosjektet er å utvikle en maskinlæringsmodell som kan estimere oppholdslengden basert på pasientdata som fysiologiske, demografiske og sykdomsalvorlige faktorer. Arbeidet innebærer dataforberedelse, der flere datasett kombineres og manglende verdier imputeres, etterfulgt av modellering med ulike teknikker som lineær regresjon, Lasso og Random Forest. Modellenes ytelse evalueres ved hjelp av RMSE, og rapporten diskuterer både utfordringer og mulige forbedringer for fremtidig utvikling.

2. Datatilberedning

I denne delen gjennomgås datasettbeskrivelsen og nødvendige tilpasninger i dataframen. Dataen deles deretter opp i trenings-, validerings- og testsett. Deretter vil det bli gjort forskjellige imputeringsteknikker, variabel utvinning og tilslutt visualisering av relevante variabler.

2.1 Sammenslå datasettene

Sammenslåingsprosedyre

Datasettene slås sammen, en viktig ting vi må gjøre før vi slår sammen datasettene til en dataframe er at dataen I Informasjon om sykdomsalvorlighet er formatert som en json fil, for å håndtere dette bruker vi pd.explode.

Dataene fra de fire datasettene blir slått sammen på variabelen *pasient_id* for å etablere et datasett med en felles struktur for videre analyse. Dette sikrer at alle tilgjengelige variabler og informasjon om pasientene samles i ett datasett for mer effektiv modellering og prediksjon av *oppholdslengde*.

Datahåndtering

Ved sammenslåing av datasettene møter vi utfordringer med manglende verdier. For å håndtere dette, fyller vi manglende verdier hentet

fra https://hbiostat.org/data/repo/supportdesc, utarbeidet av Professor Frank Harrell. Dette er realistiske estimater som kan gi et solid grunnlag for variablene som mangler data.

Variabler som *bilirubin* fjernes fra dataframen fordi målingene foretas først på dag 7. Videre fjernes også variablene *dødsfall* og *sykehusdød*, ettersom disse ikke bidrar til å estimere lengden på sykehusoppholdet. ID variablene pasient_id og sykdomskategori_id blir fjernet fordi det er bare en Indikator variabel som ikke bidrar til å estimere sykehusoppholdet.

Jeg har vurdert å fjerne variabelen *etnisitet*, grunnet etisk grunnlag på å skille pasienter etter etnisitet. Likevel, har jeg valgt å beholde variabelen da den kan gi oss verdifull informasjon og hjelpe modellen vår på å predikere sykehusopphold.

2.2 Deling av trening, test og validerings data

Vi velger å dele inn I 80% trening, 10% test og 10% validerings data. Denne inndelingen gir oss mye data på modell trening og samtidig beholder vi en god del mengde data til evalurering og testing. Grunnen for at vi velger en 80/10/10 fordeling er fordi datasettet vårt er veldig stort og inneholder 7742 rader og 38 kolonner. Dette sikrer oss en bedre evaluering med mere data å kunne trene modellen på.

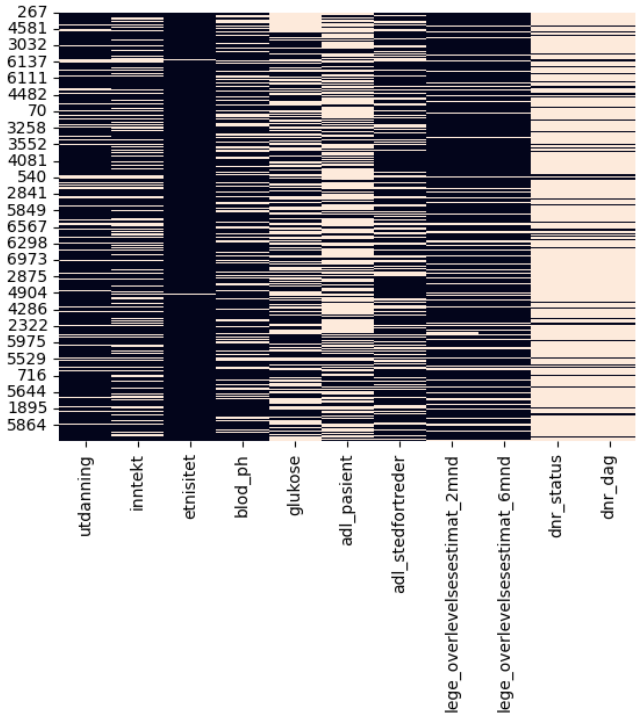
2.3 Håndtering av manglende og feil verdier

Feil verdier

Startet med å finne alle negative variabler i treningsdata, deretter sjekker vi databeskrivelsen for å se om det stemmer. Vi finner ut at det har skjedd feil i målingene og finner negative verdier for alder og oppholdslengde. Det blir da fornuftig å fjerne alle radene som inneholder negative verdier for disse to variablene, da vi ikke kan være sikre på om resten av verdiene i raden er korrekt.

2.3.1 Fjerning av variabler og imputasjon for manglende verdier

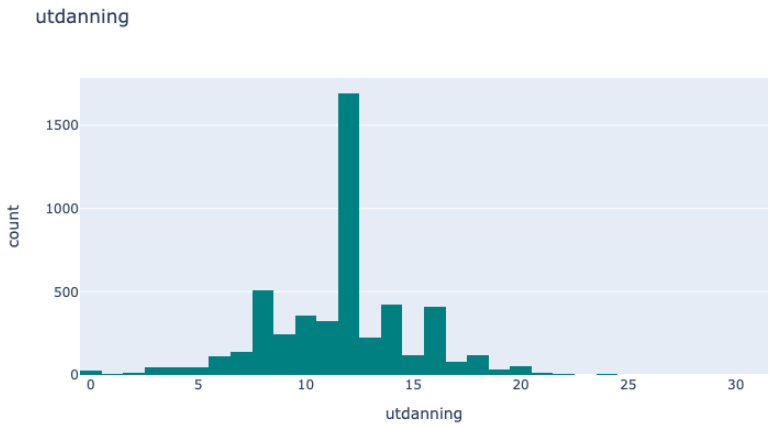
For å bedre kunne se de variablene med manglende verdier, lager vi en egen dataframe med alle variablene som mangler verdier og visualiserer med et heatmap.



Figur 1, visualisering av alle variablene med manglende verdier

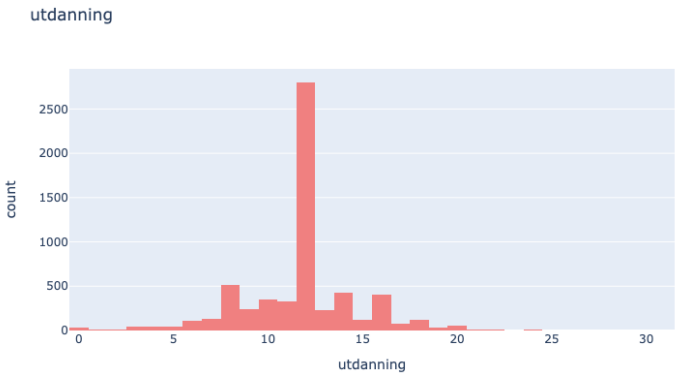
Jeg satte en grense på 50 % manglende verdier, hvor variabler med mer enn 50 % manglende data blir vurdert for fjerning. Jeg fjernet variablene *glukose*, *dnr_status,* og *dnr_dag* siden de hadde for høy andel manglende data og kunne ikke manipuleres på en meningsfull måte.

Utdanning representerer antall år med utdanning og mangler 18.1% av verdiene.



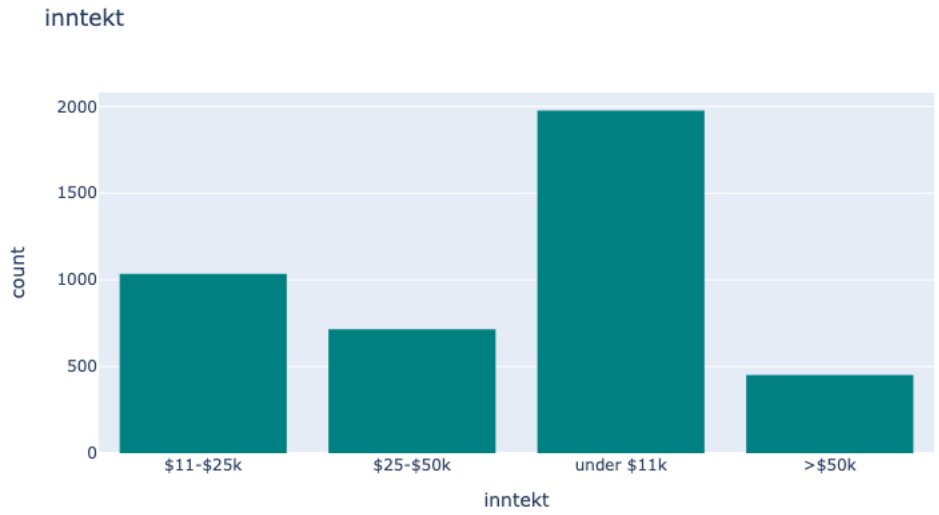
Figur 2, stolpediagram av utdanning variabelen

Utdanning har et høyt antall pasienter med 12 år utdanning, jeg valgte derfor modus for å erstatte de manglende verdiene med verdien som forekommer mest oftest, som er 12 år. Jeg har valgt å bruke modus for å sikre en mer nøyaktig og representativ imputering av utdanningsnivået. Ved å bruke modus fremfor median eller gjennomsnitt unngår vi påvirkningen av ytteverdier som ikke representer den typiske fordelingen.



Figur 3, stolpediagram av utdanning variabelen etter modus imputasjon

I den kategoriske variabelen *Inntekt,* mangler det 32.49% verdier. Siden det er så mange manglende verdier, velger vi å ikke ha en imputerings strategi, men heller lage en *inntekt_mangler* variabel og legge til de manglende verdiene til variabelen.

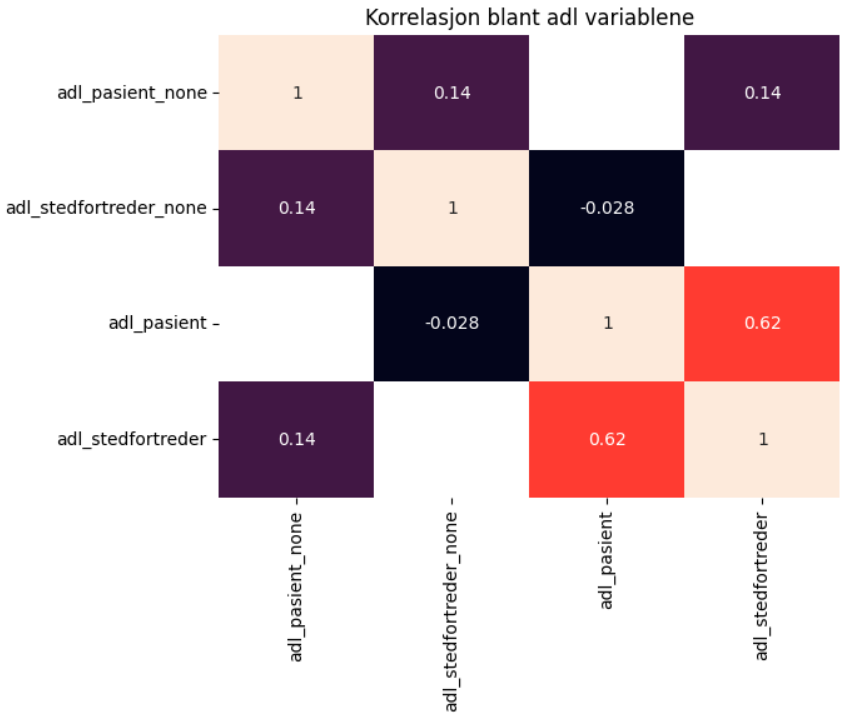


Figur 4, stolpediagram av inntekt variabelen

For *blod_ph*, *lege_overlevelsesestimat_2mnd* og *lege_overlevelsesestimat_6mnd* ser vi ved describe() metoden at verdiene er jevnt fordelt, derfor bruker vi imputasjon med middelverdi for å erstatte manglende verdier.

2.3.2 Variabelutvinning

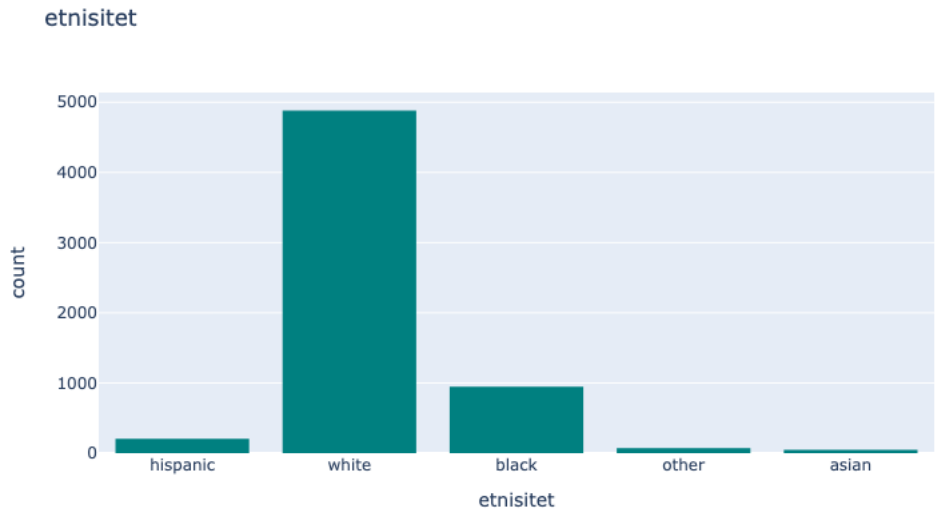
adl_pasient og *adl_stedfortreder* hadde 61,93 % og 31,69 % manglende verdier. For disse variablene valgte jeg en annen tilnærming, da adl_pasient fylles ut av pasienten på dag 7, mens adl_stedfortreder fylles ut av en stedfortreder på dag 1. Vi setter opp en korrelasjonsmatrise for å se korrelasjonen mellom variablene og null verdiene.



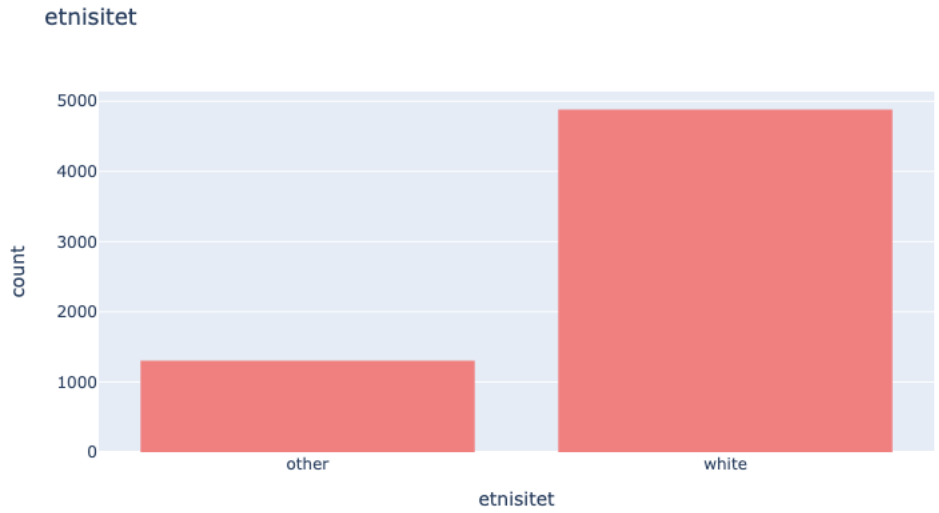
Figur 5, korrelasjonsmatrise av adl_pasient, adl_stedforteder og null verdiene

Jeg konstruerte to nye variabler: *adl* og *adl_endring*. Variabelen adl kombinerer informasjon fra både *adl_pasient* og *adl_stedfortreder*, der verdien *fra adl_pasient* prioriteres. Dersom *adl_pasient* mangler, brukes verdien fra *adl_stedfortreder* i stedet. *Adl_endring* er en variabel som forklarer oss endringen mellom *adl_stedforteder* og *adl_pasient*, siden *adl_stedfortreder* blir målt dag 1 og *adl_pasient* målt dag 7. *Adl_endring* sørger for at vi ikke relasjonen mellom *adl_pasient og ald_stedforteder.* Dette gir en variabel som samler informasjon om pasientens funksjonsnivå, uavhengig av hvem som rapporterer.
For å håndtere de resterende manglende verdiene, brukte jeg KNN-imputer (med 2 nærmeste naboer) trent på variablene adl og adl_endring. Imputeringsvalget er basert på antakelsen om at liknende ADL-verdier mellom pasienter også kan indikere liknende manglende verdier.

Etnisitet har en majoritet av hvite. Vi ser at minoritetene har mye mindre verdier, og for å unngå en ujevn fordeling samler vi alle minoritetene til en other verdi. Det er også 0.52 % manglende verdier i *etnisitet* som vi velger å også legge til i other for en jevnere fordeling.



Figur 6, stolpediagram av etnisitet variabelen

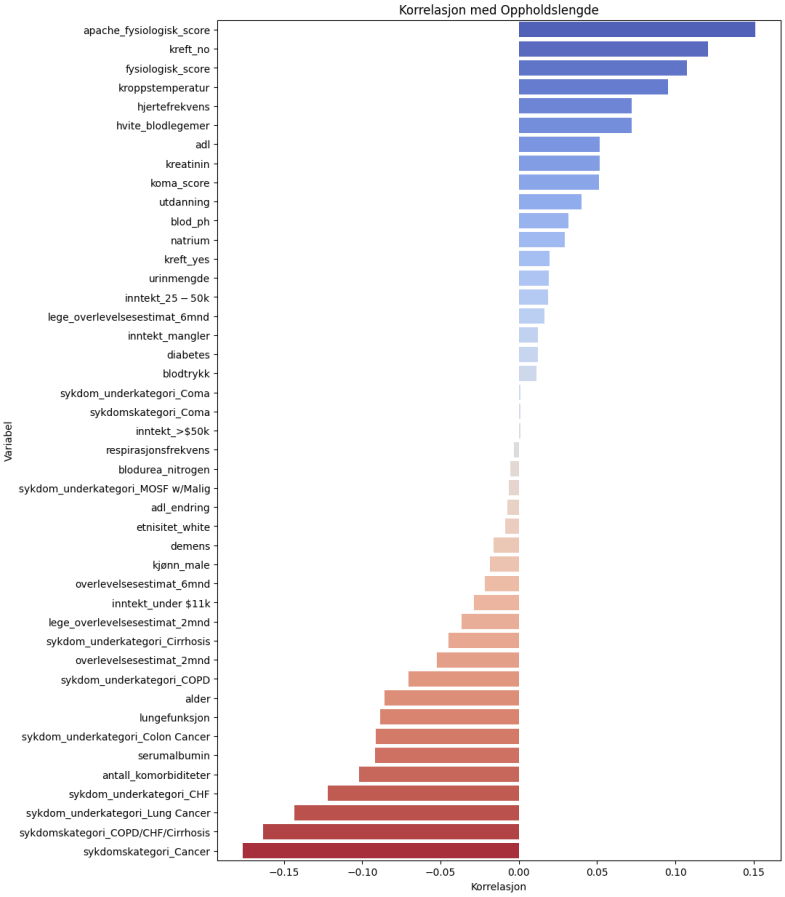


Figur 7, stolpediagram av etnisitet variabelen etter gruppering av minoritetene

Til slutt lager vi dummy-variabler for hver kategori på våre kategoriske data, med pandas get_dummies metoden. Dette gjør vi for å sikre oss at modellene våre klare å skille de forskjellige kategoriene.

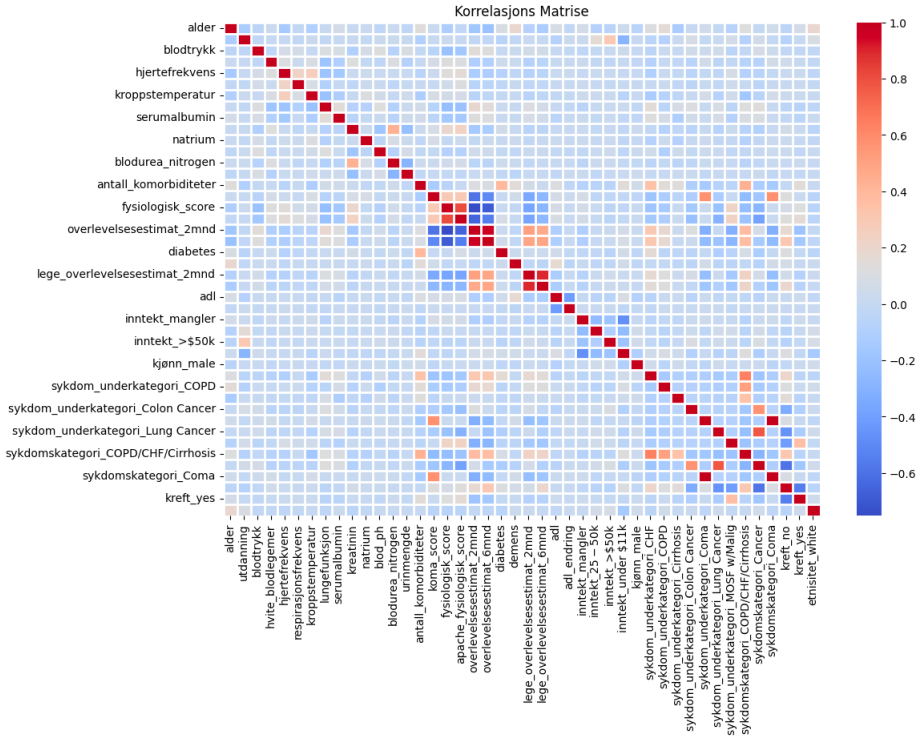
2.4 Visualisering

Starter med å se på korrelasjon mellom alle variablene i treningsdata og oppholdslengde. Vi ser i figur x at *apache_fysiologisk_score* og *fysiologisk_score* har høye verdier for korrelasjon



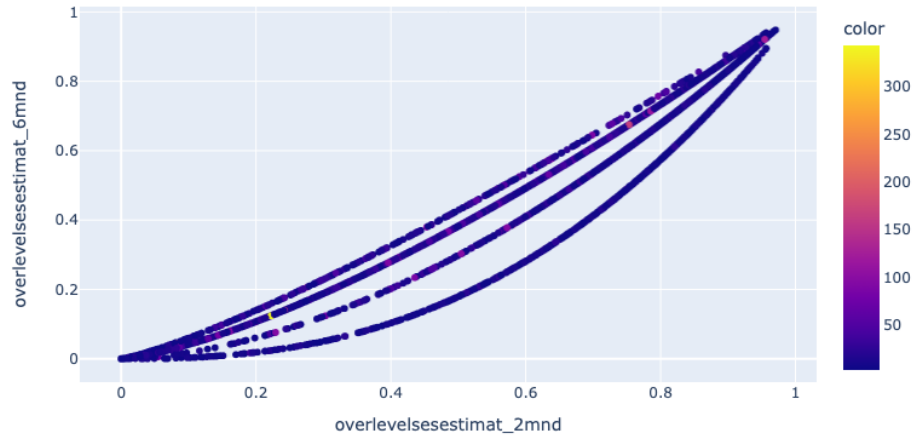
Figur 8, barplot som viser korrelasjon mellom variablene i trening settet og oppholdslengde

Videre ser på korrelasjon mellom variablene i en korrelasjons matrise, dette gir oss innsikt i forholdet mellom variabler som har positiv korrelasjon eller negativ korrelasjon. I figur 9 ser vi at vi har høy korrelasjon mellom variablene *overlevelsesestimat_2mnd* og *overlevelsesestimat_6mnd, fysiologisk_score* og *apache_fysiologisk_score* og *lege_overlevelsesestimat_2mnd* og *lege_overlevelsesestimat_6mnd*.

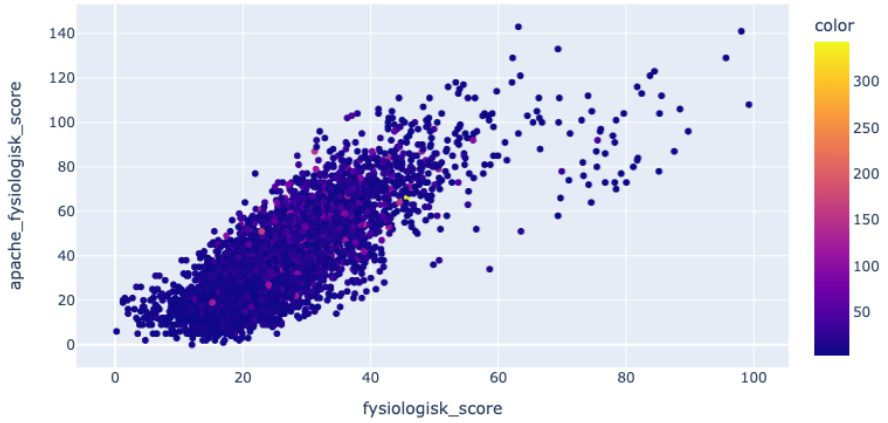


Figur 9, Korrelasjons matrise av alle variablene

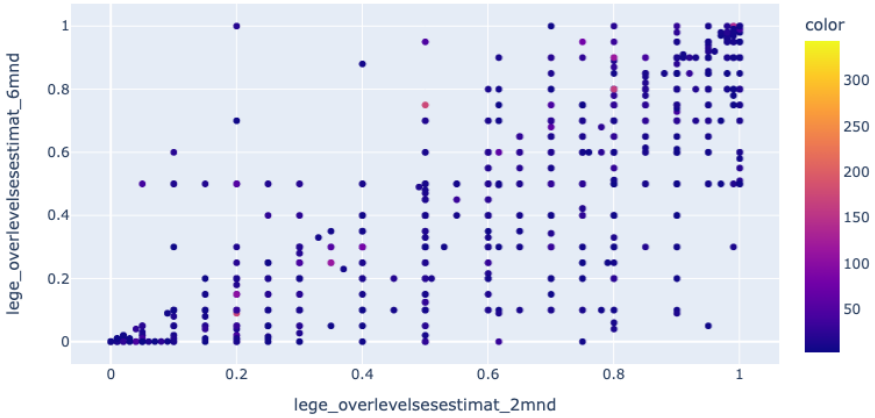
Vi velger å lage scatterplots av variablene med høy korrelasjon mellom, og setter color til å være oppholdslengden. Det gjør det mulig å se etter mulige trender med begge variablene og oppholdslengden. Ser at i Figur 10, er det tydelig et forhold mellom *overlevelsesestimat_2mnd* og *overlevelsesestimat_6mnd.* I Figur 11 er det et stor overlapp av verdier, noe som kan tyde på at variablene inneholder en del like verdier. Figur 12 ser verdiene spredt ut langs x og y aksen, noe som tilsier at variablene er unike.



Figur 10, scatterplot av overlevelsesestimat_2mnd og overlevelsesestimat_6mnd

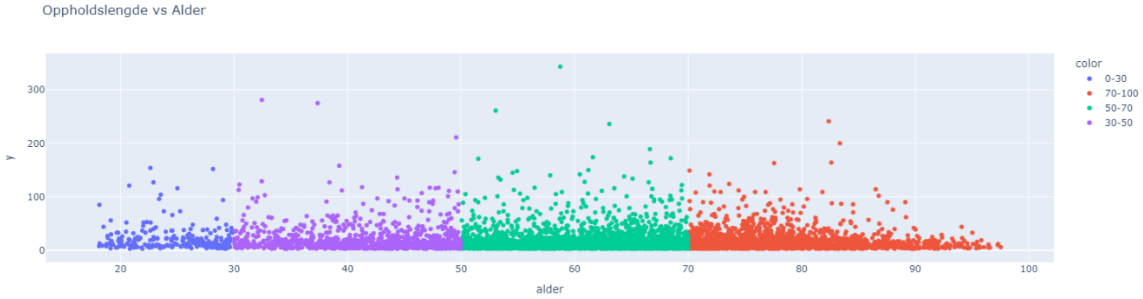


Figur 11, scatterplot av fysiologisk_score og apache_fysiologisk_score

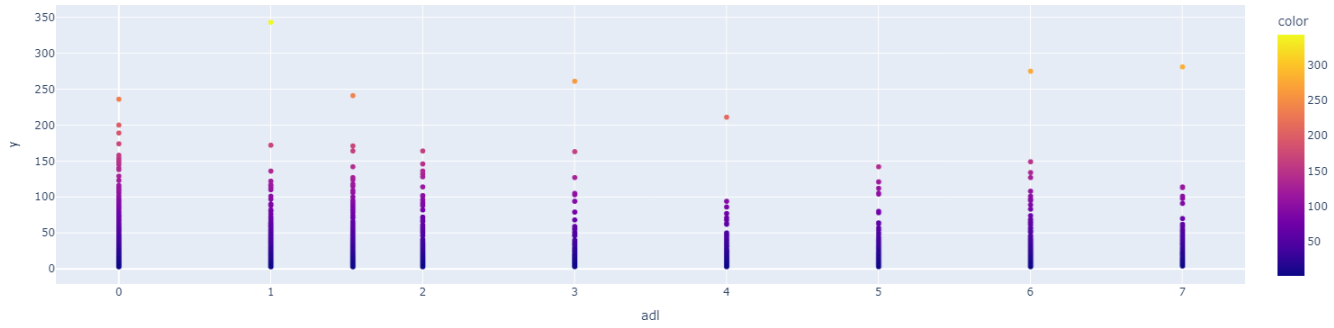


Figur 12, scatterplot av lege_overlevelsesestimat_2mnd og lege_overlevelsesestimat_6mnd

Til slutt visualiserer vi alder og adl, da alder har høye negative verdier for korrelasjon og adl har høye positive verdier for korrelasjon. Dette er for å gi et innblikk i hvordan variabler med høy og liten korrelasjon ser ut mot oppholdslengde.



Figur 13, scatterplot av oppholdslengde på y aksen og alder på x asken



Figur 14, scatterplot av oppholdslengde på y aksen og adl på x aksen.

3. Modellering

3.1 Testing av forskjellige modeller

Jeg har valgt å bruke modellene lineær regresjon, Lasso modell og random forest. I tillegg har jeg trent en grunnlinjemodell, for å evaluere om modellene man trener gir gode nok resultater. Jeg har valgt regresjonsmodeller, ettersom målet er å predikere en kontinuerlig variabel

For å evaluere modellene har jeg sammenliknet den gjennomsnittlig kvadrert feil (MSE) og Kvadratroten av gjennomsnittlig kvadrert feil (RMSE).

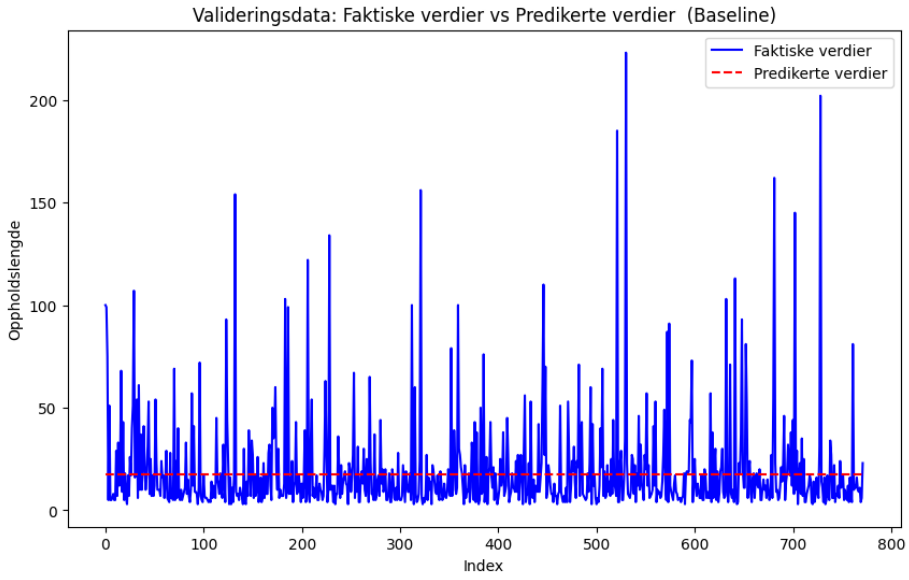
$$\sum_{i=1}^{N}(f(x_i) - y_i)^2.$$

- N er antall datapunkter,
- fx_ier modellen sin prediksjon for input xi
- y_i er faktiske den verdien for nummer i

Jo lavere verdi i for (MSE), jo lavere forskjell mellom predikert oppholdslengde og den faktisk oppholdslengden for validerings settet.

Baseline

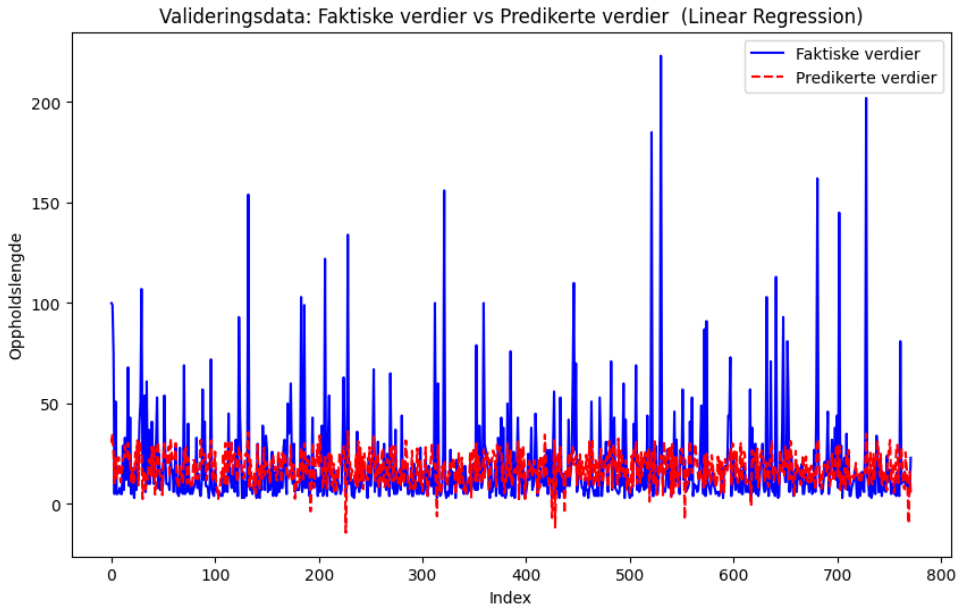
Vi starter med å evaluere en baseline modell. Dette gir oss en MSE på 597.33 og RMSE på 24.44, hvor middelverdien er satt til 17.53.



Figur 15, baseline modell prediksjon på valideringsdata

Lineær Regresjon

Lineær regresjon gir en lavere gjennomsnittlig kvadrert feil (MSE) enn baseline modellen, med en MSE på 501.052 og en RMSE på 22.38. Dette er som forventet, siden prediksjonene her avhenger av inputvariablene.

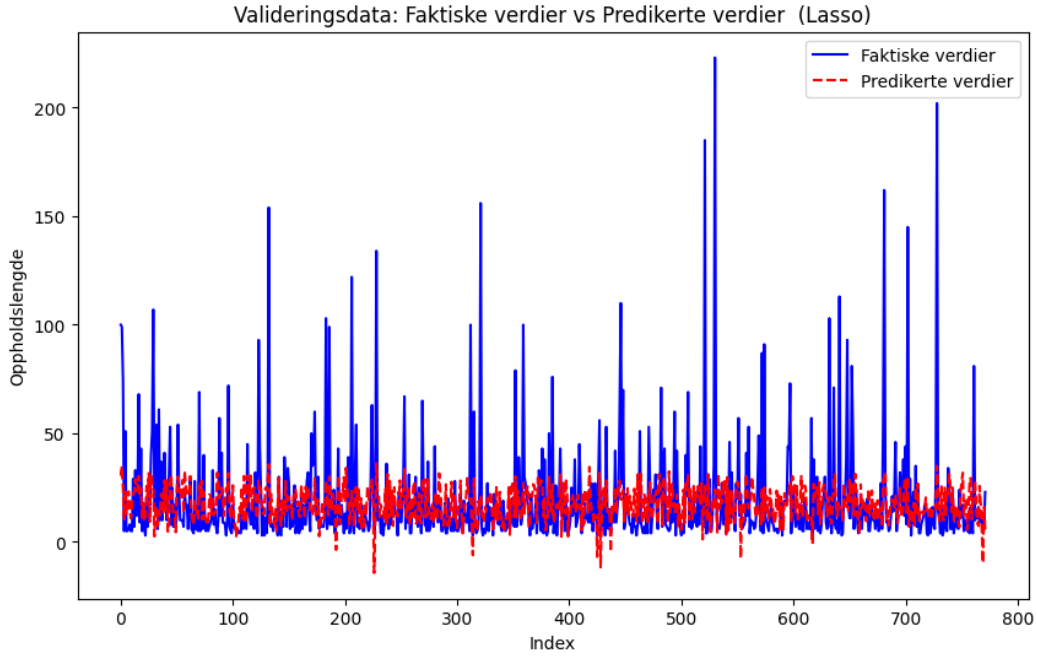


Figur 16, Linear Regresjon modell prediksjon på valideringsdata

Lasso modell

Den neste modellen er en lasso regresjon, hvor en alpha-verdi kontrollerer graden av regulering av koeffisientene. For å finne riktig alpha, testet vi flere verdier mellom 0 og 5, totalt 50 alternativer. Deretter valgte vi lasso modellen med lavest MSE. Den

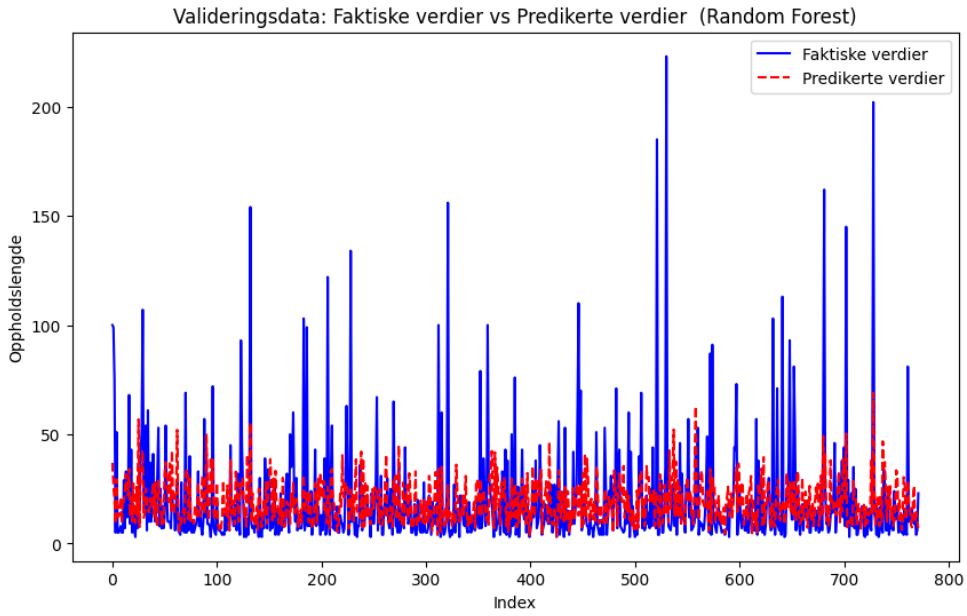
beste lasso modellen oppnår en MSE på 501.052 og en RMSE på 22.38.



Figur 17, Lasso modell prediksjon på valideringsdata

Random Forest

Random Forest-modellen gir den laveste gjennomsnittlige kvadrerte feilen, med en MSE på 496.298 og en RMSE på 22.278.



Figur 18, Random Forest modell prediksjon på valideringsdata

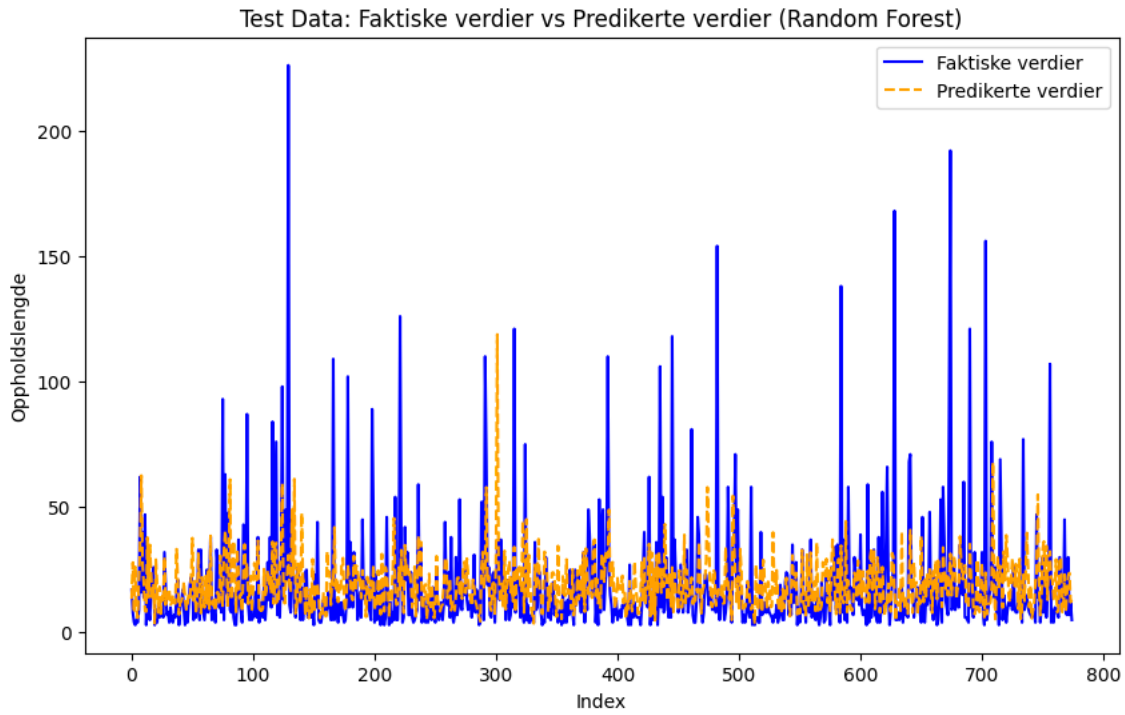
3.2 Modellutvalg

For valg av modell er det viktig å ta hensyn til modellens formål. I denne oppgaven skal vi lage en modell som predikerer lengden på sykehusopphold basert på tilgjengelige pasientdata. Baseline modellen bruker kun middelverdien som prediksjon, noe som ikke er ønskelig fordi en modell som kun gir gjennomsnittet for oppholdslengde uavhengig av input, ikke er nyttig.

Både lineær regresjon og lasso modellen gir prediksjoner som tilpasser seg inputdataene og dermed reflekterer variasjoner. Imidlertid viser figur 16 og 17 at disse modellene av og til predikerer negative oppholdslengder. Dette er ulogisk og gjør at modellene ikke er pålitelig nok.

Random Forest-modellen gir den laveste gjennomsnittlige kvadrerte feilen (MSE), og dermed også den laveste RMSE. Modellen har en gjennomsnittlig feilmargin på 22,78 dager på testdataene. Videre ser vi i figur 18 at Random Forest ikke predikerer negative verdier for oppholdslengde, noe som gjør modellen mer pålitelig i praktisk bruk.

På bakgrunn av disse vurderingene velger vi derfor Random Forest som vår endelige modell for prediksjon på testdataene.



Figur 19, *Random Forest modell prediksjon på testdata*

3.3 Prediksjon på sample data

Før vi kan predikere sykehusoppholdet for sample data, må vi utføre de samme endringene som vi gjorde på treningsdata i vår opprinnelige dataframe. Siden jeg allerede har dokumentert og begrunnet hvert steg, velger jeg å lage en Python-fil, «script.py», der jeg definerer en funksjon som utfører de samme endringene på sample data. Denne funksjonen, *fill_missing_values*, vil returnere en klargjort dataframe som kan brukes til prediksjon i modellen vår. Modellen blir brukt til å predikere sykehusoppholdet og lager filen «predictions.csv»

3.4 Nettside implementasjon

For å sikre god brukervennlighet på nettsiden, har jeg fokusert på å gjøre det enkelt for alle brukere å fylle inn de nødvendige verdiene. Enkelte variabelnavn kan være vanskelige å forstå, men jeg har lagt til en lenke til databeskrivelsen øverst på siden, slik at brukerne enkelt kan slå opp informasjon om variablene.

For kategoriske variabler er det mulig å velge mellom «ja» og «nei» for å forenkle prosessen og gjøre skjemaet mere forståelig. Når skjemaet er sendt inn, tildeles de oppgitte verdiene til de riktige variablene. Dataene sorteres deretter i samme rekkefølge som i treningsdataene og legges til i en dataframe. Prediksjonen vises nederst på skjermen, slik at brukeren kan se resultatet.

4. Refleksjon og Forbedringer

4.1 Diskusjon

Prosjektet har gitt god trening i prosessen med å utvikle en modell som predikerer sykehusoppholdslengde. Gjennom analysen av datasettet ble det klart at manglende verdier og skjevt fordelte verdier krevde datarensing og imputasjon. Det stilte krav til å lese nøye igjennom databeskrivelsen, for å finne ut av verdien til hver variabel og data.

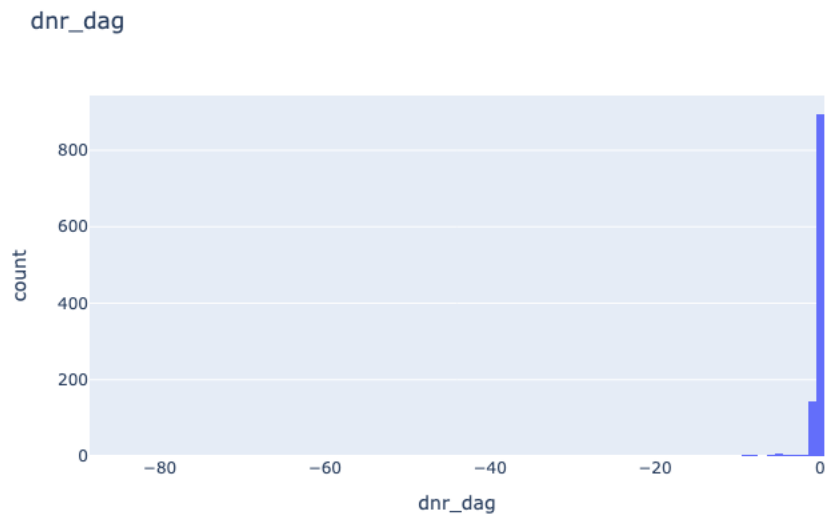
Når det gjelder hyperparametere, ga Lasso-modellen utfordringer med valg av optimal alpha-verdi. Etter testing av ulike verdier mellom 0 og 5, viste det seg å være en tidkrevende, men bidro til lavere gjennomsnittlig kvadrert feil. Hadde jeg hatt mere tid, burde jeg testet ut hyperparametere for min endelige modell, random forest. Dette hadde gitt lavere gjennomsnittlig kvadrert feil, og ført til en bedre

Visualiseringene, inkludert korrelasjonsmatriser og scatterplots, ga en tydelig fremstilling av relasjoner mellom variabler og understøttet forståelsen av datasettets struktur.

4.2 Implementering som ikke ble med i oppgaven

I en sånn type oppgave, er det mye prøving og feiling. Her er noen av tingene som ble prøvd, men kom ikke med i oppgaven.
Endra alle manglede verdier i dnr_dag til 1. Tanken her var at grunnen til at det var vedlig mange NaN verdier i dnr_dag, var fordi dersom målingene ble gjort etter at pasiententen ble innlagt, ble verdien NaN. Tenkte det kunne være lurt å gjøre om

manglende verdier til 1 for å håndtere manglende verdier, men modellen ble forvirret av verdiene og fikk en mye høyere gjennomsnittlig kvadrert feil.



Figur 20, dnr_dag ved implementering av 1 for manglende verdier

Jeg rundet ned alle aldersverdiene fordi en pasient ikke fyller en bestemt alder før bursdagen sin. Modellen presterte imidlertid dårligere med de avrundede verdiene, så jeg valgte å droppe denne implementasjonen. Likevel syntes jeg det var en interessant tilnærming, men jeg valgte det bort for bedre modellresultater.

4.3 Mulige forbedringer

Selv om modellen gir gode resultater, kan flere tiltak ytterligere forbedre prediksjonskraften og nøyaktigheten. Med mer tid og ressurser kunne det vært lurt å utforske flere modeller for bedre håndtere komplekse forhold mellom variabler og mulig gi mer presise prediksjoner.

En mer detaljert håndtering av kategoriske variabler, som target encoding, kunne mulig gi bedre representasjon av enkelte variabler sammenlignet med binære dummyvariabler. Dette kan gi modellen mer nyansert informasjon om variabler med flere kategorier.

Å bruke mer tid på variabel utvinning kunne gjort det mulig å inkludere flere relevante variabler, som for eksempel i figur 10 kan vi se at *overlevelsesestimat_2mnd og overlevelsesestimat_6mnd* har klare trender sammen. Å lage et forholdstall som fanget opp trendene mellom disse variablene hadde vært en god ide.

Til tross for at modellen presterer godt, er det klart at forbedringer innen både dataforberedelse og modellutvikling kan optimalisere prediksjonene ytterligere. Prosjektet har vært en verdifull læringsopplevelse, og dersom jeg hadde hatt mere tid ville jeg tatt i bruk disse tilnærmingene for bedre modellering.