Estimering av sykehusopphold

Prosjekt rapport

Gedvyde Petkeviciute, Universitet i Bergen 2024

INTRODUKSJON OG FORMÅL

Dette datavitenskapsprosjektet har som mål å utvikle en maskinlæringsmodell for å predikere forventet oppholdende for kritisk syke pasienter på sykehus. Rapporten dokumentere hele prosessen, fra utforskende datanalyse til modellutvikling og utvelgelsen av den endelige modellen samt identifiserer muligheter for videre forbedringer. Den endelige modellen er designet for å håndtere komplekse datasett med pasientopplysninger, inkludert fysiologiske, demografiske og sykdomsalvorlighetsdata på tvers av ni sykdomskategorier.

Innholdsfortegnelse

1. Tilberedning av data

1.1 Rensing og klargjøring av data for analyse

Før jeg setter i gang med utforskende dataanalyse, starter jeg med å sette sammen rådata til et df. I neste steg fjerner jeg pasient duplikater fra df. Videre fjerner jeg alle pasienter som har en negativ oppholdslengde, da oppholdslengden er den variabelen vi ønsker å predikere, og negative verdier i denne konteksten er ikke meningsfulle. Deretter erstatter jeg alderen til alle pasienter med negativ alder med NaN-verdier, ettersom negative aldersverdier er ugyldige. Til slutt endrer jeg indeksen til df slik at den er basert på pasient-ID. Dette fordi pasient-ID kun har en identifiserende funksjon og vil ikke bidra til prediksjonen av oppholdslengden.

```
df shape før: (7742, 41)
df shape etter: (7734, 40)
```

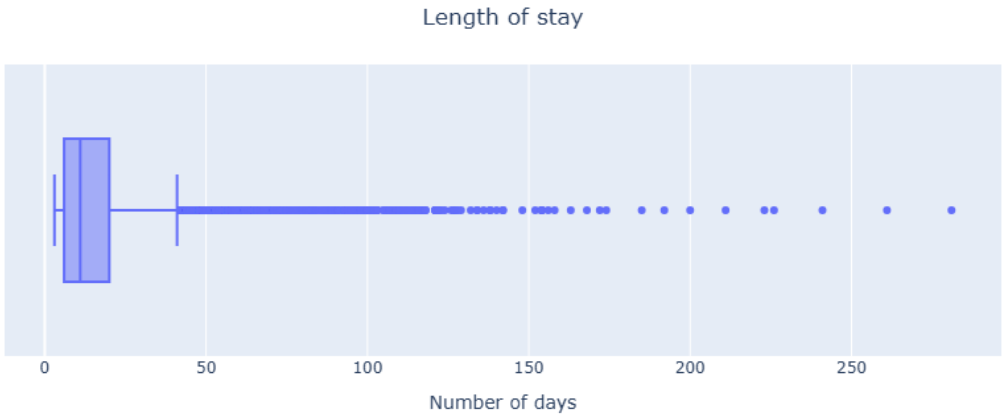
1.2 Innledende dataanalyse og beskrivelse av treningsdata

For å gjennomføre utforskende dataanalyse deler jeg df inn i trenings, validerings og test df`s. Jeg benytter *describe*-metoden for å få en statistikk oppsummering av trenings dataen. Dette gir meg flere viktige innsikter. I raden for antall målinger (count) observerer jeg at verdien varierer mellom kolonnene. Dette indikerer at noen kolonner mangler mer data enn andre, noe jeg må ta hensyn til i videre analyser.

I kolonnen til oppholdslengden ser jeg at oppholdslengden har en relativt lav median (11 dager) sammenlignet med gjennomsnittet (17 dager), samtidig et stort standardavvik (20 dager). Variabelen har også en høy maksimumsverdi (281 dager) og lav minimumsverdi (3 dager). Dette indikerer at en liten del av pasientene har en signifikant lengre oppholdslengde og bidrar med å trekke opp gjennomsnittet. Samlet sett indikerer dette en høyre-skjev fordeling, noe jeg kan bekrefte eller avkrefte gjennom visualisering. Den etterfølgende figuren understøtter denne konklusjonen.

	sykehusdød	oppholdslengde	alder	utdanning	blodtrykk	hvite_blodlegemer	hjertefrekvens
count	5413.000000	5413.000000	5409.000000	4464.000000	5413.000000	5290.000000	5413.000000
mean	0.261777	17.431184	62.625146	11.778898	84.371790	12.413301	97.116515
std	0.439643	20.795726	15.650004	3.452983	27.813685	9.281022	31.600071
min	0.000000	3.000000	18.041990	0.000000	0.000000	0.000000	0.000000
25%	0.000000	6.000000	52.790990	10.000000	63.000000	7.000000	71.000000
50%	0.000000	11.000000	64.913940	12.000000	77.000000	10.599609	100.000000
75%	1.000000	20.000000	74.208980	14.000000	107.000000	15.298828	120.000000
max	1.000000	281.000000	101.847960	30.000000	195.000000	128.781250	232.000000

(Se hele tabellen i jupyter notebook)



1.3 Konvertering av datatyper for kontinuerlige og heltallige variabler

For å se hvilken type data hver variabel inneholder bruker jeg dtypes-egenskapen til df. Jeg ser at flere av de kontinuerlige og heltallige variablene er lagret som objekter. De er med andre ord lagret i tekstformat, til tross for at den underliggende verdien er av typen int/float. Jeg konverterer disse variablene til float slik at variablene er lagret i korrekt datatype.

1.4 Vurdering og fjerning av variabler

Variablene adl_pasient og adl_stedfortreder måler begge pasientens daglige funksjonsaktiviteter, men med en viktig forskjell: adl_pasient er utfylt av pasienten selv på dag 1, mens adl_stedfortreder er fylt ut av en stedfortreder allerede på dag 1. I et forsøk å redusere antall manglende verdier og antallet variabler valgte jeg opprinnelig å slå disse to variablene sammen som en form for variabel utvinning. Data dokumentasjonen har i etterkant blitt oppdatert og jeg ble opplyst at adl_pasient først er tilgjengelig på dag 7. Siden målet med modellen er å predikere pasientenes oppholdslengden fra dag 1, kan vi ikke inkludere variabler som ikke er tilgjengelig fra denne dagen. Derfor velger jeg å fjerne variablene adl_pasient, samt andre variabler som kun blir tilgjengelig etter dag 1, slik som bilrubin, sykehusdød og dødsfall.

Variablene dnr_status og dnr_dag gir informasjon om hvorvidt pasienten har en ikke-gjenopplivningsordre (DNR). Dnr_status angir om pasienten har en DNR-ordre ved eller før innleggelse, mens dnr_dag viser hvilken dag DNR-ordren ble gitt. Manglende verdier i dnr_dag oppstår fordi pasienter uten DNR-ordre på dag 1, får tildelt Nan-verdi. Siden modellen skal predikere oppholdslengden basert på informasjon fra dag 1, hadde det uansett vært irrelevant å vurdere DNR-ordre som først blir tilgjengelige senere. I tillegg mangler begge variablene verdier i 86% av tilfellene, som gjør de mindre nyttig for analysen og jeg velger derfor å fjerne dem fra datasettene.

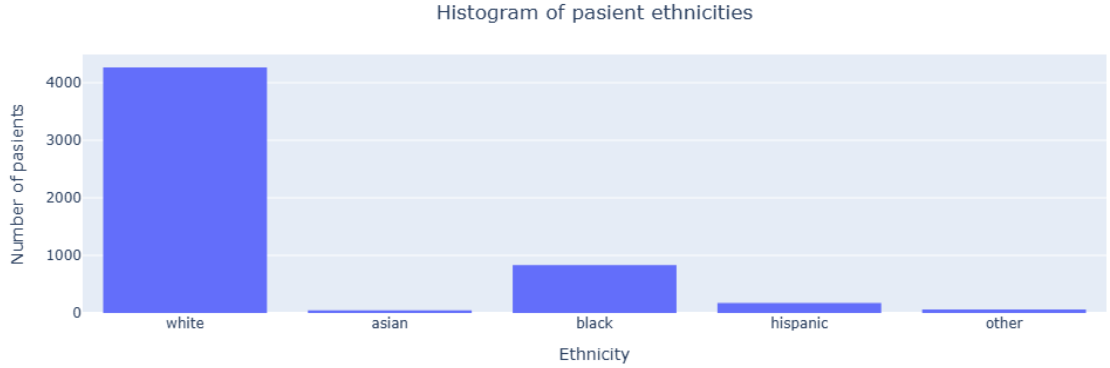
1.5 Bruk av anbefalte verdier for å fylle inn manglende data

I den opprinnelige datakilden (Harrel, 2022) som fulgte med datasettet, ble det presentert noen anbefalte verdier som hadde vist seg å være nyttige for å fylle inn manglede verdier. Jeg benytter disse verdiene for å eliminere antallet manglende verdier for følgende variabler: serumalbumin, lungefunksjon, kreatinin, blodurea nitrogen, hvite blodlegemer og urinmengde.

1.6 Imputasjonsstrategi for kategoriske og numeriske variabler

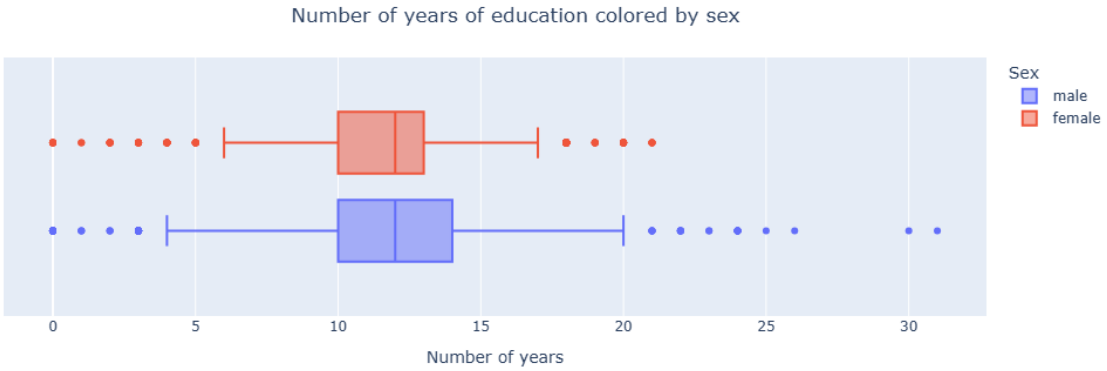
Etnisitet

Etnisitet er en kategorisk variabel som representerer pasientens etnisitet, med mulige verdier {asiatisk, svart, latinsk, mangler, annen, hvit}. I treningsdatasettet mangler det kun 1% av verdier, dette betyr at imputasjons metoden jeg velger vil ha minimal innvirkning. Ettersom «hvit» utgjør den desidert største andelen av pasientene, vil bruk av imputasjons metoden «most frequent» (hyppigst forekommende) bidra til å bevare datasettets fordeling og integritet. Med denne vurderingen velger jeg «most frequent» som imputasjonsstrategi for etnisitet.



Utdanning

Kolonnen for utdanning mangler 18% av sine verdier. Siden jeg ikke ser en systematisk grunn for at utdanningsverdier mangler, antar jeg at verdiene mangler helt tilfeldig. Jeg vurderer derfor å bruke enten middelverdi- eller median som imputasjonsstrategi. For å ta en informert avgjørelse undersøker jeg utdanningskolonnen for å se om data har stor spredning og/eller var skjevt fordelt. Om dataen er tilnærmet normalt fordelt, vil både gjennomsnitt og median være gode imputasjons strategier. Hvis dataen er skjevt fordelt, vil median vært et foretrukket alternativ. Jeg bruker describe-metoden og ser at utdanning har et relativt lavt standardavvik (3 år) og at gjennomsnitt (11.8 år) og median (12 år) er tilnærmet like. Jeg bekreftet også dette med en visualisering. I visualiseringen deler jeg dataen inn etter kjønn og observerer at både, men og kvinner har lik median. For å slippe å håndtere avrunding, velger jeg median som imputasjonsstrategi for utdanning.



Inntekt, alder

Inntekt kolonnen har 33% manglende data, noe som gjør det mer avgjørende å velge en passende imputasjons metode. Dataene er relativt skjevt fordelt, med flest pasient som har inntekt under 11 000, og en median inntekt på 11 -12k. Min første ide for å imputere inntekt var å undersøke sammenhengen mellom inntekt og utdanning. Hvis det finnes en sammenheng, kan jeg bruke utdanningsverdiene til å imputere manglende inntektsverdier.

Jeg bruker krysstabell og chi-kvadrat-test (chi2_contingency) for å vurdere statistisk signifikans mellom de to kolonene. Nullhypotese er at det ikke eksisterer sammenheng mellom utdanning og inntekt. Chi-kvadrat-testen ga en høy chi2-verdi og en lav p -verdi. En høy chi2-verdi indikerer at resultat avviker betydelig fra den forventende verdien, altså har nullhypotesen har et stort avvik. Det er standard praksis å anse en p-verdi under 0.5 eller 0.1 som lav. Resultat vi fikk har en p-verdi som er mindre enn begge disse verdiene. Jeg kan derfor kan jeg med høy sikkerhet forkaste nullhypotesen, altså bekrefter dette at det er statistisk signifikant sammenheng mellom de inntekt og utdanning.

Alderskolonnen har på den andre siden kun 0.1% manglende data, noe som gjør det mindre kritisk, men fortsatt viktig, å velge en hensiktsmessig imputasjonsmetode. Det er rimelig å anta at alderen til pasient vil til en grad henge sammen med både alder og utdanning, da yngre individer ofte har begrenset utdanning og lavere inntekt. For å bekrefte denne antagelsen, gjennomførte jeg en ny chi-kvadrat-test for denne gangen vurdere den statistiske sammenhengen mellom alder og utdanning. Resultat viser en høy chi2-verdi og lav p-verdi, som igjen indikerer en signifikant sammenheng mellom alder og utdanning.

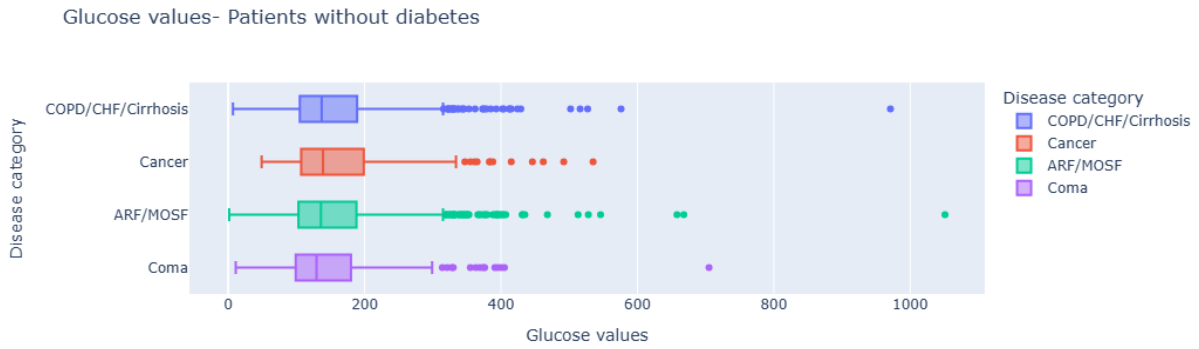
	Chi-square stat (chi2)	p-verdi
Utdanning og inntekt	1047.04	4.2133198336971265e-166
Utdanning og alder	134000.82	0.023231212541368444

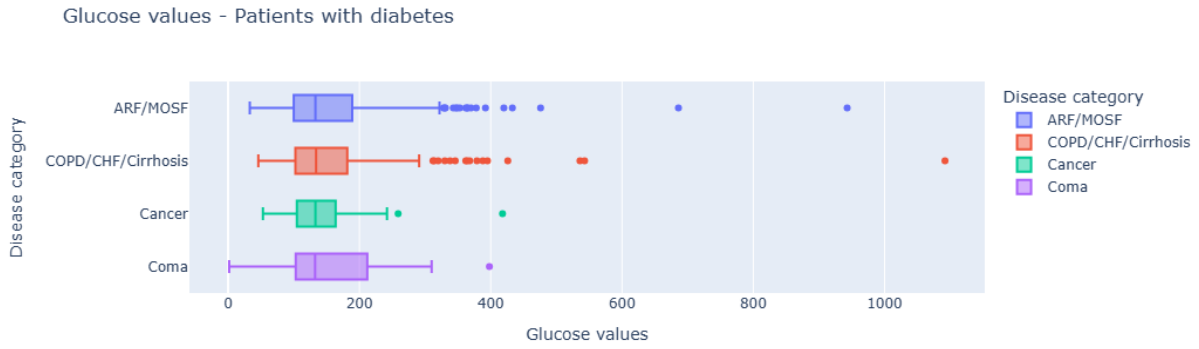
Basert på dette velger jeg å imputere alder og inntekt kolonnen sammen med utdanningskolonnen. Før imputasjonen konverterer jeg inntektskolonnen til numeriske verdier. Deretter benytter jeg K-nærmeste nabo-imputasjon (KNN) med én nærmeste nabo for å estimere de manglende verdiene i begge kolonner.

Glukose

Glukose kolonnen mangler 49% av data. For å undersøke om de manglende dataene er tilfeldig fordelt, lager jeg en krysstabell for å vurdere sammenhengen mellom manglende glukose verdier og diabetesstatus. Ved hjelp av krysstabellen oppdager jeg at det er en betydelig andel manglende glukose verdier i begge gruppene, noe som antyder at dataene ikke manglet helt tilfeldig. Jeg visualiserer fordelingen av glukoseverdier med histogrammer og ser at det er stor spredning i målingene. Begge gruppene, både pasienter med og pasienter uten diabetes, viste en konsentrasjon rundt medianen og en tung høyre hale. På grunn av den skjeve spredningen i begge gruppene, velger jeg å imputere de manglede verdiene med median. Median er mer robust mot ekstreme verdier og skjevfordelinger enn gjennomsnittet noe som gjør det til et bedre imputasjonsstrategi i dette tilfelle. Jeg fyller inn manglende verdier for de som har diabetes med medianen for denne gruppen, og tilsvarende for pasienter uten diabetes.

	Mangler glukose måling	Har glukose måling
Ikke diabetes	2185	2151
Diabetes	475	602





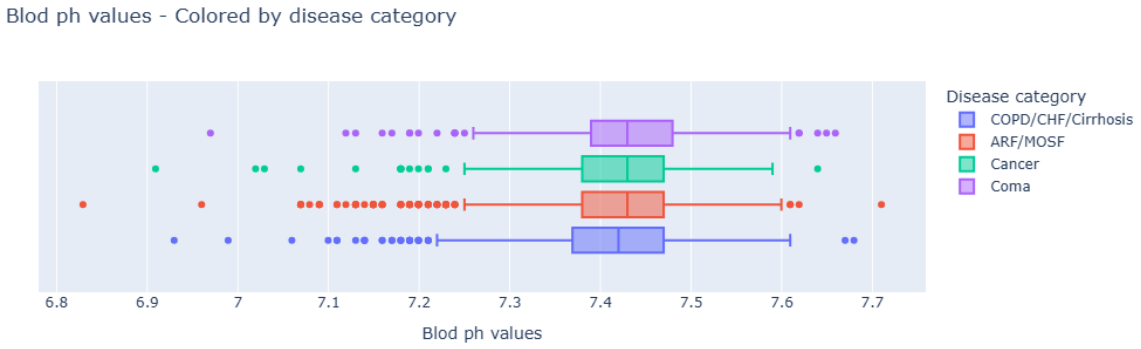
Blod-pH

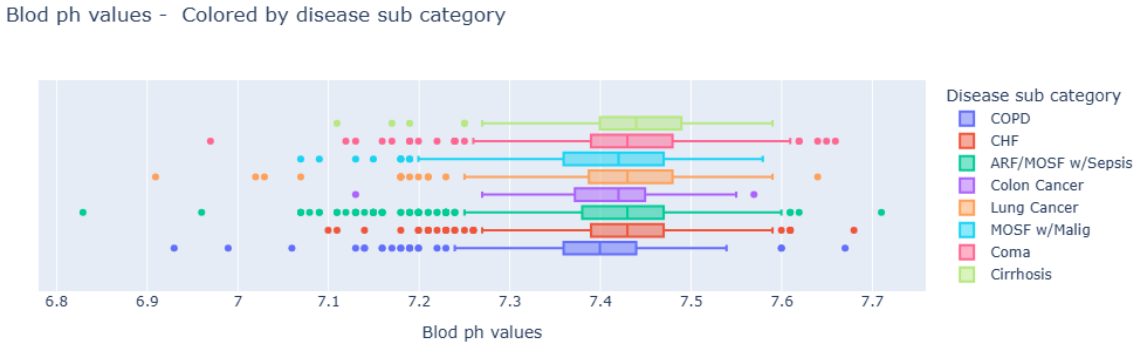
Blod-pH kolonnen mangler 25% av data. I henhold til data dokumentasjonen kan unormale verdier for blod-pH skyldes ulike pasient tilstander, inkludert lungesykdom, nyresykdom, metabolske sykdommer, medisiner eller skader som påvirker pustefunksjon (f.eks. hode- eller nakkeskader). Siden jeg ikke har informasjon om pasienten har metabolske sykdommer, medisiner eller skader som kan påvirke pH-nivåer undersøker jeg de spesifikke sykdom kategoriene som er relatert til lunge – og nyrefunksjon.

Tabellen nedenfor gir en oversikt over sykdomskategoriene og underkategoriene som er knyttet til lungesykdommer og nyresykdommer.

	Lungesykdom	Nyresykdom
Sykdoms kategori		
"ARF/MOSF" (Akutt respirasjons svikt / multiple organ failure)	x	x
"KOLS/CHF/Cirrhose" (Lunge/ Akutt hjerte svikt/ leversykdom)	x	
«Kreft"		
"Koma".		
Sykdom underkategori		
"ARF/MOSF m/Sepsis"	x	x
"CHF" (Akutt hjerte svikt)		
"KOLS" // COPD	x	
"Cirrhose"		
"Tykktarmskreft"		
"Koma"		
"Lungekreft"	x	
"MOSF m/Malig" (Multiple organ failure)	x	x

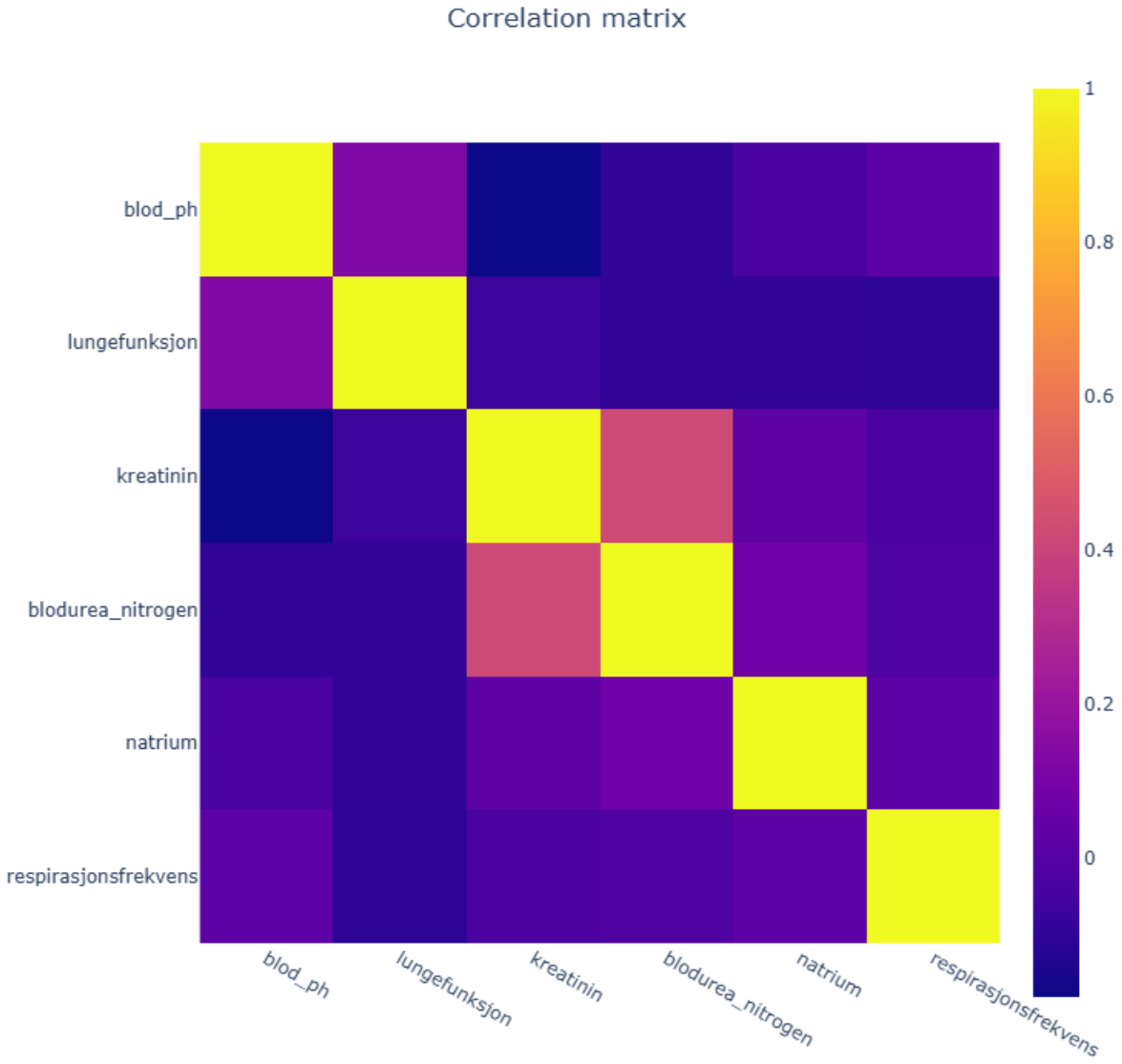
For hver av disse sykdomskategoriene og underkategoriene henter jeg ut blod-pH verdier. Jeg fant at median og gjennomsnittet til blod-pH verdiene var tilnærmet like på tvers av disse kategoriene, med små variasjoner. Dette antyder at lunge og nyresykdommer ikke har en signifikant innvirkning på blod-pH. Videre bekrefter jeg dette med visualiseringer som viser at fordelingen på tvers av sykdoms kategorier og sykdoms under kategoriene er svært lik.





I datasettet har vi fått flere variabler som gir en indikasjon på pasientens lunge -og nyrefunksjon. Basert på datadokumentasjonen identifiserte jeg følgende variabler som relevante: lungefunksjon, kreatinin, blodurea nitrogen, natrium, og respirasjonsfrekvens. Disse variablene måler i forskjellig grad lunge og nyrefunksjon til pasienten og jeg bruker de derfor videre for å finne den beste imputasjons strategien til blod -pH.

For å vurdere om noen av disse variablene korrelerer med blod-pH, lager jeg en korrelasjonsmatrise. Jeg ser at det er lav korrelasjon mellom blod-pH og de utvalgte variablene. Dette indikerer at ingen av de variabelen som indikerer lunge- og nyrefunksjonen til en pasient har sterk lineær sammenheng med blod-pH.

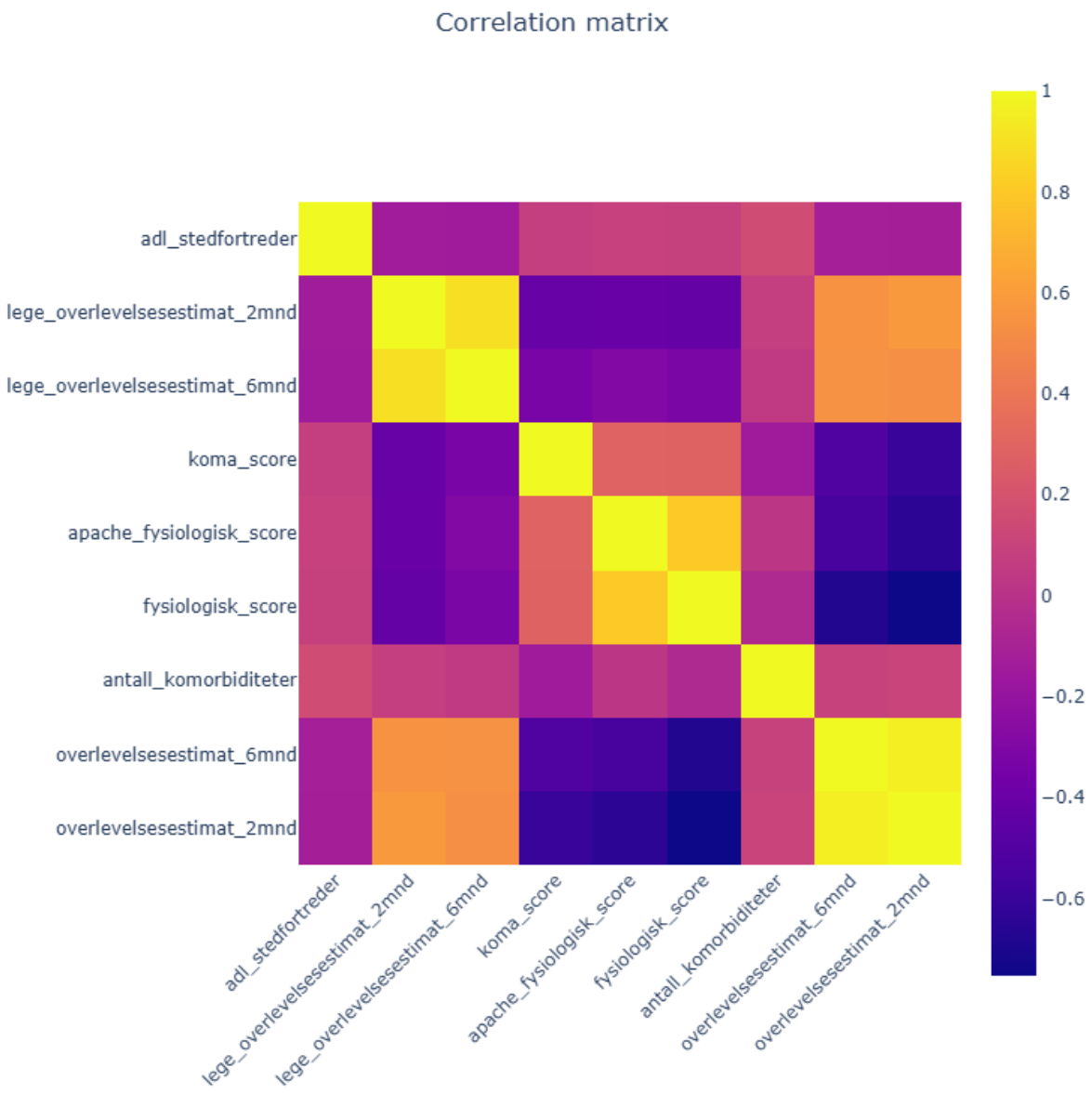


Siden ingen av variablene alene viste sterk korrelasjon med blod-pH, velger jeg å kombinere dem og benytte K nærmest nabo imputasjons strategi (KNN-imputer) for å fylle inn manglende verdier for blod-pH. Det er viktig å bemerke at KNN imputering vil fylle inn manglende verdier i alle kolonner vi bruker, men i data settet er det kun blod-pH som mangler verdier. Dette betyr at KNN imputering vil påvirke kun manglende verdier mens verdier som ikke mangler vil forbli uendret. Dette fungerer til min fordel, i det tilfelle at noen av de andre variablene mangler i validerings og test data så vil de bli fylt ut her.

Jeg lager en funksjon som tester hvor mange nærmeste naboer (n_neighbors) som ga den beste prediksjonen for blod-pH. Resultat viste at en KNN-imputering med 1 nærmest nabo ga lavest RMSE for estimering av blod-pH. Bruker derfor denne imputasjons strategien videre til å fylle inn all manglende blod-pH verdier.

adl_stedfortreder, lege_overlevelsesestimat_2mnd, lege_overlevelsesestimat_6mnd

For å imputere de manglende verdiene i variablene adl_stedfortreder (32%), lege_overlevelsesestimat_2mnd (18 %), og lege_overlevelsesestimat_6mnd (18 %) starter jeg med å ta i bruk data dokumentasjonen. Ifølge den er alle tre variablene indikatorer på pasientens sjanse for overlevelse. I tillegg til de tre variable, inkluderer jeg variablene antall_komorbiditeter, overlevelsesestimat_2mnd, overlevelsesestimat_6mnd, koma_score, fysiologisk_score og apache_fysiologisk_score som alle på en aller annen måte reflekterer enten pasientens overlevelsessjanse eller sykdomsalvorligheten til pasienten.



For å undersøke sammenhengen mellom disse variablene lager jeg en korrelasjonsmatrise. Matrisen viser flere tydelig positive og negative korrelasjoner mellom flere variabler. Dette indikerer at mange av variablene har lineær sammenheng med hverandre. Resultatet fra korrelasjons matrise i kombinasjons med informasjonen jeg hentet ut fra data dokumentasjonen antyder at det kan være nyttig å bruke en kombinasjon av variablene til å imputere manglende verdier for de ulike overlevelses estimatene. Jeg velger derfor å bruke K-nærmeste nabo (KNN) som imputasjonsstrategi fro de manglende variablene (adl_stedfortreder, lege_overlevelsesestimat_2mnd, lege_overlevelsesestimat_6mnd).

Jeg anvender den samme funksjonen som tidligere for å identifisere hvilken verdi for hyperparameteren n_neighbors som gir lavest RMSE ved imputering. Resultat viser at selv om

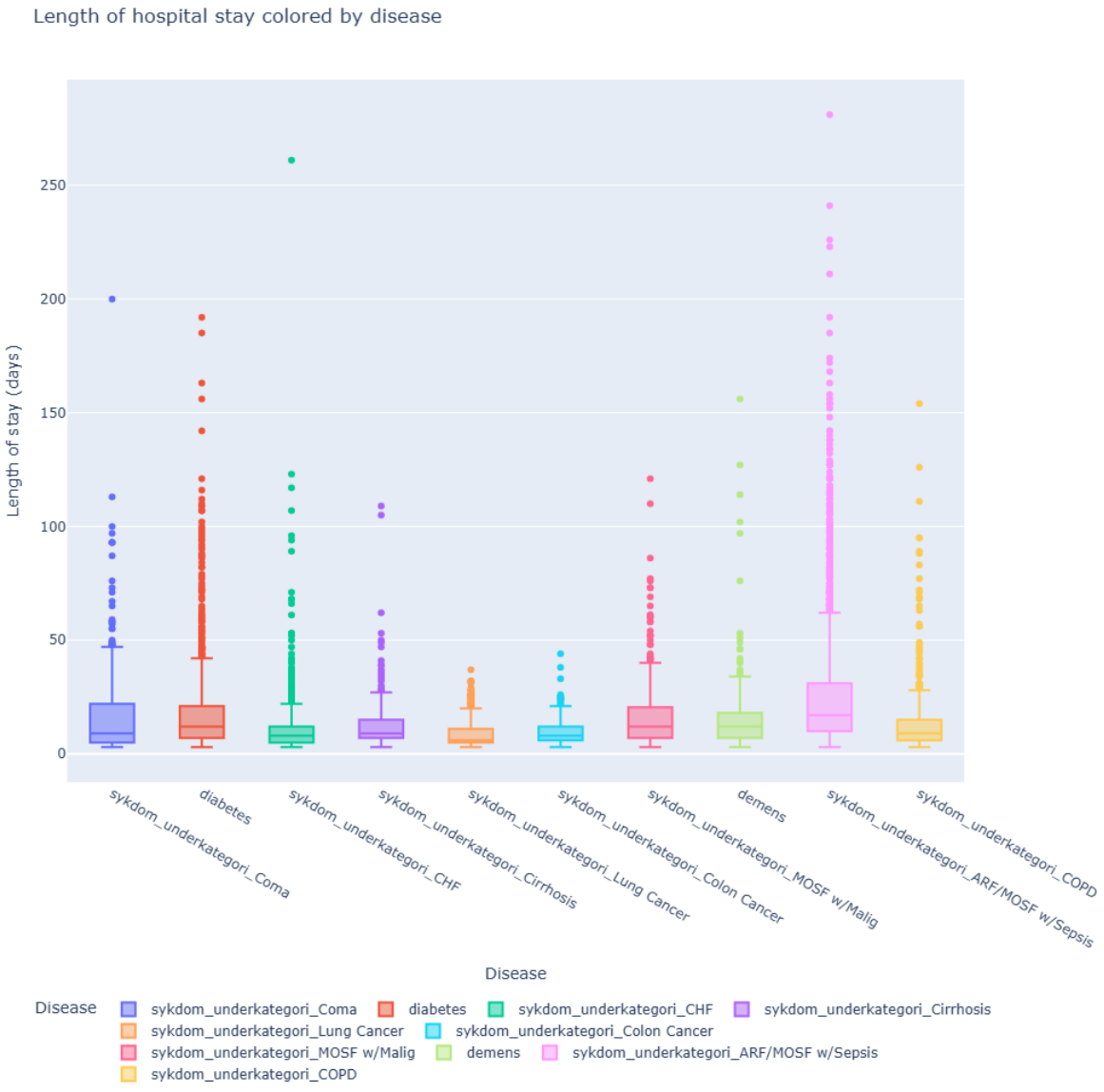
forskjellene i RMSE mellom de ulike verdier for n_neighbors er små, oppnås den laveste RMSE når antallet naboer settes til 6 naboer. Derfor velger jeg denne imputasjonsstrategi for som benytter informasjon om nærliggende verdier, baser på relevante variabler imputere manglende variabler, spesielt siden variablene viser en sterk lineær korrelering.

1.7 Transformasjon av kategoriske variabler til numeriske verdier

For å forberede dataen til modelltrening, transformerer jeg alle objekt kolonnene til numeriske verdier ved hjelp av OneHotEncoder. Denne prosessen konverterer kategoriske variabler til såkalte «dummyvariabler» der hver kategori representeres som en egen kolonne med binære verdier (0 eller 1). Jeg tilpasser OneHotEncoder til treningsdataen og deretter anvender den på validerings og test data for å sikre konsistens. Denne variable utvinningen gjør det mulig å bruke kategoriske variabler i modell treningen senere og sikrer at modellen kan tolke og prosessere alle variabler på en numerisk skala.

1.8 Visualisering av sykdomskategorier og oppholdslengde

Nå som hver kategori representeres som egen kolonne, kan jeg enkelt visualisere de ulike kategoriene. Jeg lager en figur som viser fordelingen av oppholdslengden blant de forskjellige sykdomskategoriene. Figuren viser en viss variasjon i oppholdslengden, noe som kan være interessant å undersøke nærmere. Det blir spesielt relevant å se hvilke av variablene de ulike modellene senere vil vurdere som viktige for å forklare oppholdslengden.



2. Modellering

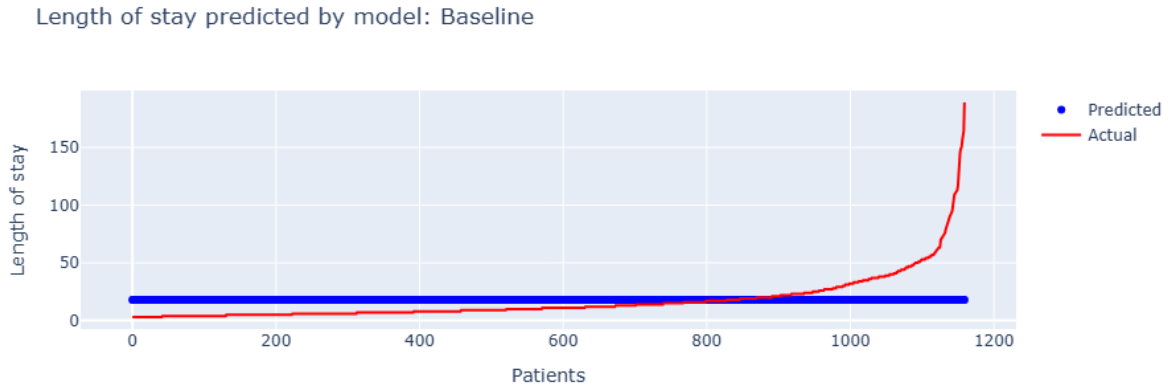
2.0 Separasjon av variabler og oppholdslengden før modelltrening

Før jeg begynner med å trene og validere modeller, skiller jeg variablene (X) fra målvariabelen (y), som i dette tilfelle er oppholdslengden. Dette er et nødvendig steg for å sikre at modellene kun trenes på variablene uten noe informasjon om oppholdslengden lekkes inn i trenings- og/eller prediksjonsprosessen.

2.1 Grunnlinjemodell

Jeg lager først en grunnlinjemodell, denne vil fungere som referanse for å vurdere ytelsen til de andre modellene. I dette prosjektet skal jeg bruke RMSE (gjennomsnittlige kvadrerte feilen) for å evaluere modeller. Og baseline modellen er essensiell for å kunne sammenligne nøyaktigheten i prediksjonen for oppholdslengden

Grunnlinje modellen har en RMSE på 20.6 dager. Dette betyr at modellen i gjennomsnitt predikerer oppholdslengden med et avvik på 20.6 dager. Dette avviket er relativt stort, særlig om jeg tar i betraktning at den gjennomsnittlig oppholdslengden er 17 dager, samt en enda lavere median på 11 dager. Den høye RMSE-verdien indikerer at modellen har begrenset evne til å forutsi oppholdslengden presist, noe som kan tilskrives den store spredning i dataen vi observerte i utforskende dataanalysen. Noe som skyldes at grunnlinje modellen alltid predikerer gjennomsnittet.

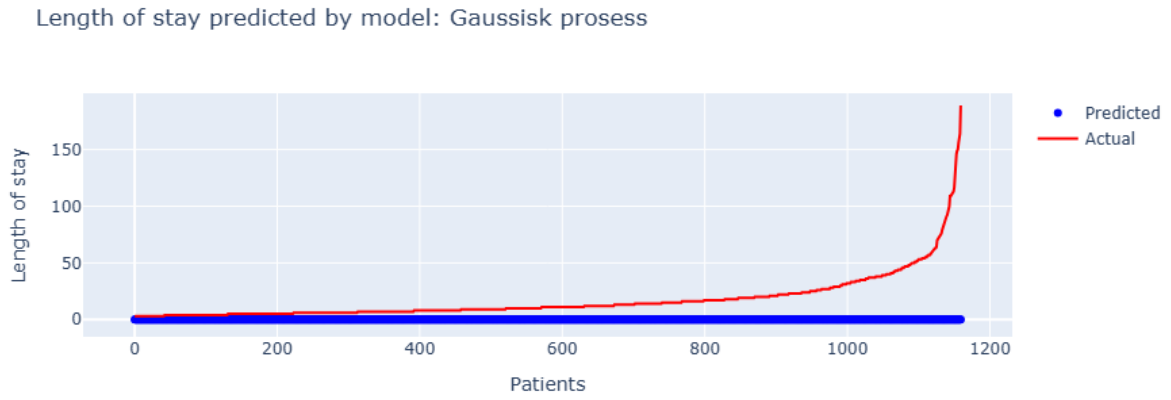


2.2 Generell modellering og visualisering av prediksjoner

Etter å ha trent flere modeller, får jeg følgende RMSE-verdier. Resultatene er lovende, da 7/10 modeller allerede presterer bedre enn grunnlinjemodellen, som har RMSE på 20.6 dager. For å videre evaluere hvor godt modellene predikerer oppholdslengden, visualiserer jeg prediksjonene.

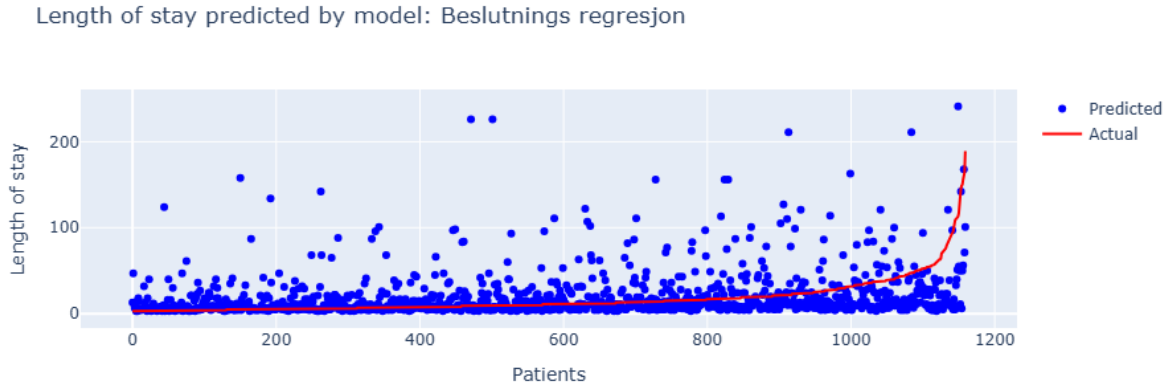
Modeller	RMSE original
Ridge	19.09
Lineær Regresjon	19.10
Nevrale nett	19.46
Ekstra skog regresjon	19.56
Lasso regresjon	19.60
Elastisk nett regresjon	19.64
Tilfeldig skog regresjon	19.82
Support vektor regresjon	**21.50**
Gaussisk prosess	**26.97**
Beslutnings regresjon	**30.18**

Gaussian prosess regresjon presterer imidlertid dårlig, med en RMSE på 26.97 dager sammenlignet med grunnlinjen. Dette indikerer at modellen ikke generaliser godt til dataene, noe som forårsaker at den predikerer nær en konstant verdi, i dette tilfelle null. Selv om det kan være mulig å forbedre modellen gjennom hyperparamerter-tuning, for eksempel ved å justere kjernes parameter og skalere dataen, velger jeg å forkaste denne modellen fra videre analyser.

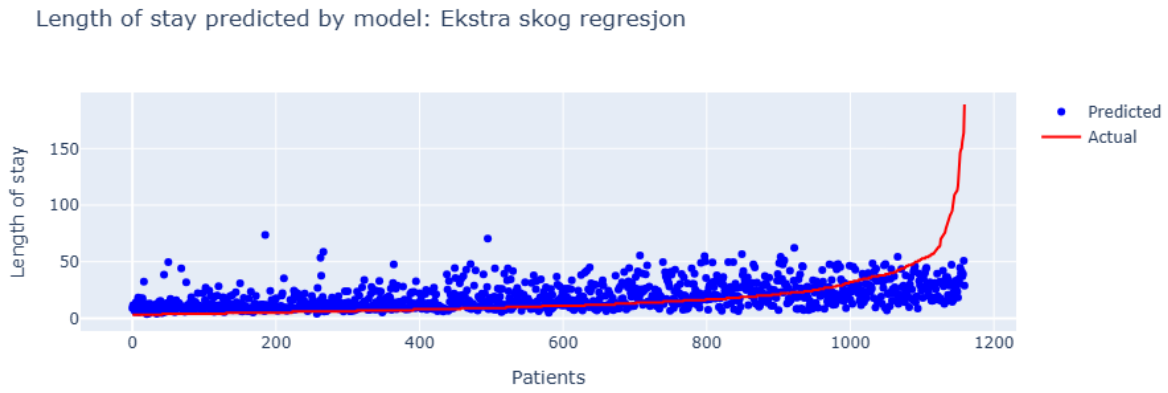


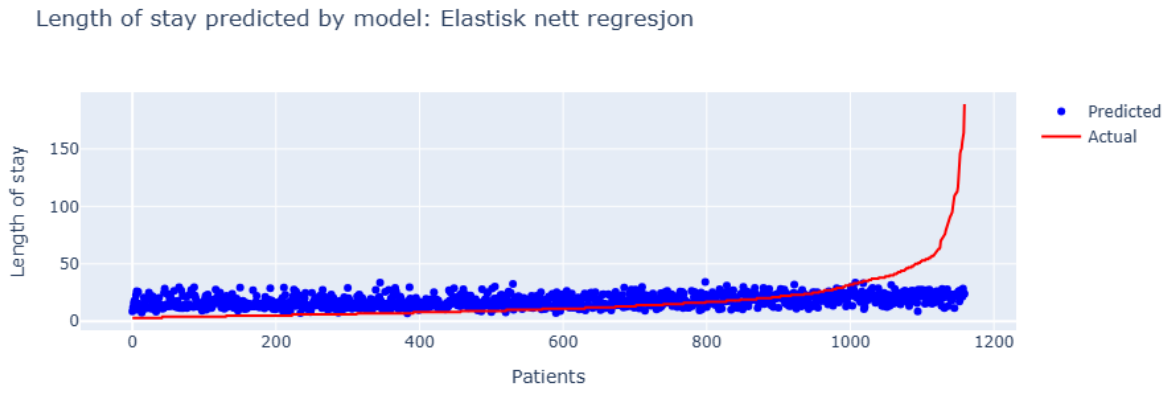
Det samme viser Support Vektor Regresjon (SVR) som oppnår en RMSE på 21.5 dager. Visualiseringen av prediksjonen indikerer at modellen predikerer tilnærmet samme verdi hver gang. Skjevfordeling og tilstedeværelsen av ekstremverdier i trenings dataen kan være årsaken til at SVR modellen er like sensitiv til variasjon av de ulike variablene, som i gjengjeld resulterer i homogene prediksjoner. Selv om SVR også kan hyperparamerter-tunes, velger jeg å ikke bruke denne modellen videre, grunnet modellen begrenset evne til å fange opp kompleksiteten i data i kontrast med andre modeller som allerede demonstrerer mer lovende resultater.

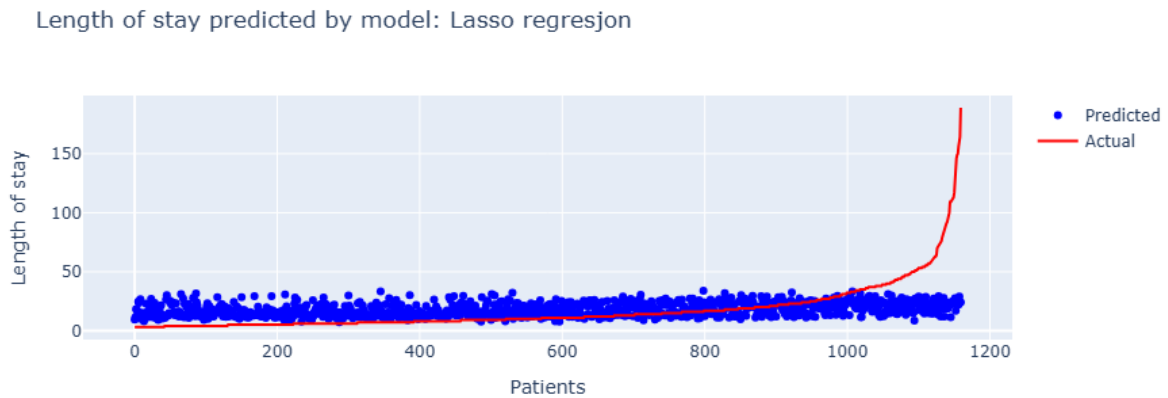
Beslutnings regresjon modellen (DecisionTreeRegressor) har den dårligst ytelsen med en RMSE på 31.18. Visualiseringen bekrefter også dette, da modellen ofte predikerer veldig høye verdier. Siden flere modeller presterer betydelig bedre, velger jeg å se vekk fra denne modellen.



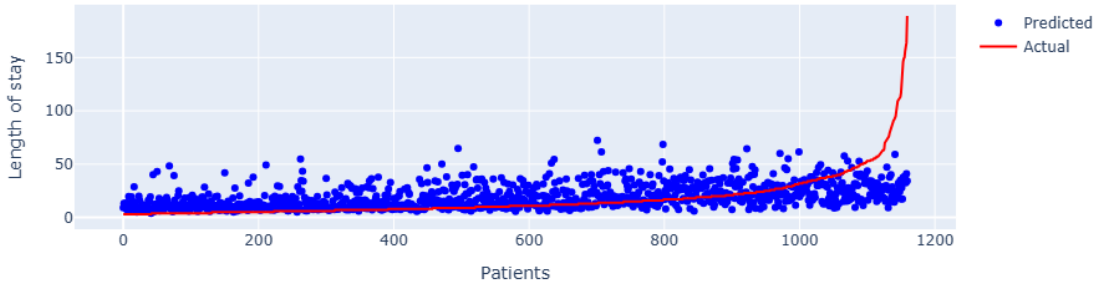
Prediksjonen for de gjenværende modellen ser noe lik ut. Ingen av modellene klarer å predikere ekstremalverdier riktig. De fleste modeller ser ut til å unngå å predikere ekstremverdier og predikerer i stor grad alltid mellom 0 – 50 dager. Det er også interessant å merke seg at flere av modellene, blant annet Lineær regresjon, Nevrale nett og Ridge predikerer negative verdier for oppholdslengden. Generelt ser det ut til at modellene gir sammenlignbare resultater, noe som også bekrefter av RMSE-verdien i tabellen.



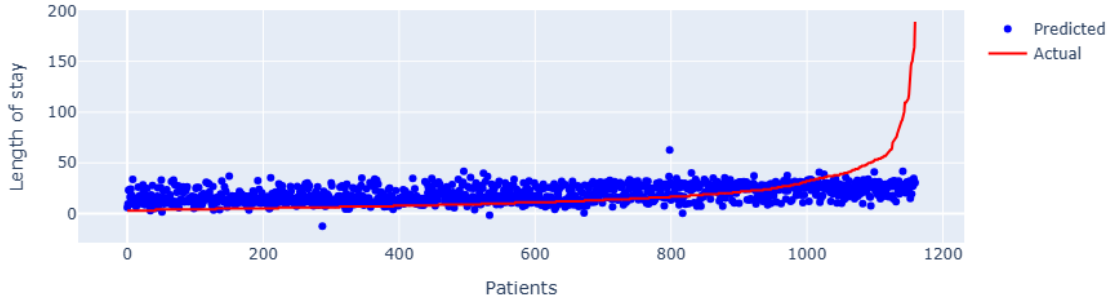




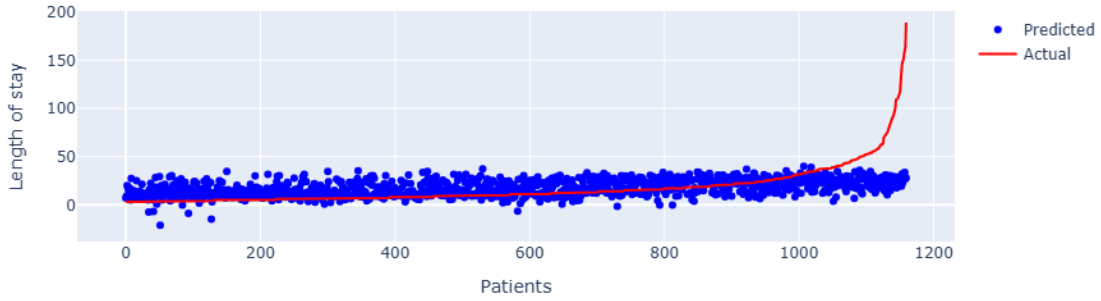
Length of stay predicted by model: Tilfeldig skog regresjon



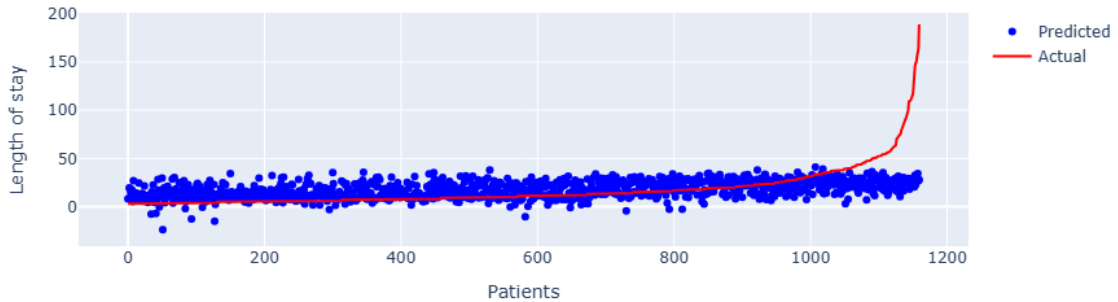
Length of stay predicted by model: Nevrale nett



Length of stay predicted by model: Ridge



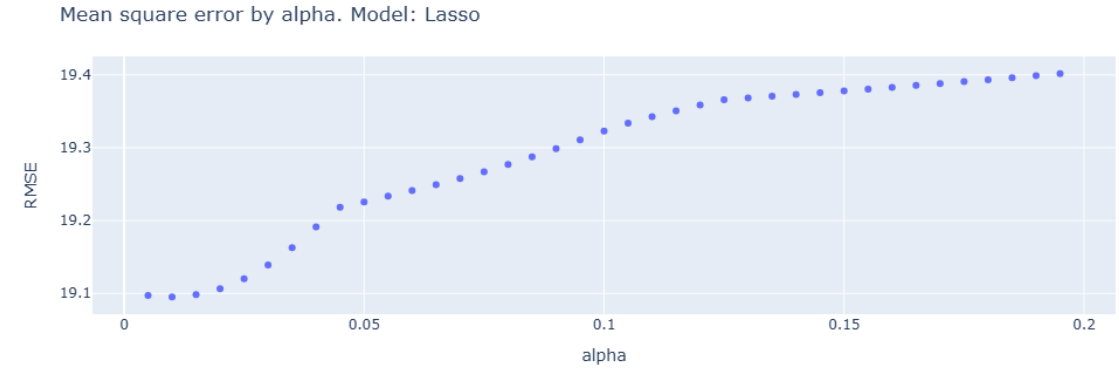
Length of stay predicted by model: Lineær Regresjon

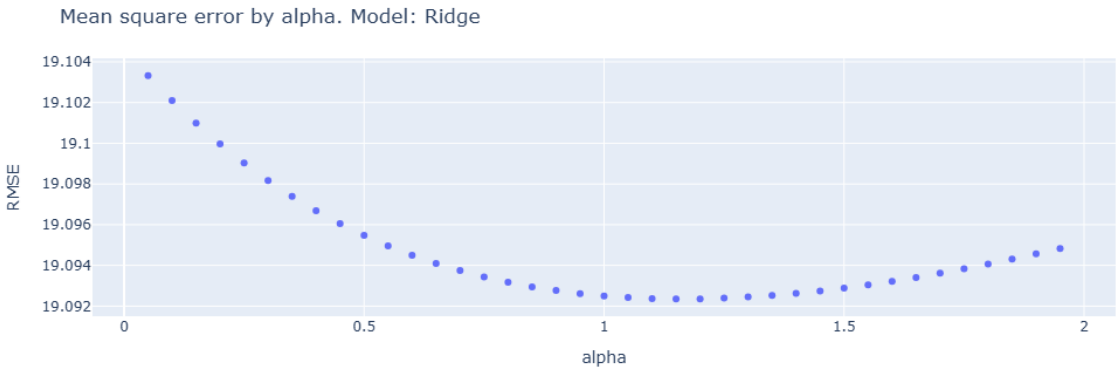


2.3 Effekt av hyperparamerter alpha i Lasso-, Ridge og Elastisk nett Regresjon.

Jeg lager en funksjon som trener og beregner RMSE for Lasso-, Ridge- og Elastisk nett modeller med varierende verdier for hyperparameteren alpha. Visualisering av resultatet viser at Lasso-modellen oppnår et bunnpunkt hvor RMSE er lavest (19.09 dager) med en alpha-verdi på 0.01. Ridge modellen har en tilsvarende RMSE (19.09 dager) når alpha er satt til 1.15. Elastisk nett regresjon har en litt høyere RMSE (19.17 dager) når alpha er 0.005. Hyperparamerter-tuningen viser ingen reduksjon i RMSE for Ridge modellen, men ved å optimaliser alpha-verdien for Lasso og Elastisk nett modellene reduserer jeg RMSE betydelig.

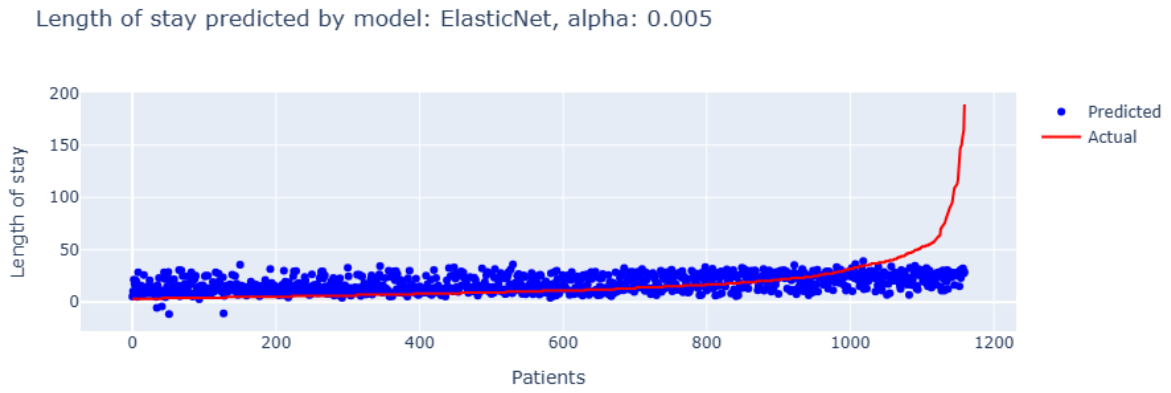
Modeller – Alpha hyperparamerter	RMSE original	RMSR alpha
Ridge	19.09	19.09
Lasso regresjon	19.60	19.10 ▼
Elastisk nett regresjon	19.63	19.17 ▼

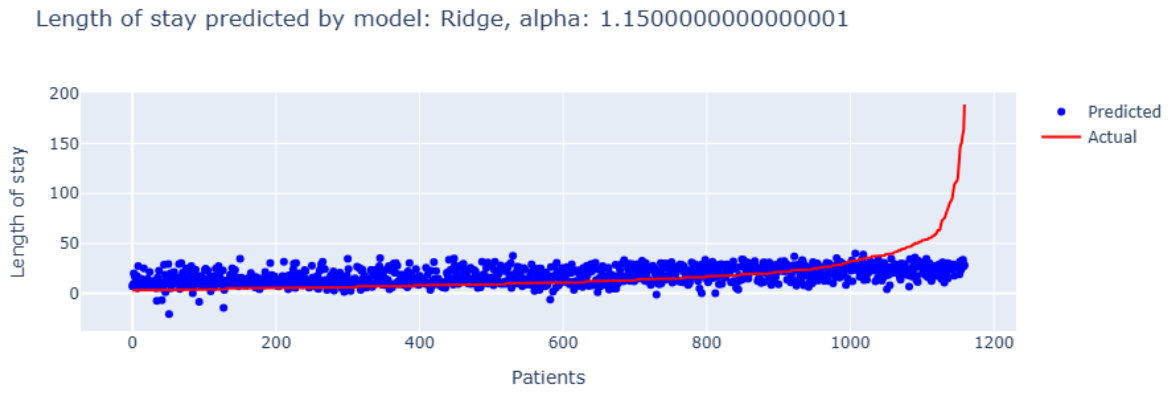


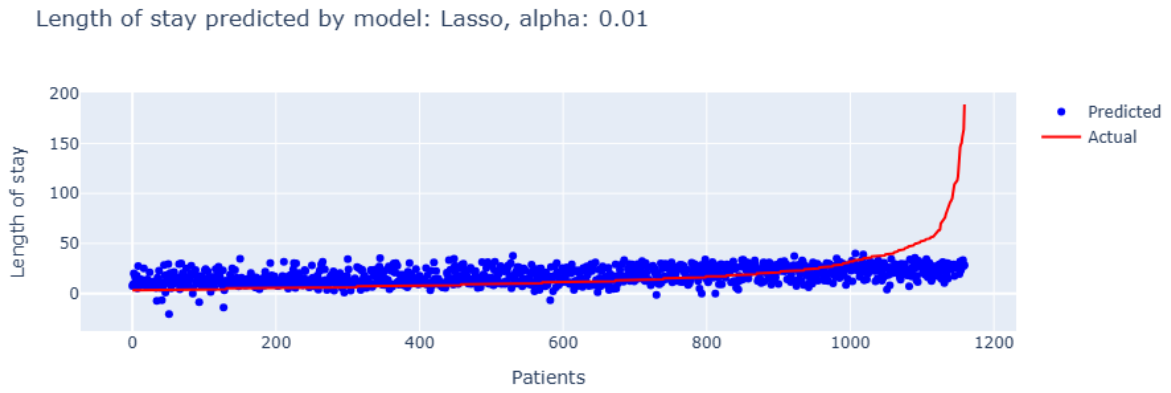


Videre viser visualisering at prediksjonen til Lasso, Ridge og Elastisk nett regresjon er nærmest identiske. Alle tre modellen forsøker å finne best mulige koeffisienter og reduserer overfitting med å straffe store koeffisienter, men de gjør det på forskjellige måter. Ridge-regresjon redusere

størrelsen på koeffisientene uten å nødvendigvis å sette dem til null, mens Lasso eliminere enkelte koeffisienter helt, noe som resulterer i en modell med færre funksjoner, også kjent som feature selection. Elastisk nett modellen gjør en kombinasjon av begge strategiene. Dermed inkluderer Ridge og Elastisk nett vanligvis alle variabelen i den endelige modellen, noe som er hensiktsmessig om jeg antar at alle variabelen har en viss effekt på oppholdslengden. Lasso derimot, kan fullstendig fjerne variabler som identifiseres som unødvendige.







Gitt at datasettet inneholder mange variabler, og ettersom jeg ønsker å identifisere hvilke av disse som faktisk er signifikant for oppholdslengden til pasienter, vil Lasso-modellen være mer gunstig i denne sammenhengen. Dette i kombinasjon av at RMSE til alle tre modellen er tilnærmet lik, og predikasjonen av oppholdslengden er nærmest identiske velger jeg å velge vekk Ridge og Elastisk nett modellen fra videre analyser.

2.4 Effekt av ulike polynomgrader

Innledningsvis trente jeg kun en lineær regresjonsmodell, som ga en RMSE på 19.1 dager. For å undersøke muligheten for å forbedre modellytelse, har jeg nå evaluert en direkte generalisering av lineær regresjon ved å teste ulike polynomgrader. Polynomgraden betraktes i denne sammenhengen som en hyperparamerter. Etter å ha evaluer polynomgradene 1,2,3, finner jeg at førstegradspolynomet, som er identisk med lineær regresjon, gir lavest RMSE. Dette bekrefter at modellen jeg først trente, er den best mulige i dette tilfelle. Det er verdt å merke seg at selv om høyere grad polynomer kan fange opp mer kompleksitet i dataene, vil ikke dette nødvendigvis resultere i bedre ytelse. Dette er tilfelle nå. Jeg kan med sikkerhet bruke lineære regresjon for videre modellering av oppholdslengden.

2.5 Effekt av persentil klipping

Klipping av y -verdier

En alternativ tilnærming til håndtering av outliers er å klippe y-verdiene (oppholdslengden) i treningsdataen på bestemte persentil. Dette innebærer å modifisere oppholdslengden for pasienter som overstiger en valgt persentil, slik at disse ekstreme verdiene blir satt lik den valgte persentil verdien. Jeg mistenker at ekstremalverdiene i datasettet kan ha en negativ innvirkning på modellenes ytelse. For å evaluere denne hypotesen lager jeg en funksjon som klipper trenings data ved ulike persentiler. Jeg testet først et bredt intervall mellom 75-100%, og dette indikerte at klipping rundt persentilene mellom 95% og 98% ga de best resultatene. Jeg har derfor modifisert funksjonen slik at den kun tester dette intervaller for å redusere kjøretiden. Etter klipping av y-verdiene ved 96% og påfølgende ny trening av modellene, observerer jeg at Ekstra skog regresjon og Tilfeldig skog regresjon oppnår betydelig bedre RMSE. Klipping av topp 4% y-verdier ser derfor ut til å redusere det gjennomsnittlige avviket i modellprediksjonene for disse modellene, sannsynligvis fordi treningsdataene inneholder flere ekstreme outliers.

Modeller – Persentil klipp (96%) y-verdier	RMSE før	RMSE etter
Ekstra skog regresjon	19.54	**18.84 ▼**
Tilfeldig skog regresjon	19.82	**18.91 ▼**
Lineær Regresjon	**19.10**	19.11 ▲
Nevrale nett	**19.47**	19.60 ▲
Lasso regresjon	**19.09**	19.77 ▲

Det er imidlertid viktig å diskutere hvor etisk det er å ekskludere pasientdata som overstiger 96%. Ved å ekskludere pasientdata som overstiger 96% persentilene, reduseres modellens potensial til å gi presise prediksjoner for lengre opphold. Til tross for dette velger jeg å ta med denne innsikten i videre analyser, ettersom den gir en klar forbedring i RMSE.

Klipping av X - verdier

Videre testet jeg også effekten av klipping på X verdiene basert på persentiler. I funksjonen tester alle mulige kombinasjoner av persentil nedre persentil klipping mellom 0-5% og øvre persentil klipping mellom 75-100% Resultater fra denne tilnærminger viste imidlertid ingen forbedring i RMSE for noen av modellene, noe som kan indikere at ekstremverdiene blant input variablene ikke påvirker modellens prediksjoner betydelig. Det er også mulig å velge spesifikke variabler for klipping, men på grunn av manglende forbedring ved generell klipping velger jeg å utelate denne tilnærmingen fra videre analyser.

Modeller – Persentil klipp (10% & 95%) X- verdier	RMSE før	RMSE etter
Ekstra skog regresjon	**18.83**	19.73 ▲
Tilfeldig skog regresjon	**18.91**	19.80 ▲
Lineær Regresjon	**19.10**	19.20 ▲
Nevrale nett	**19.47**	22.19 ▲
Lasso regresjon	**19.09**	19.71 ▲

2.6 Effekt av intervallaggregasjon

I et forsøk på å redusere støy i dataene og forbedre modellenes ytelse, undersøker jeg muligheten for å aggregere verdiene i de ulike kolonnene etter forhåndsbestemte intervaller.

Disse intervallene utarbeider jeg basert informasjonen og vedlagte lenker i datadokumentasjonen samt eksterne kilder fra nett. Jeg inkluderte kolonnene lungefunksjon (Nickson, 2024), blod-pH (Harrel, 2022), kroppstemperatur (Wikipedia contributors, 2024), hvite blodlegemer (Wikipedia contributors, 2024), serumalbumin (Wikipedia contributors, 2024), natrium (Wikipedia contributors, 2024) og kreatinin (Berman & Dugdale, 2023). Dessverre viste denne tilnærmingen ingen / lite forbedring i modellprestasjonene.

Modeller – Aggregasjon	RMSE før	RMSE etter
Lineær Regresjon	19.10	**19.08 ▼**
Nevrale nett	19.47	**19.40 ▼**
Lasso regresjon	**19.09**	19.62 ▲
Ekstra skog regresjon	**18.83**	19.65 ▲
Tilfeldig skog regresjon	**18.91**	19.75 ▲

Jeg forsøker også å trene modellene på de aggregerte X-verdiene, der y-verdiene var klippet til 96% persentilen. Til tross for at klipping av oppholdslengden tidligere hadde vist seg å gi gode resultater for Ekstra skog regresjon og Tilfeldig skog regresjon, viste denne tilnærmingen ingen forbedring. Faktisk resulterte aggregeringen av X-verdiene i en høyere RMSE, jeg velger derfor å se vekk fra intervallaggregeringen i videre analyse.

Modeller – Aggregasjon og persentil klipp (96%)	RMSE før	RMSE etter
Lineær Regresjon	**19.10**	19.12 ▲
Nevrale nett	**19.40**	20.11 ▲
Lasso regresjon	**19.09**	19.80 ▲
Ekstra skog regresjon	**18.83**	18.87 ▲
Tilfeldig skog regresjon	**18.91**	19.00 ▲

2.7 Effekt av data skalering

I et forsøk på å forbedre modell ytelse og redusere effekten av ekstreme verdier, utfører jeg en sammenligning av tre vanlige skaleringsmetoder: standardisering, normalisering og robust skalering. Standardisering (StandardScaler) sentrerer dataen rundt gjennomsnitt (0) og normaliserer til et standardavvik på 1. Normalisering (MinMaxScaler) transformerer dataen innenfor et vis intervall (vanligvis 0-1). Og Robust skalering (RobustScalar) skalerer verdiene med median og interkvartilområder (IQR) istedenfor gjennomsnitt og standardavviket. Robust skalering er derfor mer rustet til å håndtere outliers, og gjør dataen mindre følsom for ekstremal verdier.

Resultatet viser forbedret ytelse hos Nevrale nett etter skalering, med en reduksjon i RMSE (fra 19.47 til 18.75). For andre modeller som Ekstra skog og Tilfeldig skog -regresjon ser jeg ingen reduksjon i RMSE etter skalering, noe som kan tilskrives modellenes innebygde robustheter mot skala. Jeg undersøker også kombinasjon av å skalere X-verdier samtidig som jeg klipper y-verdiene ved 96% persentil, som tidligere har forbedret resultatene. Denne strategien ga igjen positiv effekt for Nevrale nett modellen, men påvirket ikke de andre modellene betydelig. Samlet sett indikere dette at kun Nevral nett modellen har fordel av dataskalering.

Modeller – Skalert X	RMSE før	RMSE Robust	RMSE MinMax	RMSE Standard
Lineær Regresjon	**19.10**	19.10	19.11	19.12
Nevrale nett	19.40	19.07	**18.75**	19.59
Lasso regresjon	**19.09**	19.65	19.65	19.50
Ekstra skog regresjon	**18.83**	19.56	19.56	19.56
Tilfeldig skog regresjon	**18.91**	19.82	19.82	19.82

Modeller – Skalert X og klippet y -verdier	RMSE før	RMSE Robust	RMSE MinMax	RMSE Standard
Lineær Regresjon	**19.10**	19.11	19.11	19.13
Nevrale nett	19.47	18.86	**18.75**	19.05
Lasso regresjon	**19.09**	19.84	19.84	19.65
Ekstra skog regresjon	**18.83**	18.83	18.83	18.83
Tilfeldig skog regresjon	**18.91**	18.92	18.91	18.91

2.8 Hyperparamerter tuning og feature selection av Tilfeldig skog regresjon.

Basert på tidligere analyser vet jeg at Tilfeldig Skog regresjon oppnår en positiv effekt på RMSE ved å klippe oppholdslengden på 96% persentilen. For å videre optimaliser modellen, utfører jeg et grid search på flere hyperparamerter, både med og uten klipte y-verdier.

Beste hyperparamerter fra grid search uten klipte y -verdier:

```
Best hyperparameters for Tilfeldig skog regresjon (clip=False):  {'bootstrap': True, 'max_depth': 50, 'max_features':
'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 500, 'random_state': 42}
```

Beste hyperparamerter fra grid search med klipte y-verdier:

```
Best hyperparameters for Tilfeldig skog regresjon (clip=True):  {'bootstrap': False, 'max_depth': 20, 'max_features':
'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 250, 'random_state': 42}
```

Jeg ser at det er noen små forskjeller i hyperparamerter avhengig av om y-verdiene var klipte eller ikke. Til tross for omfattende grid search, viser RMSE liten forbedring i modell ytelsen. Optimaliseringen av hyperparamerter gir noe reduksjon i RMSE, men resultatet indikere at persentil-klipping har en sterkere innvirkning på resultatene enn hyperparamerter tuningen. Jeg benytter hyperparamerter fra grid search med klipte verdier for videre analyser og trener modellen på klipte y-verdier siden denne tilnærmingen har lavest RMSE på 18.79.

Tilfeldig skog regresjon RMSE	Uten klipte y-verdier	Med klipte y-verdier
Uten hyperparamerter	19.82	18.91
Med hyperparamerter fra grid search	18.83	-
Med hyperparamerter fra grid search med klipte y-verdier	-	**18.79**

I et forsøk på å ytterligere optimalisere modellen, henter jeg ut de viktigste variablene og undersøker hvordan RMSE påvirkers ved å redusere totalt antall variabler i modell treningen. Ved å trene modellen på ulike antall av de mest viktige variablene, ser jeg at modellen får en stabil RMSE under 18.9 så lenge 15 eller fler variabler brukes. Jeg observerer også at modellen får en lavere RMSE når modellen trenes på kun 55 variabler.

Antall variabler	RMSE
10.0	19.16
15.0	18.94
20.0	18.90
25.0	18.92
30.0	18.83
35.0	18.80
40.0	18.80
45.0	18.81
50.0	18.80
55.0	**18.77**
Alle	18.79

Resultater indikerer at feature selection kan gi en modell med enklere tolking uten betydelig ytelses tap. I kontekst av dager gir 18.8 og 18.79 et likt avvik på 19 dager. Ved å redusere antallet variabler kan vi lage en modell med enklere tolkning.

Tabellen på neste under viser oversikt over de 20 viktigste variabelen til tilfeldig skog modellen.

Feature	Importance	Feature	Importance
fysiologisk_score	0.061250	hvite_blodlegemer	0.032084
overlevelsesestimat_2mnd	0.057972	blodtrykk	0.030835
overlevelsesestimat_6mnd	0.054540	kroppstemperatur	0.029318
apache_fysiologisk_score	0.052515	hjertefrekvens	0.028528
Alder	0.043506	lungefunksjon	0.028058
sykdom_underkategori_ARF/MOSF w/Sepsis	0.040861	respirasjonsfrekvens	0.026696
lege_overlevelsesestimat_2mnd	0.040199	natrium	0.025487
sykdomskategori_ARF/MOSF	0.040143	serumalbumin	0.025340
lege_overlevelsesestimat_6mnd	0.036154	blod_ph	0.024374
sykdomskategori_id_A1s	0.032328	kreatinin	0.024048

2.9 Hyperparamerter tuning og feature selection av Ekstra skog regresjon.

For å optimalisere modellen Ekstra skog regresjon utfører jeg et grid search for å finne den optimale kombinasjonen av hyperparamerter. Bildene nedenfor viser resultatene fra grid search med og uten klipping av y -verdier. Tidligere observasjoner indikerte at prosentilbasert klipping av y -verdier bidro til en lavere RMSE både for Ekstra skog og Tilfeldigskog regresjon, sannsynligvis fordi klipping reduserte effekten av ekstreme verdier og støynivå i dataen.

Beste hyperparamerter fra grid search uten klipte y -verdier:

```
Best hyperparameters for Ekstra skog regresjon (clip=False):  {'bootstrap': False, 'max_depth': 20, 'max_features':
'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 500, 'random_state': 42}
```

Beste hyperparamerter fra grid search meg klipte y -verdier:

```
Best hyperparameters for Ekstra skog regresjon (clip=True):  {'bootstrap': False, 'max_depth': 50, 'max_features':
'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 500, 'random_state': 42}
```

Resultatene viser at Ekstra skog regresjon oppnår lavere RMSE uten klipping, men med hyperparamerter tuning. Dette resultatet stemmer godt overens med forventningene, ettersom modellen Ekstra skog drar større nytte av hyperparamerter tuning sammenlignet med Tilfeldig skog, som har lavere følsomhet for slik tuning.

Ekstra skog regresjon RMSE	**Uten** klipte y-verdier	**Med** klipte y-verdier
Uten hyperparamerter	19.56	18.76
Med hyperparamerter fra grid search **uten** klipte y-verdier	**18.74**	-
Med hyperparamerter fra grid search **med** klipte y-verdier	-	18.79

Videre henter jeg ut de viktigste variablene til modellen ved hjelp av feature-importance-funksjonen. Tabellen under viser de 20 viktigste variablene.

Feature	Importance	Feature	Importance
sykdom_underkategori_ARF/MOSF w/Sepsis	0.057014	alder	0.026489
sykdomskategori_ARF/MOSF	0.047296	adl_stedfortreder	0.022747
sykdomskategori_id_A1s	0.044575	natrium	0.022221
overlevelsesestimat_2mnd	0.042372	blodtrykk	0.021599
lege_overlevelsesestimat_2mnd	0.041902	respirasjonsfrekvens	0.020911
overlevelsesestimat_6mnd	0.039805	serumalbumin	0.020836
lege_overlevelsesestimat_6mnd	0.037334	kreatinin	0.020828
fysiologisk_score	0.033706	kroppstemperatur	0.019338
apache_fysiologisk_score	0.033213	hjertefrekvens	0.019326
koma_score	0.02780	lungefunksjon	0.019130

For å undersøke robustheten til modellen, utfører jeg feature selection. Reduksjon i antallet variabler viser relativt lav påvirkning på modell ytelse, men en gradvis reduksjon i RMSE etter hvert som flere variabler blir inkludert. Modellen får også en lavere RMSE ved bruk av kun 50 variabler.

Antall variabler	RMSE
10	19.06
15	18.95
20	18.85
25	18.87
30	18.82
35	18.75
40	18.77
45	18.76
50	**18.73**
55	18.76
Alle	18.74

Resultatene viser at Tilfeldigskog regresjon og Ekstra skog regresjon har relativt lik ytelse. Slik jeg har forstått det benytter modellen lignende ensemble-strukturer, men skiller seg på hvordan de bygger beslutningstrær. Tilfeldig skog finner den beste splitten, noe som gir mer presise trær, men høyere varians. Ekstra skog derimot, velger tilfeldig split-punkt, noe som gir lavere varians, men høyere bias. Ekstra skog er derfor mer robust mot overtilpasning på datasett med høy varians og støy, grunnet randomiseringen. Gitt at jeg ønsker å velge en modell som er god på å generalisere ny data velger jeg å bruke Ekstra skog regresjon videre i analysen.

2.10 Hyperparamerter tuning og feature selection av Nevrale nett

Likt som tidligere finner jeg de beste hyperparametere for Nevrale nett modellen ved hjelpe av grid search. Tidligere observasjoner viste at skalering av X -verdier ga positiv effekt på Nevrale nettverk modellen. Jeg utfører derfor grid search, med og uten skalerte X-verdier for ytterligere undersøke hvordan skaleringen påvirker modellens ytelse.

Bilde av de beste hyperparamerter uten skalerte X-verdier:

```
Best hyperparameters for Nevrale nett (scale=False):  {'activation': 'relu', 'hidden_layer_sizes':
(50,), 'learning_rate': 'constant', 'max_iter': 200, 'random_state': 42, 'solver': 'adam'}
```

Bilde av de beste hyperparamerter med skalerte X-verdier:

```
Best hyperparameters for Nevrale nett (scale=True):  {'activation': 'relu', 'hidden_layer_sizes':
(100,), 'learning_rate': 'adaptive', 'max_iter': 200, 'random_state': 42, 'solver': 'sgd'}
```

Nevrale nettverk -modellen gir lavest RMSE med skalerte X-verdier. Dette stemmer overens med forventningene siden modeller er gode på å lære komplekse mønster og ikke lineære sammenhenger i store data sett. Dette gjør modellen godt egnet til å predikere oppholdslengde til pasienter.

Nevrale nett regresjon RMSE	**Uten** skalerte X-verdier	**Med** skalerte X-verdier
Uten hyperparamerter	19.47	18.76
Med hyperparamerter fra grid search **uten** skalerte X-verdier	19.73	-
Med hyperparamerter fra grid search **med** skalerte X-verdier	-	**18.58**

Slik som jeg har observert med flere modeller tidligere har ikke reduksjon i antall variabler ført til store ytelses tap for Nevrale nettverk modellen. Selv ved bruk av kun de 10 viktigste variablene presterer Nevrale nett modellen bedre enn Ekstra skog regresjon med 30 variabler. Selv om Nevrale nettverk modellen tradisjonelt er en «black box» modell, kan reduksjon av variabler øke tolkvarheten og bidra til å gjøre modellen den enklere å forstå.

Antall variabler	RMSE
10.0	18.76
15.0	18.75
20.0	18.63
25.0	18.66
30.0	18.62
35.0	18.67
40.0	18.59
45.0	18.59
50.0	**18.57**
55.0	18.58
Alle	18.58

Dette er de viktigste variablene for nevrale nett modellen:

Feature	Importance	Feature	Importance
lege_overlevelsesestimat_6mnd	0.028519	sykdomskategori_id_ChE	0.005778
lege_overlevelsesestimat_2mnd	0.027893	etnisitet_hispanic	0.004780
overlevelsesestimat_6mnd	0.021964	kjønn_male	0.004588
koma_score	0.020434	sykdomskategori_Cancer	0.004549
apache_fysiologisk_score	0.017207	kjønn_female	0.004461
overlevelsesestimat_2mnd	0.013810	antall_komorbiditeter	0.004417
alder	0.013575	diabetes	0.004412
sykdom_underkategori_ARF/MOSF w/Sepsis	0.011845	fysiologisk_score	0.004378
sykdom_underkategori_Lung Cancer	0.007217	adl_stedfortreder	0.003801
sykdom_underkategori_MOSF w/Malig	0.006815	sykdom_underkategori_Cirrhosis	0.003782

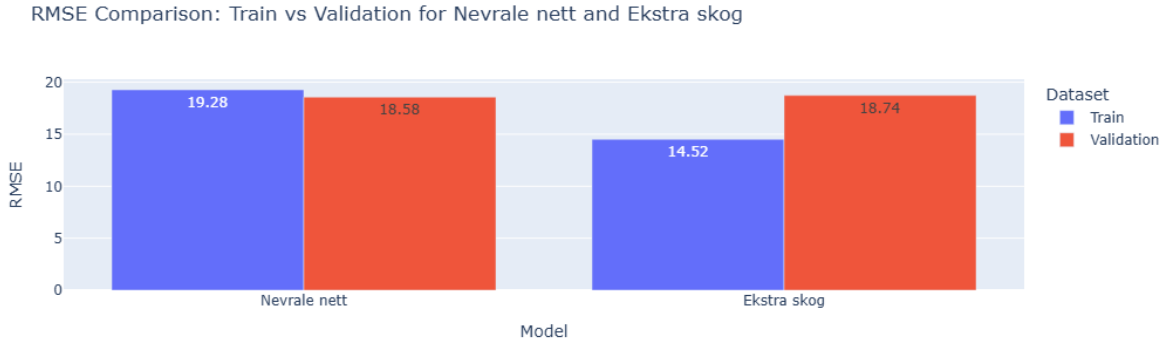
3. Endelig modell

3.0 Modellutvalg

For å finne den beste modellen til å predikere oppholds lengden til pasient, har jeg optimalisert flere modeller til min beste evne. Jeg startet med 10 modeller og har klart å redusere dette antallet ned til 4. Lasso og lineær regresjon har høyere RMSE og som jeg observerte tidligere predikerer modellene negative verdier, noe som ikke gir mening i sammenhengen av denne oppgaven. Dermed utelukker jeg disse modellene fra videre vurdering.

Modeller	RMSE
Lineær Regresjon	19.10
Nevrale nett	18.58
Lasso regresjon	19.09
Ekstra skog regresjon	18.74

RMSE indikere også at Nevrale nettverk modellen prestere bedre enn Ekstra skog regresjon. Jeg velger å undersøke om hvorvidt modellene under eller overfitter før jeg tar en endelig vurdering på hvilken modell jeg skal velge. Visualiseringen illustrerer at Nevrale nett modellen har en relativ liten differanse mellom RMSE på treningsdataen (19.28) og valideringsdataen (18.58), noe som tyder på svak underfitting. Dette kan indikere at modellen har ufullstendig optimalisering og har litt utfordring med å fange alle mønstrene i dataen. Sammenlignet med Ekstra skog, der RMSE er betydelig lavere på treningsdataen (14.52) enn på valideringsdataen (18.74). Det er klart at Ekstra skog modellen overfitter, det vil si at den tilpasser seg godt treningsdataen, men sliter med å generaliser ny data.



Videre undersøker jeg hvordan reduksjon i antall variabler påvirker RMSE. Tabellen viser at Nevrale nett modellen presterer bedre enn Ekstra skog, selv med redusert antall variabler. Å redusere antallet variabler fra opprinnelige 55 kan bidra til å forbedre tolkbarheten og eliminere mindre relevante variabler, som kan bidra til å forstå og analysere resultatene.

Antall variabler	RMSE Nevrale nett	RMSE Ekstra skog
10.0	18.76	19.06
11.0	18.76	19.02
12.0	18.76	18.95
13.0	18.78	18.94
14.0	18.76	18.94
15.0	18.75	18.95
16.0	18.67	18.94
17.0	18.66	18.95
18.0	18.65	18.87
19.0	18.65	18.90
20.0	18.63	18.85
21.0	18.62	18.81
22.0	18.63	18.79
23.0	18.64	18.84
24.0	18.63	18.81

På bakgrunn av disse funnene velger jeg Nevrale nett modellen til å være min endelige modell. Men før jeg avslutter denne modell analysen, ønsker jeg å undersøke effekten av ulike imputasjons strategier samt muligheten for ytterligere variabel utvinning.

3.2 Effekt av imputasjons strategier

Å velge en god imputasjonsstrategi kan ha en betydelig innvirkning på modellens ytelse, særlig når modellen trenes på datasett med flere manglende verdier. Siden treningsdataene opprinnelig har flere manglende verdier, ønsker jeg å vurdere effekten av ulike imputasjons strategier på Nevrale nettverk modellen. Merk at jeg fremdeles imputere noen variabler med anbefalte verdier som var vedlagt i data dokumentasjonen og at jeg imputere kategoriske kolonner med most frequent strategien.

Innledningsvis imputere jeg de forskjellige manglende verdiene med imputasjons strategier basert på hvilken variabel som skulle imputeres og valgte en strategi som ga best tilnærming til den manglende verdier. Selv om en slik strategi gir «riktig» imputasjons verdier, er det ikke gitt at den bidrar positivt til modellens predikasjons evne.

Imputasjons strategi	RMSE
Original	18.58
mean_imputer	18.58
knn_imputer 5	18.57
knn_imputer 10	18.55
knn_imputer 50	18.55
knn_imputer 150	18.55
knn_imputer 100	**18.54**
median_imputer	18.64
most_frequent_imputer	18.67

Jeg undersøker derfor flere ulike imputasjons strategier og skalere verdiene med Min Max -skalering, og ser at KNN – imputer med 100 naboer gir den laveste RMSE. Jeg velger derfor å bruke KNN imputer i kombinasjon med den endelig Nevrale nett modellen.

3.3 Effekt av variabelutvinning

Jeg utprøver ulike variable utvinninger for å undersøke om dette forbedrer prediksjonsevnen til Nevrale nett modellen. Denne prosessen inkluderte å fjerne kolonner for sykdomskategori_id og andre overflødige koloner, samt å slå sammen fysiologisk score og overlevelsesestimatene.

Variabel utvinning	RMSE
Original	18.54
Merge physiological scores	18.56
Merge survival estimate scores	18.57
Drop redundant cols	18.54
Merge both scores	18.59
Merge both scores and drop redundant cols	18.59
Drop disease category id	18.54
Drop redundant and disease category id	18.52

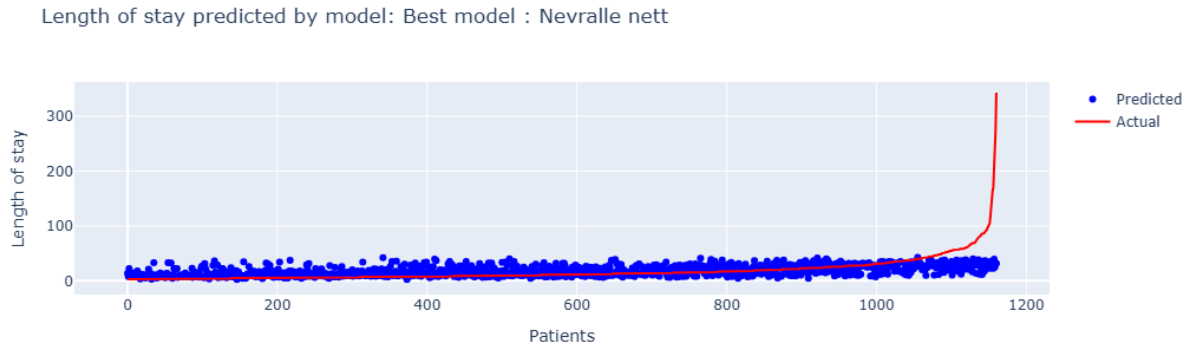
Tidligere analyser viste at de viktigste variablene for modellen inkluderer ulike overlevelsesestimater (både lege-estimater og generelle estimater) samt fysiologisk score. Resultatene viser å slå sammen disse kolonnene har imidlertid ingen signifikant forbedring i RMSE. Flere overflødige kolonner, som kjønn_male (motsatt av kjønn_female) og etnisitet_other, har begrenset tilleggsverdi. Ettersom å fjern sykdomskategori id kolonnene og overfladiske kolonner minker RMSE velger jeg å fjerne disse fra endelig modell analyse.

Antall kolonner	RMSE
13.0	18.75
14.0	18.73
15.0	18.74
16.0	18.73
17.0	18.74
18.0	18.74
19.0	18.71
20.0	18.66
21.0	18.64
22.0	18.66
23.0	18.66
24.0	18.60
25.0	18.63
26.0	18.61
…	..
Alle	18.52

Jeg utfører en ny feature selection og forsøker å trene modellen på nytt med forskjellig antall variabler nå som noen kolonner er fjernet. Likt som tidligere observerer jeg at reduksjon i antall variabler fører til små økning i RMSE (For hele tabell se jupyter notebook). Jeg observerer at desto flere variabler som blir tatt med i modellen, desto bedre presterer den. Dette gir også intuitiv mening, da modellen har mer data å predikere med. Nevrale nett modellen presterer best når alle variablene er tatt med.

3.4 Test av endelig modell

Jeg tester den endelige modellen, modellen oppnår en RMSE på 21.63 dager. Dette er økning fra RMSE på valideringsdataen, som var 18.52 dager. Sammenlignet med baseline som hadde en RMSE på 20.6 dager, presterer den endelige modellen fremdeles under det ønskede nivået. Visualiseringen av resultatene viser at modellen predikerer verdier under 50 dager med relativ lavt avvik, men har utfordringer med å forutsi ekstreme verdier, noe som fører til større totalavvik. Dette er som forventet da modellen også under modelleringen ikke predikerte ekstremalverdier.



For å forbedre brukervennligheten har jeg utviklet to modeller: en simplifisert modell med de 20 viktigste variablene for nettsiden og en fullverdig modell med alle variablene som jeg bruker for å predikere sample data. Den forenklede modellen har em RMSE på 18.66 dager på validering dataen, kun litt høyere enn RMSE på endelige modellen (18.52 dager). Selv om det teoretisk kunne være en liten forbedring i RMSE ved å inkludere flere variabler, gir den simplifiserte modellen bedre brukervennlighet på nettsiden uten en betydelig reduksjon i nøyaktighet.

Basert på min generelle kunnskap innen data science og forståelse av datasettet, forventet jeg et visst avvik på grunn av kompleksitet og variasjonen i pasient data fra data tilberedningen. I første del av data tilbedelsen observerte jeg stor spredning og skjev fordeling i oppholdslengden. Den endelige modellen fikk utfordringer med nettopp dette, å forutsi ekstreme verdier, selv om den prester tilfredsstillende for kortet opphold, som samsvarer med mine forventninger.

Modellen har potensiale for praktisk bruk, spesielt i prediksjonen av kortere oppholdslengde. Begrensningene med ekstreme verdier, tilsier at modellen passer best som en estimerisk verktøy heller enn en nøyaktig prediksjonsmodell.

3.5 Forbedringer og styrker, drøfting av resultat

Styrker ved modellen:

- Imputasjon: Ved å imputere kolonnespesifikk imputasjon sikret jeg gode imputasjons verdier, selv om dette ikke er bruk i endelig modell fikk jeg muligheten til å sammenligne ulike imputasjons strategier.
- Håndtering av outliers: Jeg har gjort utforskende data analyse og fjerning av outliers ved hjelp av persentil klipping på både x og y-verdier. Jeg undersøkte også flere ensemble-metoder for å forbedre modellenes generaliseringsevne og prediksjonspresisjon. Dette ble igjen ikke bruk i endelige modellen, men viste stor effekt på flere modeller.
- Dataaggregering og Skalering: Ved å utforske dataaggregering, skaleringsmetoder (spesielt for nevrale nettverk), kunne modellen bedre håndtere ekstreme verdier og forbedre generaliseringsevnen.
- Modellforenkling: Redusere antallet variabler i den simplifisert modell gjør modellen eklere å tolke, uten at det gikk mye utover presisjonen.

I tillegg til forbedringen som er kommentert underveis i rapporten, er dette noen av de mulige forbedringene jeg hadde utforsket hadde jeg hatt mer tid:

- Utforske flere modeller og bruk av RFECV for å systematisk teste kombinasjoner av variabler kunne avdekket bedre modeller med andre variabel kombinasjoner.
- Samle mer data for å øke modellens robusthet og forbedre modellen generaliseringsevne.
- En utvidet hyperparametertuning kunne ytterligere forbedret modellens ytelse.
- Utføre ytterligere variabelutvinning med innsikt fra fagperson med spesifikk domeneekspertise kunne styrke datagrunnlaget.
- Innføre metoder for å håndtere negative prediksjoner.
- Benytt aggregerte data for å redusere støy og få et tydeligere datagrunnlag.
- Undersøk dimensjonsreduksjonsteknikker som PCA for å redusere antall funksjoner og støy i datasettet.

Bibliografi

Berman, J., & Dugdale, D. C. (2023, august 20). *Creatinine blood test*. Retrieved november 2, 2024, from Mount Sinai: https://www.mountsinai.org/health-library/tests/creatinine-blood-test

Blaser, N. (2024). *Forlesningsnotater fra innføring i data science INF161*. Retrieved from github: https://blasern.github.io/data-science-forelesningsnotater/intro.html

Harrel, F. (2022). *UC Irvine Machine Learning Repository*. Retrieved from Support2: https://archive.ics.uci.edu/dataset/880/support2

Nickson, C. (2024, Juli 6). *Life in the fastlane*. Retrieved from PaO2/FiO2 Ratio (P/F Ratio): https://litfl.com/pao2-fio2-ratio/

Wikipedia contributors. (2024, oktober 19). *Human body temperature*. Retrieved november 2, 2024, from Wikipedia: https://en.wikipedia.org/wiki/Human_body_temperature

Wikipedia contributors. (2024, mars 24). *Human serum albumin*. Retrieved november 2, 2024, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Human_serum_albumin&oldid=1215286494

Wikipedia contributors. (2024, september 10). *Sodium in biology*, 1245041649. Retrieved november 2, 2024, from Wikipedia: https://en.wikipedia.org/w/index.php?title=Sodium_in_biology&oldid=1245041649

Wikipedia contributors. (2024, september 7). *White blood cell*. Retrieved november 2, 2024, from Wikipedia: https://en.wikipedia.org/w/index.php?title=White_blood_cell&oldid=1244459796