Predikere Sykehusopphold

INF161

Andreas Sløgedal

03.11.2024

Contents

1 Introduksjon

Hensikten med denne oppgaven er å anvende alt materialet lært i INF161 til å utvikle en maskinlæringsmodell som kan forutsi sykehusopphold for nye pasienter. Modellen vil benytte pasientopplysninger fra fire ulike datasett og trenes ved hjelp av avanserte maskinlæringsteknikker. Ved å oppgi helserelaterte og sosioøkonomiske data, ønsker vi å få en helhetlig forståelse av pasientens helse. Målet er at modellen skal kunne forbedre prediksjonsnøyaktigheten og dermed effektivisere pasientbehandling på sykehuset.

1.1 Beskrivelse av datasettene

`Demographic.csv` inkluderer variabler som kjønn, alder, utdanning, inntekt og etnisitet. Disse dataene er ikke nødvendigvis direkte tilknyttet pasientens helsetilstand, men det er aktuelt å undersøke om de har innvirkning på sykdomskategorier og behandlingsresultater. Derfor vil det være interessant å undersøke om det er sammenhenger mellom disse demografiske variablene og helseutfallene i dataanalysen.

`Hopsital.csv` inneholder informasjon om pasienter som har fått behandling på sykehuset. Variablene inkluderer oppholdslengde, eventuelle dødsfall og innleggelse. Her fungerer oppholdslengden som den avhengige variabelen, og er dermed målvariabelen vi ønsker å predikere. Målvariabelen er sentral i treningsdataene, ettersom maskinlæringsmodellen har som hensikt å undersøke hvor lenge pasienten vil oppholde seg på sykehuset.

Både `severity.json` og `physiological.txt` inneholder uavhengige variabler som alle er nødvendige for å undersøke pasientens tilstand. `Physiological.txt` inkluderer numeriske mål som hjertefrekvens, respirasjonsfrekvens, kroppstemperatur og nivåer av kreatinin. Data over sykdomsalvorlighet, på den andre siden, tar for seg mer kategorisk informasjon, som blant annet pasientens diagnose, sykdomsunderkategori, og overlevelsesestimater. Dette er alle sentrale mål for å forstå pasientens helse og prognose.

I praktisk sammenheng ønsker vi å samle inn data og utvikle en modell for å diagnostisere pasienter på en mest mulig effektiv måte. Målet er å optimalisere behandlingsprosessen, noe som maksimerer sannsynligheten for overlevelse.

2 Innledende dataforberedelse

2.1 Klargjøring av data

All datamanipulasjonen i den innledende dataforberedelsen er utført med hensyn til at testdataene ikke skal undersøkes før modellen er ferdig trent. For å kunne gjennomføre dataanalyse på en effektiv og oversiktlig måte importerer jeg Python-biblioteket `Pandas`, som er et kraftig verktøy ved dataanalyse-, og manipulering. Ved hjelp av Pandas laster jeg inn datasettene `demographic.csv`, `hospital.csv`, `physiological.txt` og `severity.json`. Disse lagres i variablene henholdsvis `demographic_df`, `hospital_df`, `phychological_df` og `severity_df`.

2.1.1 Demografiske data

For å kunne slå sammen datasettene er det nødvendig de har samme antall rader. Ulikt antall rader kan føre til uventede resultater i analysen, som duplikater eller manglende data. Vi kan kontrollere antall kolonner og rader i datasettene ved å bruke Pandas attributtet `shape`. `demographic_df` har 7742 rader, mens de øvrige datasettene har 7740 rader. Denne ulikheten skyldes duplikater i datasettet for demografiske data, som må fjernes før sammenslåing. Vi bruker `drop_duplicates()` på dette datasettet.

2.1.2 Fysiologiske data

`physiological_df` inneholder en betydelig andel NaN-verdier, noe som kan skape utfordringer i senere faser, spesielt under modellutvikling og prediksjon av oppholdslengde. For å unngå tap av verdifull informasjon ved fjerning av datapunkter, erstattes NaN-verdiene med anbefalte eksempelverdier (SUPPORT2, 2024).

2.1.3 Sykehusdata

`hospital_df` inneholder to uavhengige variabler: `sykehusdød` og `oppholdslengde`. Ved å undersøke de statistiske målene i datasettet, ser vi en verdi

som skiller seg ut: den minste observerte oppholdslengden er på `-99 dager`, som er urealistisk. Det er essensielt å undersøke om det finnes flere negative oppholdslengder, slik at de kan behandles før trening av maskinlæringsmodeller begynner.

2.1.4 Sykehusalvorlighetsdata

`severity_df` kommer i et annet format enn de andre datasettene, der alle kolonnene utenom de to første er lister. Uten et tilstrekkelig strukturert dataoppsett ville det ikke vært mulig å slå sammen datasettene. Derfor brukes `explode()`, slik at hver verdi i listen blir plasseres på egen rad, mens de andre kolonneverdiene i samme rad forblir uendret. Dette innebærer at det opprettes en ny rad for hver listeverdi i kolonnen, noe som effektivt «utvider» datasettet. Ved sammenslåingen av datasettene viser dette steget seg avgjørende for videre dataanalyse, da alle har likt antall pasienter.

2.2 Kombinering av datasett

Nå som hvert kolonne-rad-par i sykehusalvorlighetsdata inneholder én enkelt verdi, og alle datasettene har likt antall rader, gjenstår kun ett steg for å kunne kombinere dem. å fjerner den eksisterende indeksen i hvert datasett og definerer en ny standardindeks. Siden hvert datasett inneholder kolonnen `pasient_id` velger jeg å droppe denne kolonnen i de tre siste datasettene for å unngå duplisering. Deretter kombinerer jeg de fire datasettene ved å bruke metoden `concat()`. Med `axis=1` skjer sammenslåingen horisontalt, slik at kolonnene fra de fire datasettene legges ved siden av hverandre. Det kombinerte datasettet kaller jeg `df`.

2.3 Databehandling

Etter datasettet er slått sammen, ønsker vi å undersøke om dataene er klare for å brukes videre i maskinlæringsmodellene. Ved å bruke metoden `describe()` identifiserer jeg verdier som virker unaturlige, som negative verdier av `oppholdslengde` og `alder`. De negative verdiene må håndteres for å optimalisere prediksjonsevnen til modellen. Jeg velger å kun fjerne pasienter med negativ alder før inndeling av trenings-, validerings- og testdata.

Å fjerne negative verdier kan bidra til mer forklarende visualiseringer og robuste modeller. Videre velger jeg å fjerne kolonnene `dødsfall` og `sykehusdød`, da de ikke har noe reell innvirkning på modellens prediksjon av oppholdslengde på innsjekkingsdagen. Kolonnen `pasient_id` fjernes, og datasettet indekseres på nytt. `adl_pasient`, som representerer pasientens funksjonsevne, og `bilirubin`, som

måler bilirubinnivå, ble begge utfylt av pasienten ved dag syv. Disse verdiene har ikke innvirkning på prediksjon av oppholdslengde ved dag innsjekk, og fjernes av den grunn. Avslutningsvis fjerner jeg `sykdomskategori_id`, fordi denne informasjonen overlapper med `sykdomskategori`. Generelt er det essensielt å sikre at kun data som er tilgjengelig ved pasientens ankomst blir brukt i modellene, for å unngå forvirring med informasjon fra fremtidige tidspunkter.

2.4 Inndeling av trenings-, validerings- og testdata

For å bygge en maskinlæringsmodell som presterer godt på ukjente data, er det nødvendig å dele datasettet `df` inn i trenings-, validerings- og testsett. Først opprettes `X` som består av alle kolonnene utenom den avhengige variabelen `oppholdslengde`. Den avhengige målvariabelen `oppholdslengde` skal modellen predikere, og lagres i variabelen `y`. Treningssettet brukes til å trene modellen, og består av 70% av datasettet. Den resterende 30% er testdata. Dermed splittes testsettet (30% av det originale testsettet) i to like store deler. 50% av disse brukes som valideringsdata, og de resterende 50% brukes som testsett. Avslutningsvis sorterer vi treningsdataene etter `pasient_id`.

3 Dummy-kodifisering

I treningsdataen har vi en rekke kategoriske variabler, som representerer verdier som ikke kan måles på en numerisk skala. I maskinlæringsmodeller ønsker vi å bruke numeriske verdier, da de bidrar til å forbedre modellens ytelse og nøyaktighet. Kategoriske data kan være utfordrende for modellen å tolke fordi de ikke oppgir direkte numeriske verdier. For å forbedre maskinlæringsmodellene, benytter jeg feature engineering for å transformere og lage nye funksjoner fra kategoriske dataene.

Ved å bruke `OneHotEncoder` fra biblioteket `sklearn.preprocessing`, transformerer jeg de kategoriske kolonnene til binære formater, noe som gjøre det enklere for modellen å tolke dataene. De spesifikke kolonnene som transformeres, inkludert `kjønn`, `inntekt`, `etnisitet` og `sykdomskategori`, er lagret i `cols_to_encode`. De transformerte kolonnene lagres i `X_train_transformed`, og det opprettes en ny DataFrame. Det samme gjøres for både testdataene og valideringsdataene. Avslutningsvis runder jeg av verdiene til heltall, og slår sammen det originale datasettet med det nye datasettet.

4 Utforskende statistisk dataanalyse

Statistisk analyse er en viktig metode for å forstå dataene bedre og trekke ut nyttig informasjon. Den gjør det mulig å identifisere mønstre, oppdage problemer tidlig, og forutsi fremtidige hendelser basert på tidligere data. I tillegg kan statistisk analyse avdekke skjulte sammenhenger mellom ulike variabler. Dette er avgjørende for å sikre at treningsdataene er klare og pålitelige før de brukes i videre analyser og til trening av modeller.

4.1 Statistiske mål: Gjennomsnitt, standardavvik, kvantiler og mer

Vi undersøker statistiske målene i treningsdata ved å bruke metoden `describe()` på treningsdataene**.** Dette gir oss en oversikt over diverse statistiske verdier, som gjennomsnitt, minimum, maksimum og standardavvik for variablene i datasettet. Denne informasjonen er avgjørende for å forstå pasientenes helsetilstand.

Oppsummeringen av statistiske mål for kolonnene i DataFrame `X_train` viser oss blant annet at:

Gjennomsnittsalderen blant pasienter på sykehuset er 62,7 år.
Gjennomsnittlig blodtrykk er 84 slag i minuttet.
Laveste kroppstemperatur er 31,7 grader celsius.
Høyeste hjertefrekvens blant pasientene er 232 slag i minuttet.

Disse verdiene er nyttige for en maskinlæringsmodellene da de gir innblikk i underliggende mønstre i dataene. Statistisk analyse bidrar også til å identifisere ekstremverdier og andre potensielt kritiske funn. Slike funn krever videre undersøkelse for å vurdere om de er støy i dataene, eller om det er underliggende medisinske årsaker bak dem. Ved å forstå statistiske mål kan vi også tilpasse våre behandlingsteknikker for at modellen yter godt, og gir pålitelige prediksjoner. Derfor er forståelsen og oversikten over statistiske mål en viktig del av dataforberedelsesprosessen.

5 Utforskende dataanalyse – visualisering

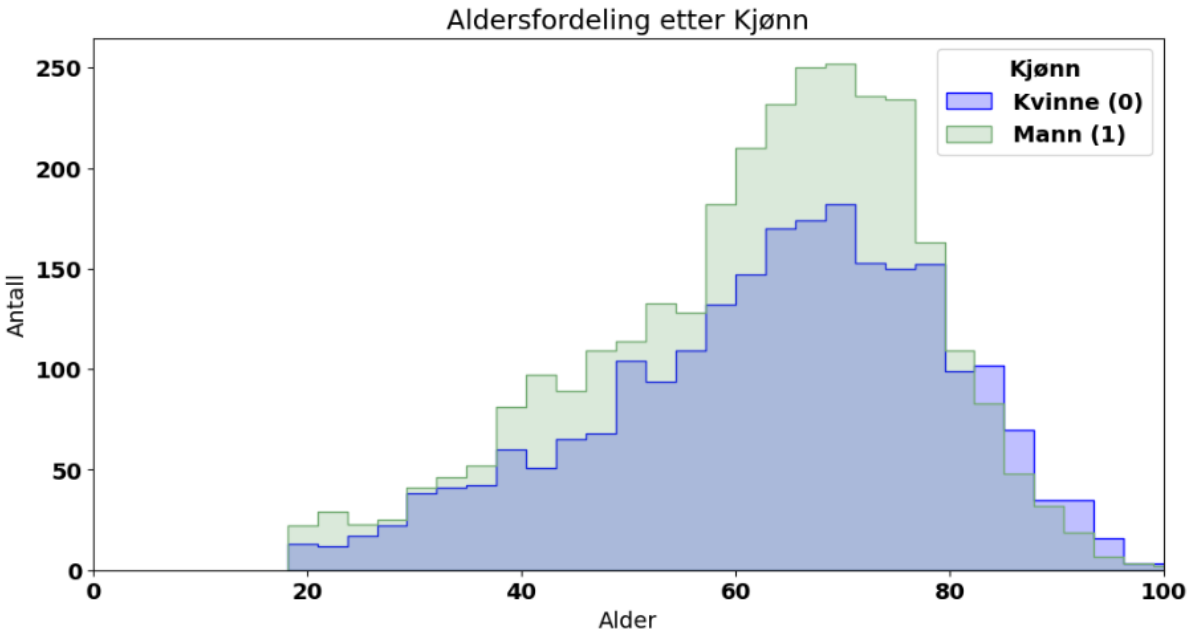
Visualisering av dataen er en effektiv og oversiktlig måte å støtte og validere funnen fra den statistiske analysen og databehandlingen. Ved å kombinere visualisering og numerisk analyse er det enklere å forstå trender og mønstre i dataene. Visualisering er også en effektiv måte å identifisere avvik og uventende resultater.

5.1 Demografiske variabler

Variabler som kjønn, alder, utdanning og inntekt er viktige demografiske faktorer som potensielt kan ha innvirkning på helseutfall. Disse variablene viser hvordan ulike samfunnsgrupper påvirkes ulikt av helseproblemer, og kan være med på å identifisere sårbare grupper som trenger spesifikke helsetjenester.

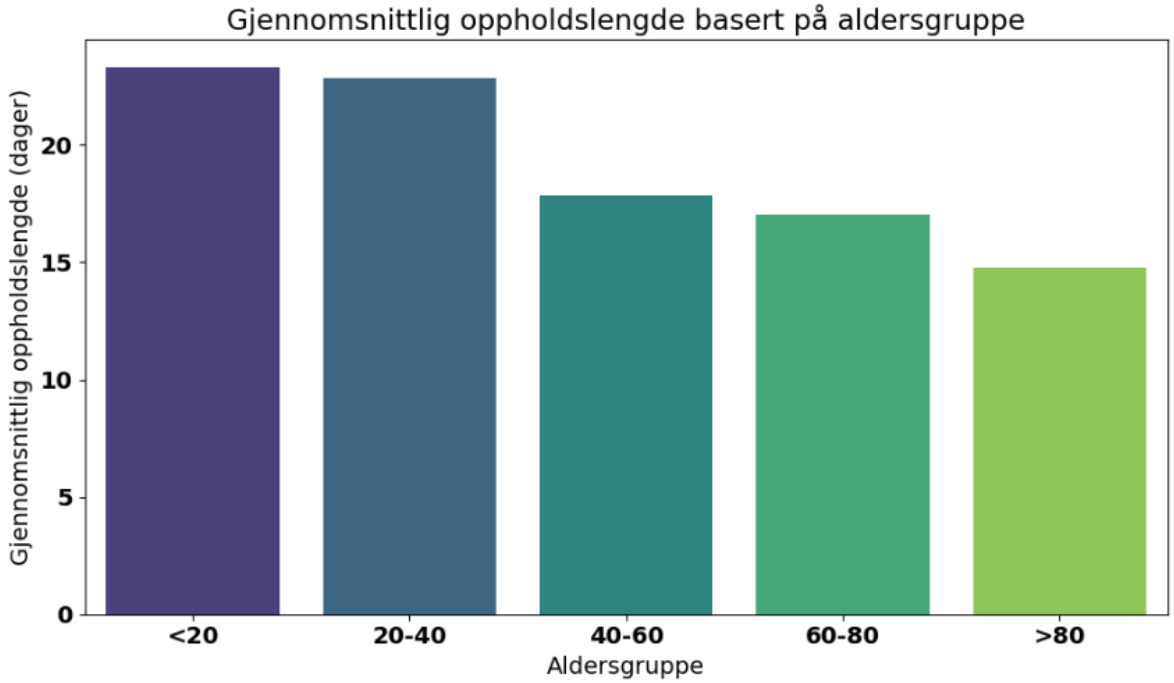
5.1.1 Alder

Alder har ofte sammenheng med sykdomsalvorlighet og oppholdslengde, da kroppen og immunforsvaret blir svakere med tid. Å få oversikt over aldersfordelingen blant pasientene kan være aktuelt fordi ulike aldersgrupper kan ha varierende behandlingsbehov. Figur 5.1.1 viser at det er en stor andel pasienter i alderen 60-80 år, av både menn og kvinner i treningsdataene.



Figur 5.1.1: Histogram som illustrerer aldersfordelingen basert på kjønn.

Videre kan fordelingen over oppholdslengde for ulike aldersgrupper være aktuelt å undersøke, slik at sykehuset kan effektivisere allokering av ressurser og effektivisere behandling basert på ulike alderskategorier. Fra figur 5.1.1 vet vi at en stor del av befolkningen er eldre pasienter. Dersom det viser seg at oppholdslengden i denne gruppen også er kort, grunnet for eksempel sykehusdødsfall, er det et tegn på at mer ressurser bør allokeres til denne gruppen. Vi undersøker om dette er tilfellet i figur 5.1.2.

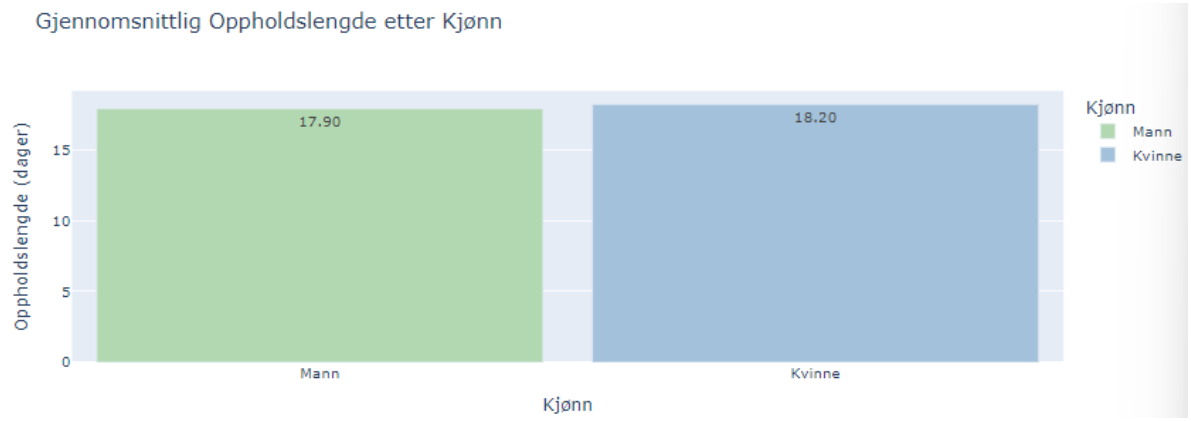


Figur 5.1.2: Stolpediagram som illustrerer gjennomsnittlig oppholdslengde basert på aldersgrupper.

Som vi ser, er oppholdslengden på sykehuset kortest blant de eldste pasientene i treningssettet. Dette forventet, da dødeligheten blant de eldste ofte er høyest som følge av svekket immunforsvar. Histogrammet gir sykehuset verdifull informasjon, som gjør at det er mulig å tilpasse ressursallokeringen opp mot aldersgrupper. Det er imidlertid verdt å merke seg at forskjellene ikke er dramatiske. Dette tyder på at selv om alder er en faktor som påvirker oppholdslengden, finnes det andre variabler som også spiller en betydelig. For å få en bedre forståelse av disse faktorene, undersøker jeg videre hvordan andre demografiske og fysiologiske variabler påvirker oppholdslengden.

5.1.2 Kjønn

Det er relevant å undersøke om kjønn påvirker helseutfall og oppholdslengde.



Figur 5.1.3: Oversikt over gjennomsnittlig oppholdslengde blant kjønnene.

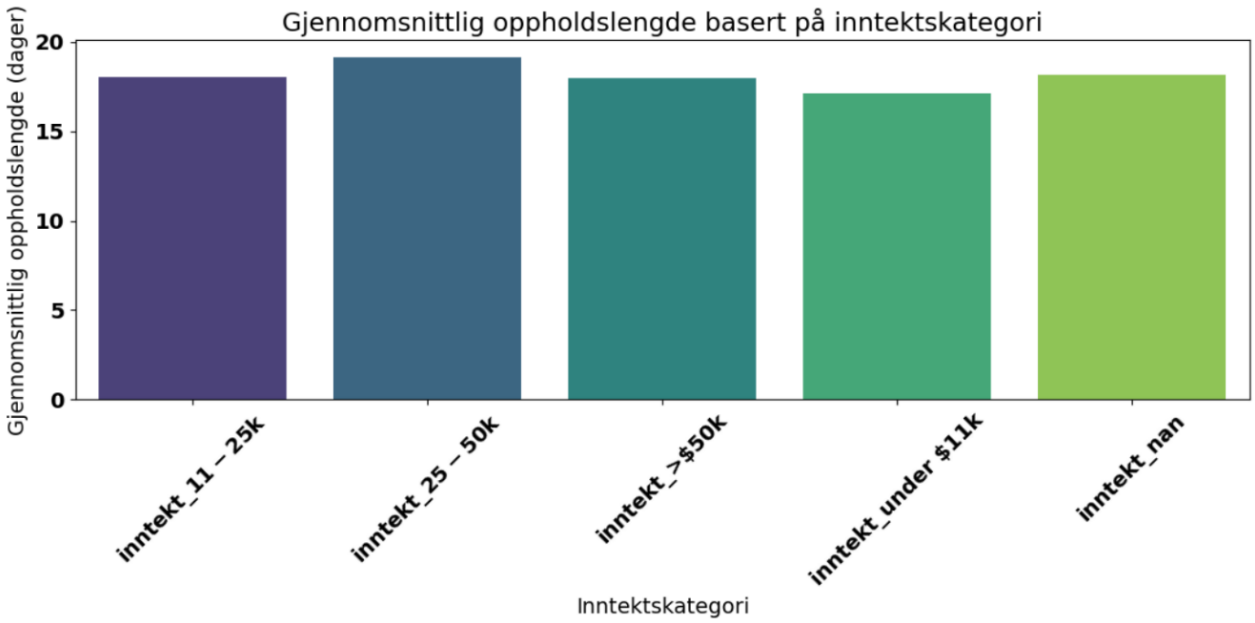
Basert på stolpediagrammet er det tydelig at kjønn ikke har en signifikant innvirkning på oppholdslengden. Derfor legger jeg mindre vekt på denne faktoren i videre analyse, og undersøker heller om andre demografiske faktorer har innvirkning på helseutfall.

5.1.3 Inntekt

Sosioøkonomiske variabler, som inntekt, kan ha innvirkning på sykdomsforekomster og alvorlighetsgrad. Min hypotese er at pasienter med høyere inntekt har bedre tilgang til medisinsk behandling, noe som kan resultere i mindre alvorlige sykdomsforløp og kortere opphold på sykehuset. For å undersøke denne hypotesen har jeg analysert gjennomsnittlig oppholdslengde basert på inntekt i et stolpediagram.

Figur 5.1.4 viser at det er marginale forskjeller i oppholdslengde på tvers av inntektskategoriene i treningssettet. Dette indikerer at inntekt ikke er en avgjørende

faktor for oppholdslengde. Basert på stolpediagrammet forkaster jeg min nullhypotese om at inntekt har en signifikant innvirkning på oppholdslengden.



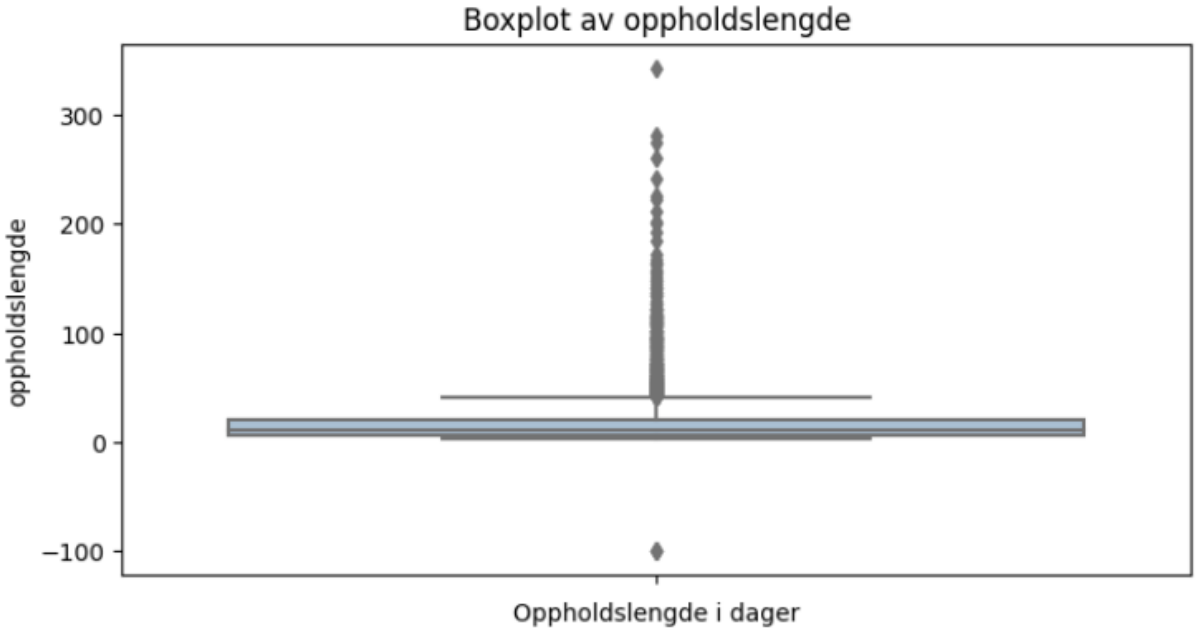
Figur 5.1.4: Stolpediagrammet illustrerer gjennomsnittlig oppholdslengde basert på inntektskategori.

5.2 Helserelaterte variabler

Helserelaterte variabler gir direkte informasjon om helsetilstanden til pasienten, og gir en direkte kobling mellom ulike variabler og helseutfall. Ved å analysere disse variablene kan man få identifisere faretegn tidlig og få innsikt i hvordan ulike variabler påvirker sykdomsalvorlighet, og behandlingsbehov. Slik informasjon kan være aktuell for å tilrettelegge behandling til pasienter, og optimalisere predikert oppholdslengde på sykehuset.

5.2.1 Oppholdslengde

Oppholdslengden er den avhengige variabelen vi ønsker å predikere i modellen, og det er viktig å få en oversikt over denne variabelen for å forstå fordelingen og eventuelle uteliggere i treningsdataene. Den gjennomsnittlig oppholdslengden i treningssettet er på omtrent `18 dager`, og det er relativt lav varians i dataene. Likevel er det flere interessante uteliggere, blant annet uteliggere med svært høye positive verdier. Dette tyder på at det er enkelte pasienter hatt lange sykehusopphold, uten å nødvendigvis å ha være alvorlig syke.



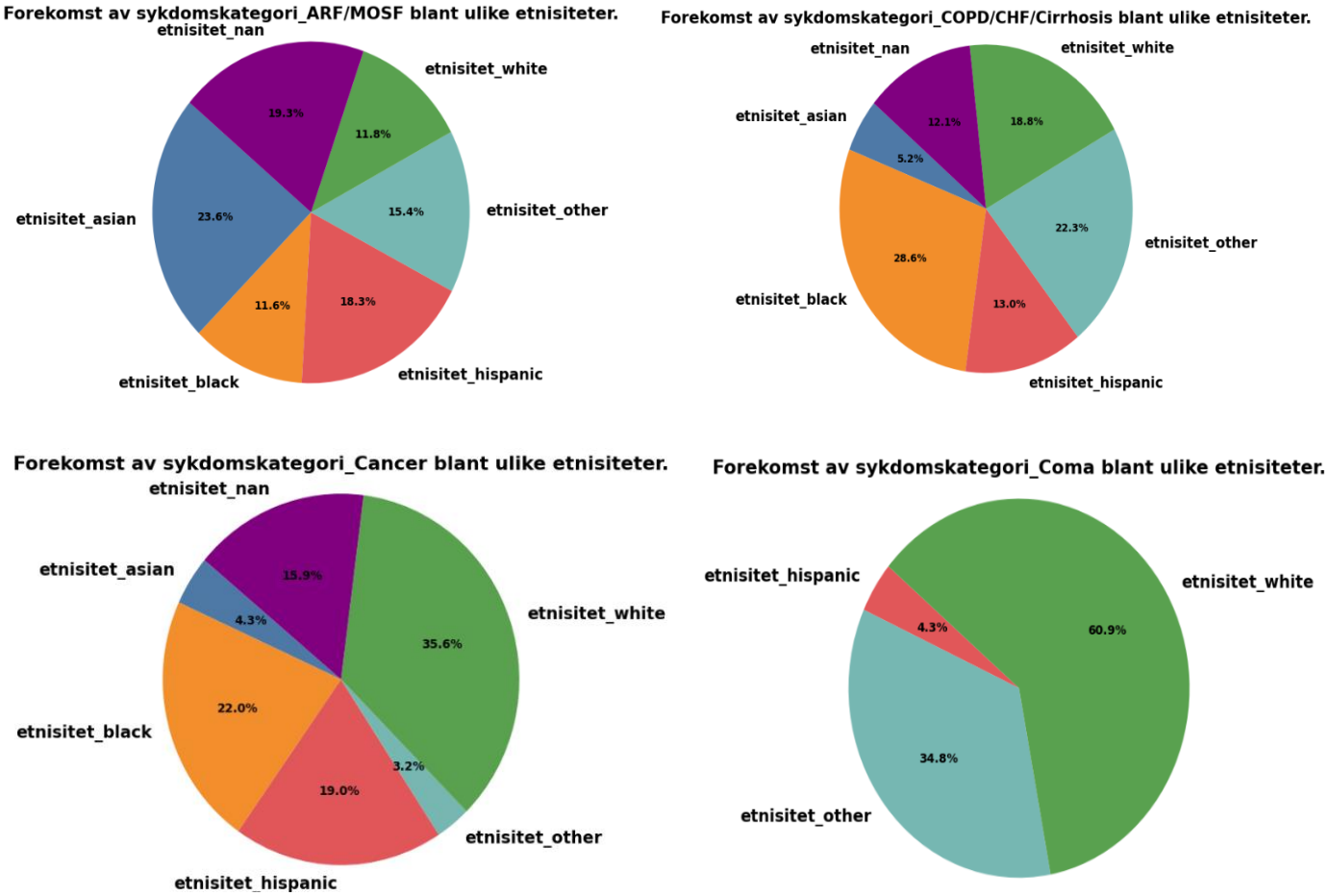
Figur 5.2.1: Boksplottet viser fordeling av oppholdslengde blant pasienter på sykehuset.

Figur 5.2.1 viser en interessant observasjon: minst én pasient har en negativ oppholdslengde. Denne verdien er ikke realistisk, så jeg bytter derfor ut alle negative verdier for medianverdien for oppholdslengde i treningssettet.

5.2.2 Sykdomskategorier blant etnisiteter

Å undersøke sykdomskategorier blant pasienter med ulike etnisiteter er viktig for å forstå om visse etnisiteter er mer utsatt for enkelte sykdommer. Om bemerkelsesverdige verdier er funnet, kan det hjelpe sykehuset med å tilpasse behandling opp mot ulike befolkningsgrupper, og effektivisere pasienters oppholds på sykehuset.

I figur 5.2.2 undersøkes sykdomsforekomster blant de ulike etnisitetene for å avdekke underliggende mønstre i treningsdataene.

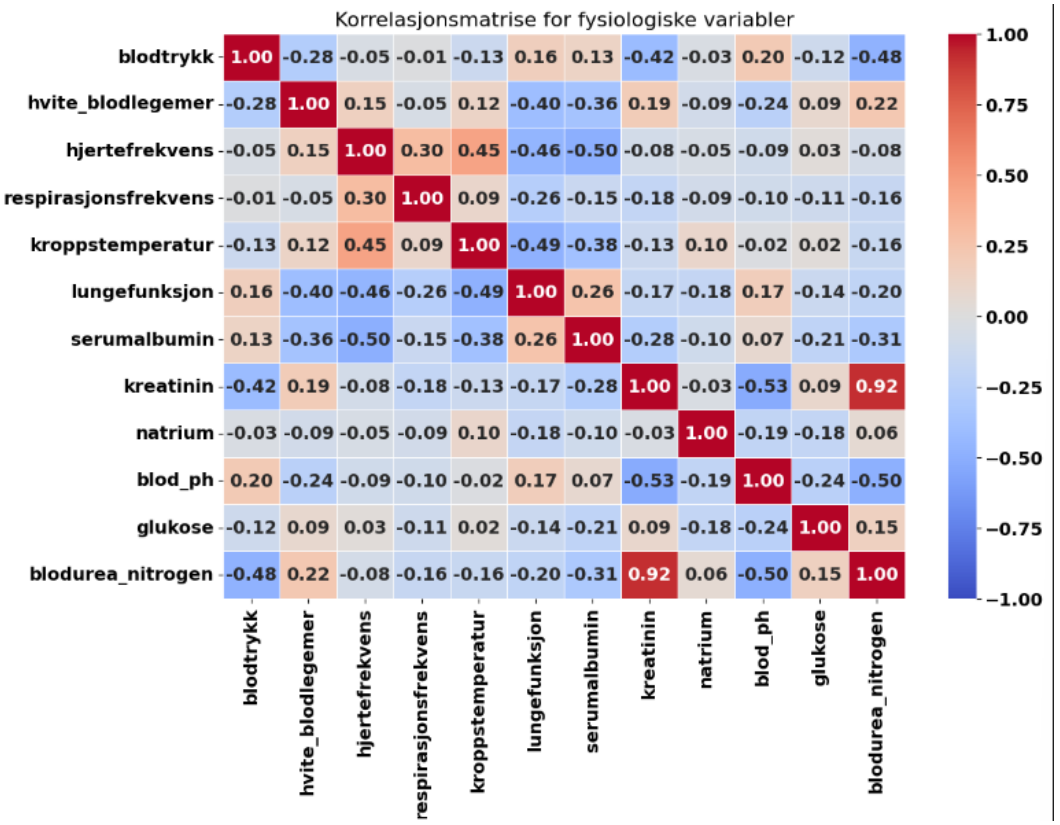


Figur 5.2.2: Sektordiagrammene gir viser fordelingen av de fire sykdomskategoriene blant etnisitetene.

Noen resultater er verdt å merke seg, blant annet den store andelen av `etnisitet_white` som faller under sykdomskategori kreft og koma. Dette foreslår mulige sammenhenger mellom etnisiteter og sykdomskategori i treningssettet. Resultatene indikerer at etnisitet kan ha en innflytelse på forekomst av sykdom, og disse variablene beholdes i treningssettet av den grunn. Funnene i figur 5.2.2 kan hjelpe sykehuset å predikere sykehusoppholdet til pasienter opp mot sykdomskategorier mer effektivt.

5.2.3 Korrelasjon mellom fysiologiske variabler

Oversikt over variabler tilknyttet fysiologiske variabler kan hjelpe med å diagnosere og behandle pasienter på sykehuset. Korrelasjonsmatriser kan hjelpe sykehuset med å forstå sammenhengene mellom fysiologiske variabler i treningsdataene. Dette kan føre til raskere og mer presise diagnoser, og dermed bedre prediksjoner av oppholdslengde. I tillegg er korrelasjonsmatriser nyttige verktøy i dataforberedelse for å eliminere variabler som korrelerer sterkt. Når vi bygger maskinlæringsmodeller, ønsker vi å eliminere høyt korrelerte variabler. Høy korrelasjon kan gjøre at modellen overtilpasser seg treningsdataene, og ikke generaliserer godt på testdataene. Ved å minimere høyt korrelerte variabler kan vi forbedre modellens evne til å predikere oppholdslengden på ukjent pasientdata.



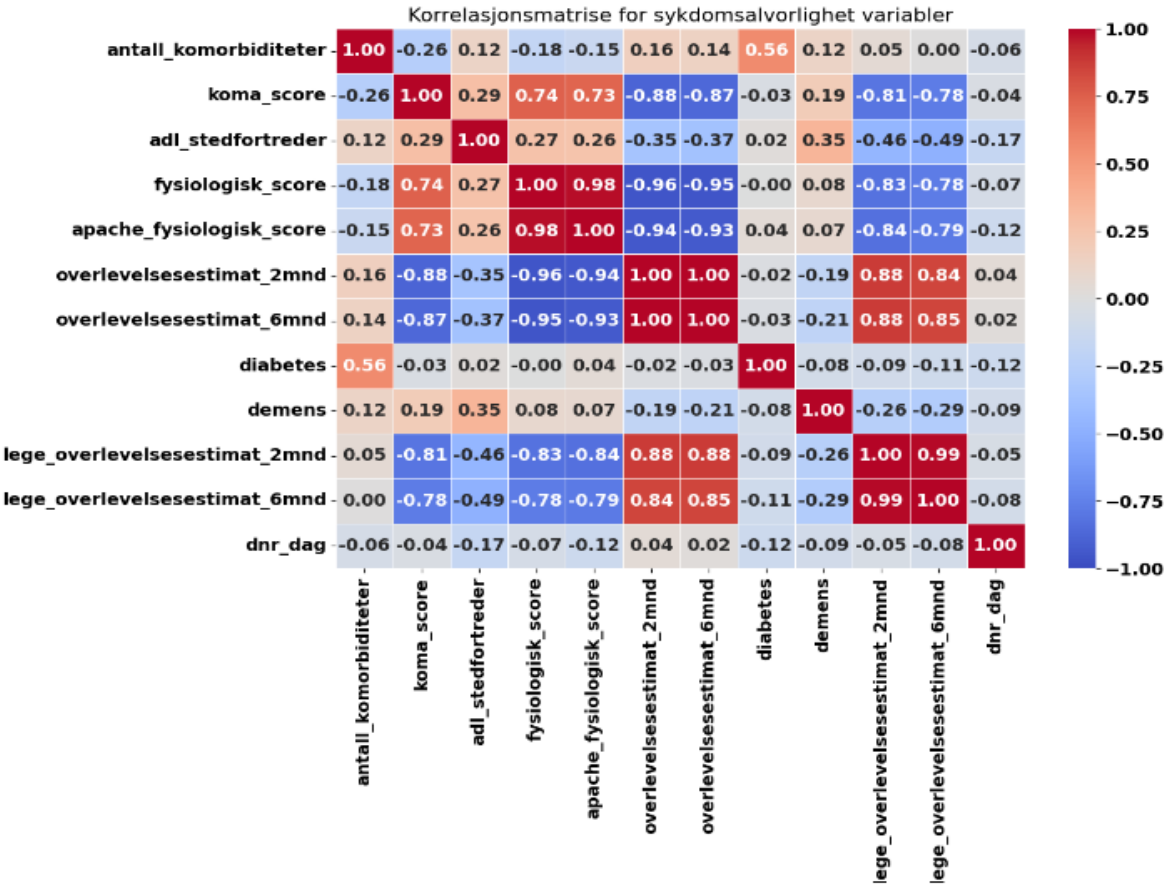
Figur 5.2.3: Korrelasjonsmatrisen gir oversikt over korrelasjon blant variablene i fysiologisk data.

Matrisen gir en god oversikt over variabler som korrelerer sterkere enn andre. Eksempelvis viser `kreatinin` og `blodurea_nitrogen` sterk positiv korrelasjon

(0.92). Selv om variablene korrelerer sterkt, velger jeg å beholde dem som en del av treningssettet, da de gir unik informasjon om pasienters helsetilstand.

5.2.4 Korrelasjon mellom variabler i sykdomsalvorlighet

For å unngå overtilpasning i modellen, er det viktig å få oversikt over korrelasjon mellom sykdomsalvorlighetsdataene i treningsdataene. I korrelasjonsmatrisen ser vi at flere variabler som har en tilnærmet perfekt negativ eller positiv korrelasjon. Vi identifiserer dem slik at vi kan tilpasse datasettene før vi bygger modellene.



Figur 5.2.3: Korrelasjonsmatrisen gir oversikt over korrelasjon blant variablene i sykdomsalvorlighetsdata.

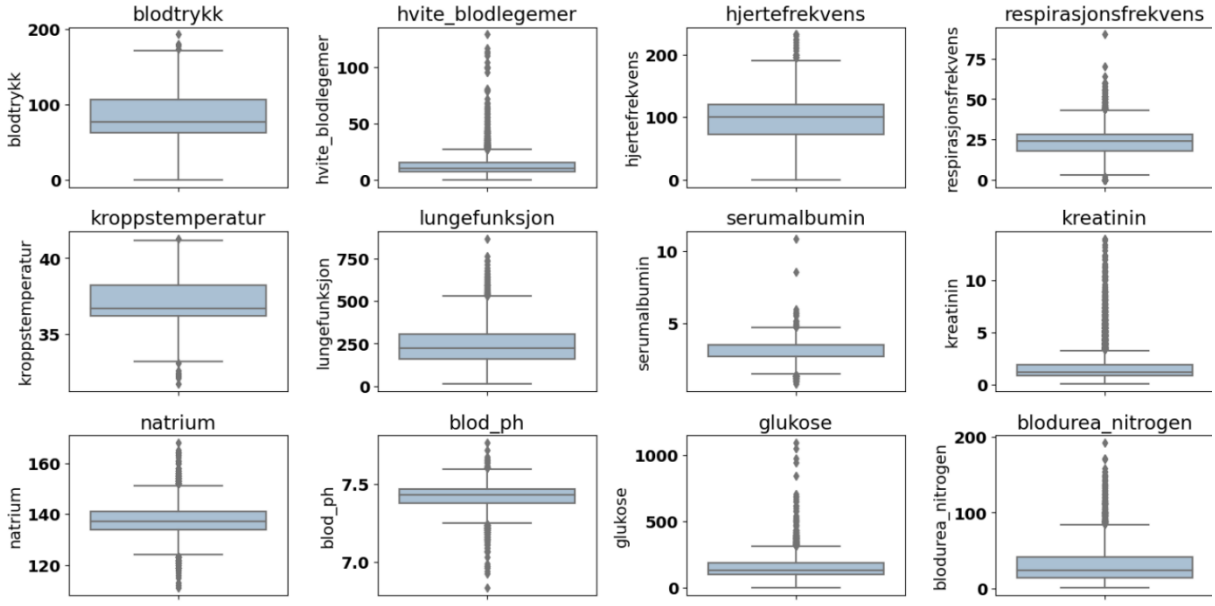
Spesielt merker vi oss at `lege_overlevelsesestimat_2mnd` og `lege_overlevelsesestimat_6mnd` har en meget sterk positiv korrelasjon med henholdsvis `overlevelsesestimat_2mnd` og `overlevelsesestimat_6mnd`. For å unngå overtilpasning av modellen kan det være hensiktsmessig å fjerne et av disse variabelparene. Tilsvarende observerer vi at `fysiologisk_score` og

`apache_fysiologisk_score` er tilnærmet perfekt negativ korrelert. Jeg velger å sette korrelasjonsgrensen på ±0.95, da dette innebærer en tilnærmet lik perfekt korrelasjon.

Ut ifra resultatene i korrelasjonsmatrisen, der korrelasjonskoeffisientene for `lege_overlevelsesestimat_2mnd` og `overlevelsesestimat_2mnd` er 0.98, samt for `lege_overlevelsesestimat_6mnd` og `overlevelsesestimat_6mnd`, velger jeg å fjerne legens estimat. Det samme gjelder `apache_fysiologisk_score`, som viser høy korrelasjon med `fysiologisk score` (korrelasjon = 0.98). Fjerning av disse variablene er essensielt fordi variablene har overlappende funksjoner, vi unngår overtilpasning på treningsdataene, og forbedrer modellens generaliseringsevne på testdataene.

5.2.5 Fysiologiske data og uteliggere

For å oppnå god ytelse for modellene er det essensielt å forstå fordelingen av de fysiologiske variablene, for å så kunne å utelukke uteliggere i dataene. Hvis treningssettet inneholder mye uteliggere, kan det føre til dårlig generaliseringsevne, og økt risiko for overtilpasning. Figur 5.2.4 gir oversikt over de fysiologiske variablene i treningssettet, inkludert de fire vitaltegnene: `blodtrykk`, `hjertefrekvens`, `respirasjonsfrekvens` og `kroppstemperatur`.



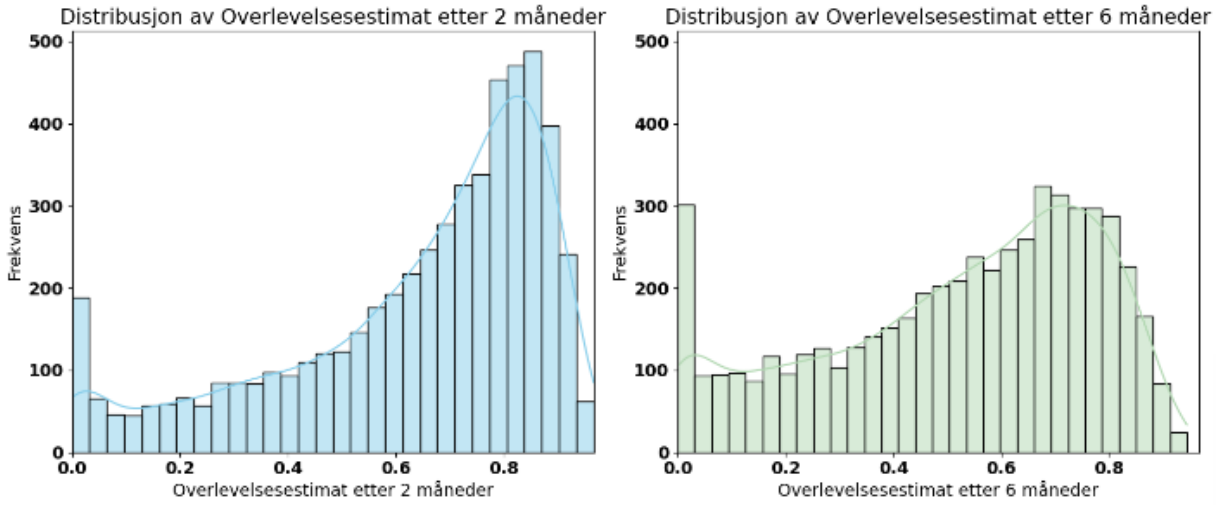
Figur 5.2.4: Boksplottene gir oversikt over diverse statistiske mål blant fysiologiske variabler, og identifiserer uteliggere.

Flere av variablene har uteliggere som kan påvirke modellens ytelse. Spesielt merker vi oss variablene `glukose`, `kreatinin`, `blod_ph` og `hvite_blodlegemer` som alle har relativt lav varians, og flere uteliggere. Dette kan tyde på at enkelte pasienter har ekstreme helseparametere som følge av alvorlig sykdommer. Disse verdiene bør vurderes nøye, da for mange uteliggere kan føre til redusert i prediksjoner. Jeg velger å ikke fjerne uteliggerne i de fysiologiske variablene i treningsdataene, da verdiene kan gi viktig informasjon om pasienters stressreaksjoner. Uteliggere kan forklare alvorlige helseproblemer, og bør generelt behandles med omhu for å unngå å gi et urealistisk bilde på hvordan sykdommer faktiske kan påvirker pasienter.

For de vitale tegnene observerer vi at `blodtrykk` har moderat varians, og få uteliggere. `hjertefrekvens` har også moderat varians, men også en del uteliggere med høye verdier. De spesielt høye verdiene er et tegn på stressreaksjon, og kan indikere alvorlig sykdom. `respirasjonsfrekvens` har også flere uteliggere av høye verdier. `kroppstemperatur` har moderat varians og uteliggere av både høye og lave verdier. Unormalt lave eller høye kroppstemperaturer er ofte assosiert med feber eller annen alvorlig sykdom. Om disse verdiene hadde blitt filtrert bort ville det ført til tap av viktig informasjon som forklarer de virkelige helseproblemene som forekommer hos pasienter.

5.2.6 Sammenligning av overlevelsesestimat: 2 vs. 6 måneder

Nå som vi har analysert både demografiske og helserelaterte variabler har vi grunnlag for å forstå deres sammenheng med overlevelsesestimatene. Diagrammene nedenfor viser fordelingen av overlevelsesestimater over 2 og 6 måneder blant pasienter i treningssettet. X-aksen representerer overlevelsesestimatene.



Figur 5.2.5: Histogrammene viser distribusjon av overlevelsesestimat etter 2 og 6 måneder.

Overlevelsesestimat over 2 måneder har en tydelig topp på rundt 0.8, noe som tyder på at mange av pasientene har gode prognoser. Det er et markant antall pasienter som har 0% i overlevelsesprognose. Disse pasientene er kritisk syke.

Fordelingen over 6 måneder er mer jevnt fordelt. Prognosene som var relativt gode etter 2 måneder svekkes over tid, noe som er forventet. Det er nå et enda høyere antall pasienter som har 0% overlevelsesestimat. Disse pasientene er kritisk syke, og har svært liten sannsynlighet for overlevelse i løpet av den gitte perioden. Modellen forventes å predikere en kortere oppholdslengde på sykehuset for pasienter med kortere overlevelsesestimat.

6 Modellering

Målet med modellering er å finne mønstre i treningsdataene som generaliserer godt til ny, usett data. Jeg ønsker å velge den modellen som presterer best på valideringsdataene, og undersøke modellens evne til å predikere sykehusopphold på testsettet (Blaser, 2023). Evalueringen vil bli utført ved bruk av rot middelkvadratfeil (RMSE), som forteller hvor mye modellens predikerte oppholdslengder avviker fra faktiske oppholdslengder.

Formelen for RMSE er:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

Her er $\hat{y}_i$ de predikerte oppholdslengdene, og y_i er de faktiske oppholdslengdene. N er antall pasienter, som tilsier antall observerte oppholdslengder.

Første steg for å utvikle maskinlæringsmodeller er å laste inn `sckit-learn (sklearn)`. Dette biblioteket inneholder verktøy for å utvikle, trene og effektivisere maskinlæringsmodeller.

6.1 Modellutvalg og forventninger

Når man velger ut modeller som skal predikere oppholdslengde, er det flere faktorer som spiller inn. Ulike modeller har forskjellige tilnærminger for evaluering av data, noe som kan gi varierende prediksjoner. Jeg har valgt å ta i bruk en grunnlinjemodell, og tre avanserte regresjonsmodeller, der den første er Gradient Boosting Regressor. Denne modellen er sensitiv til uteliggere, noe som kan svekke generaliseringsevnen og gjøre at den overtilpasser seg treningsdataene. Jeg forventer at denne modellen skal håndtere ekstremalverdier godt, men det er mulig at modellen overtilpasser seg treningsdataene, noe som reduserer generaliseringsevnen.

Random Forest Regressor er en modell som er mindre utsatt for overtilpasning, noe som gjør at modellen kan undertilpasse seg treningsdataene og styrke generaliseringsevnen. Av den grunn forventer jeg at denne modellen presterer bedre på sykehuspasienter med milde til moderate helseplage (GeeksForGeeks, 2024a).

Den siste modellen jeg velger å trene er Elastic Net. I forhold til de andre modellene, håndterer Elastic Net variabler korrelerer effektivt spesielt godt. I tilfellene der Random Forest Regressor velger en tilfeldig variabel fra en gruppe korrelerende variabler, vil Elastic Net prøve å gruppere og velge alle sammen (GeeksForGeeks, 2024b). Som en del av datatilbredningen har jeg allerede fjernet enkelte høyt korrelerende variabler, så det vil bli interessant å se om Elastic Net likevel klarer å utkonkurrere de andre modellene.

6.1.1 Grunnlinjemodell

Å definere en grunnlinjemodell er en essensiell del av modellutviklingsprosessen. Hensikten med denne modellen er at den skal bruke en enkle imputasjonsteknikk, og gi et grovt estimat på hvordan de mer komplekse modellene bør prestere. Grunnlinjemodellen avslører om de mer komplekse modellene faktisk er godt egnet til prediksjon av oppholdslengde. Jeg forventer at denne modellen blir utkonkurrert av de mer avanserte regresjonsmodellene. Dette er naturlig da modellen ikke fanger opp, og tilpasser seg, underliggende mønstre i treningsdataene.

Modellen opprettes ved en `Dummyregressor()` som gjennomfører prediksjoner ved å imputere medianverdi av oppholdslengden i `y_train`, uavhengige av input. Dermed vil alle prediksjoner for `X_val` være medianverdien til oppholdslengden, da dette er den eneste strategien modellen er trent på. Avslutningsvis sammenliknes de faktiske verdiene i `y_val` med prediksjonene.

RMSE-verdien for grunnlinjemodellen er lik `21.241`, og er et viktig referansepunkt når jeg skal trene mer avanserte modeller.

6.1.2 Gradient Boosting Regressor

Gradient Boosting Regressor er en regresjonsmodell som bygger trærne, som lærer av tidligere feil, sekvensielt. Hvert tre korrigerer for feilene som ble gjort i de tidligere trærne (GeeksForGeeks, 2024a). I våre mer avanserte modeller ønsker jeg å ta i bruk en `Pipeline()` for å teste konfigurasjonen og strukturen i modellene. Dette gir tre identifikatornavn for komponentene i konfigurasjonen. I første omgang er disse satt lik `passthrough` fordi de ikke har definert transformasjoner enda.

Neste steg involverer en vurdering av de beste kombinasjonene av hyperparametre i `gradient_boosting_parametre` ved hjelp av `RandomizedSearchCV()`. Hensikten er å oppnå den laveste RMSE-verdien. Komponentene inkluderer:

- `strat`: Verktøyet som brukes for å erstatte NaN-verdier med statistiske verdier.
- `strat_strategy`: strategier for imputering i `SingleImputer()`.

- `scaler`: Standardiserer data, noe som sørger for konsistent datastruktur.
- `model`: Hvilken modell som brukes. I dette tilfellet `GradientBoostingRegressor()`.
- `model_n_estimators`: Antall beslutningstrær i gradient boosting. Flere trær kan forbedre prediksjonsevne, men for mange kan føre til overtilpasning.
- `model_learning_rate`**:** Hastigheten på læringen. En for høy læringsrate kan gjøre modellen ustabil.
- `model_max_depth`**:** Maksimal dybde på hvert beslutningstre. Når et beslutnigstre er dypere, kan det gi bedre prediksjoner. Det kan også føre til overtilpasning.

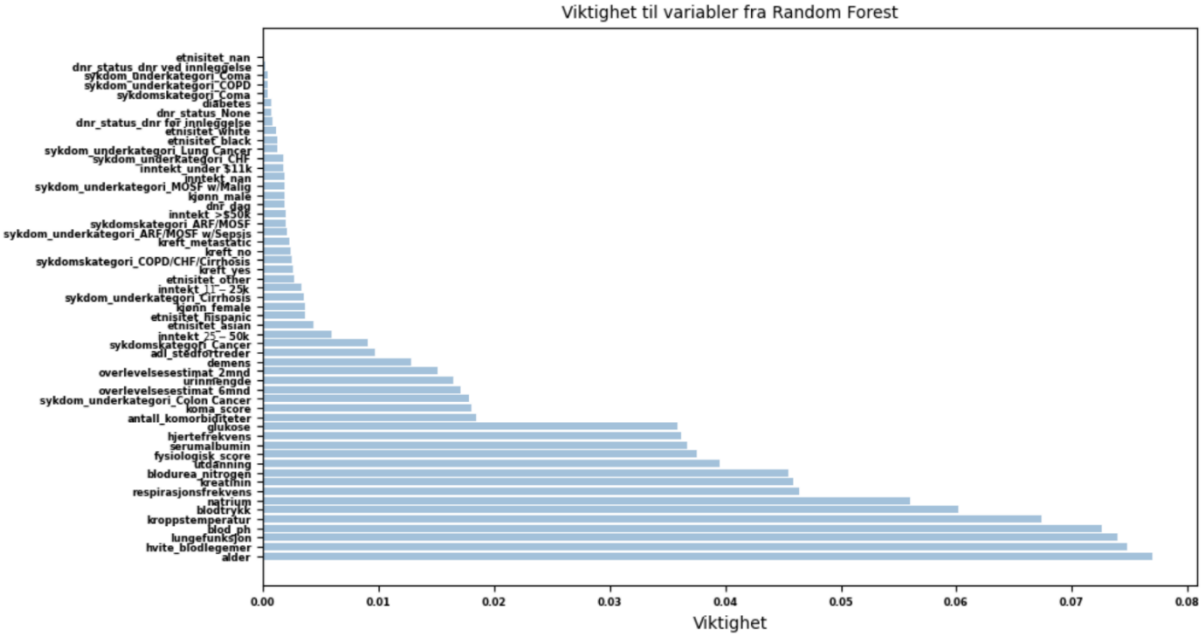
I tillegg brukes `KNNImputer()` som er et tilsvarende verktøy for imputering av NaN verdier. Når dataene deles inn i trenings-, validerings- og testdata settes `random_state=42`. For å være konsekvens og oppnå robuste resultater spesifiserer jeg den samme verdien i modellene mine også. Hvis ikke kan modellen gi uventende prediksjoner.

Den beste RMSE-verdien for Gradient Boosting er `20.035`, noe som er en klar forbedring fra RMSE-verdien i grunnlinjemodellen. Forbedringen tyder på at de avanserte teknikkene i Gradient Boosting Regressor tilpasser seg dataene bedre.

6.1.3 Random Forest Regressor

Random Forest Regressor er en regresjonsmodell som konstruerer flere uavhengige beslutningstrær. Hvert tre konstrueres på et tilfeldig utvalg av treningsdata er tilfeldig utvalgt av hyperparametere i `random_forest_parametre` (GeeksforGeeks, 2024a). I denne modellen har vi komponenten `model_min_samples_split` som bestemmer det minste antallet datapunkter som kreves for å dele en node i et tre.

Den beste RMSE-verdien i Random Forest Regressor er `19.982`, noe som er en forbedring fra både Gradient Boosting Regressor, som har en RMSE lik 20.035, og grunnlinjemodellen som har en RMSE lik 21.241. Variasjonen i RMSE illustrerer hvordan ulike modeller presterer ulikt.



Figur 6.1.3: Figuren viser variablenes innflytelse på modellens prediksjoner. Alder, antall hvite blodlegemer og kroppstemperatur påvirket modellens prediksjonsevne i størst grad.

6.1.4 Elastic Net

Elastic Net er en regresjonsmodell som kombinerer egenskapene i både Lasso-regresjon (L1) og Ridge-regresjon (L2). Denne kombinasjonen gir en mer fleksibel modell som kan håndtere multikolinearitet (GeeksForGeeks, 2024b). Denne modellen inneholder komponenten `model_alpha`, som bestemmer regulariseringsstyrken, og `model_l1_ratio`, som definerer forholdet mellom Lasso- og Ridge-regulariseringen.

Den beste RMSE-verdien for Elastic Net er `20.085`, som er betydelig bedre enn grunnlinjemodellen.

Modell	RMSE
Random Forest Regressor	19.982205
Gradient Boosting Regressor	20.034829
Elastic Net	20.084848
Grunnlinjemodell	21.241185

Tabell 6.1.4: Sammenlikning av RMSE for utvalgte regresjonsmodeller.

Ut ifra tabellen ser vi at Random Forest Regressor har den beste prediksjonsevnen. Grunnlinjemodellen presterer dårligst, som er forventet grunnet dens simple struktur, som ikke tar hensyn til komplekse sammenhenger i dataene. Siden Random Forest Regressor er den beste modellen, setter jeg denne `optimal_model`, og bruker denne når jeg skal beregne generaliseringsevne, predikere oppholdslengde til pasienter i `sample_data`, og på nettsiden .

6.1.4 Generaliseringsevne

Generaliseringsevnen beskriver modellens evne til å gi nøyaktige prediksjoner på ny, usett data som ikke er inkludert i treningssettet. Dette målet er helt sentral i praktiske sammenhenger, da den verifiserer at modellene kan gjøre troverdige prediksjoner på usett data. En modell kan for eksempel være overtilpasset treningsdataene, og dermed være dårlig egent til å utføre prediksjoner i virkelige situasjoner (What is Generalization in Machine Learning?, 2024).

I dette prosjektet ble den beste modellen Random Forest Regressor med en RMSE-verdi på `19.98` på valideringsdataene. Imidlertid, når modellen brukes på testdataene, resulterer det i en RMSE på `19.13`, noe som indikerer marginal forbedring i generaliseringsevne. Dette viser at modellen presterer godt på valideringsdataene og klarer å generalisere til usett data. Den marginale forskjellen mellom ytelse på valideringsdataene og generaliseringsevnen antyder at modellen er godt tilpasset uten tegn til overtilpassing.

```
Test RMSE for beste modell (random_forest): 19.130643601108797
```

7 Diskusjon av resultater

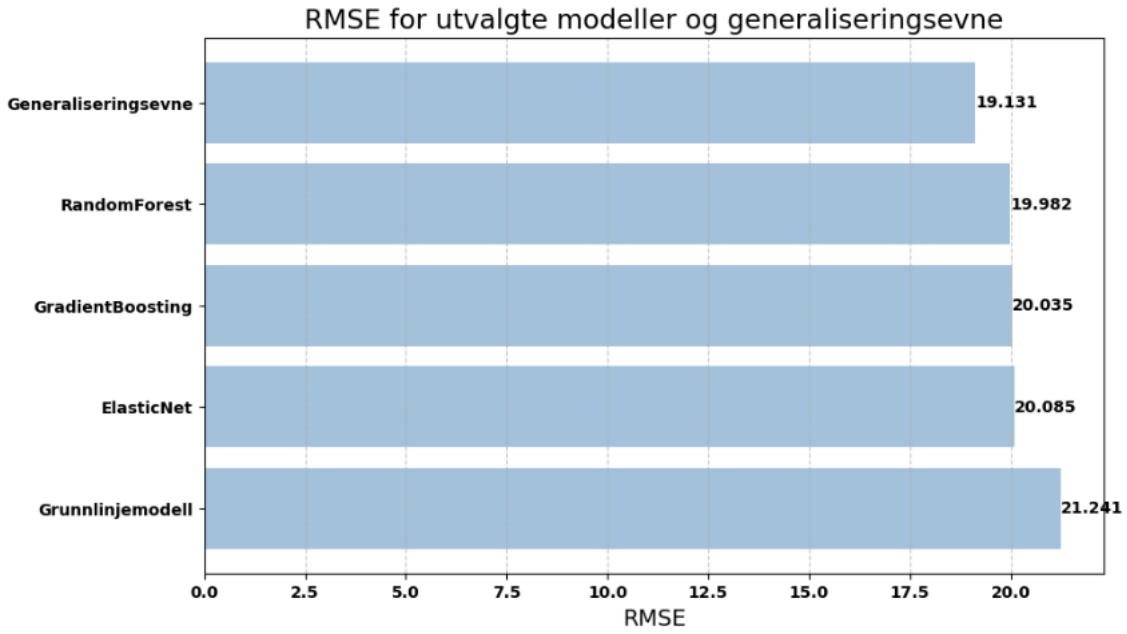
Under denne seksjonen diskuteres modellens prestasjon og ytelse, overraskelseselementer under modellering, og om modellen er praktisk anvendelig. Videre skal jeg diskutere modellens styrker og svakheter, samt inkludere forslag til forbedringer med ubegrenset tid.

7.1 Prestasjon av modell og ytelse

Random Forest Regressor viste seg å være den mest nøyaktige modellen, med en RMSE på 19.98. Sammenliknet med Gradient Boosting Regressor, leverte Random Forest Regressor mer stabile resultater på datasettet, og er generelt mindre sårbar for uteliggere og overtilpasning. Dataanalysen og visualiseringen bekreftet tilstedeværelsen av uteliggere i datasettet, og det var derfor ikke uventet at Random Forest Regressor skulle prestere best på sykehusdataene.

Gradient Boosting Regressor presterte nest best, med en RMSE på 20.035. Modellen klarte ikke å dra nytte av dens evne til å håndtere uteliggere, noe som gjorde at den ble utkonkurrert av Random Forest Regressor.

Elastic Net presterte dårligst av de mer avanserte modellene. Dette var forventet, da jeg fjernet flere av variablene som korrelerte mest under databehandlingen. Av den grunn fikk ikke modellen dra nytte av dens evne til å håndtere høyt korrelerende variabler. Det er verdt å merke seg at forskjellene i ytelse var marginale, som illustrert i figur 6.2.1.



Figur 6.2.1: Stolpediagrammet viser modellenes ytelse (RMSE) på treningsdata og generaliseringsevnen på testdata.

De mer avanserte modellene presterer bedre på både valideringsdataene og testdataene enn det grunnlinjemodellen gjør. Dette er forventet da regresjonsmodellene er bedre på å fange opp komplekse sammenhenger i datasettet, der grunnlinjemodellen kun bruker medianverdien for å gi en enkel og ufullstendig prediksjon.

7.2 Overraskelseselementer

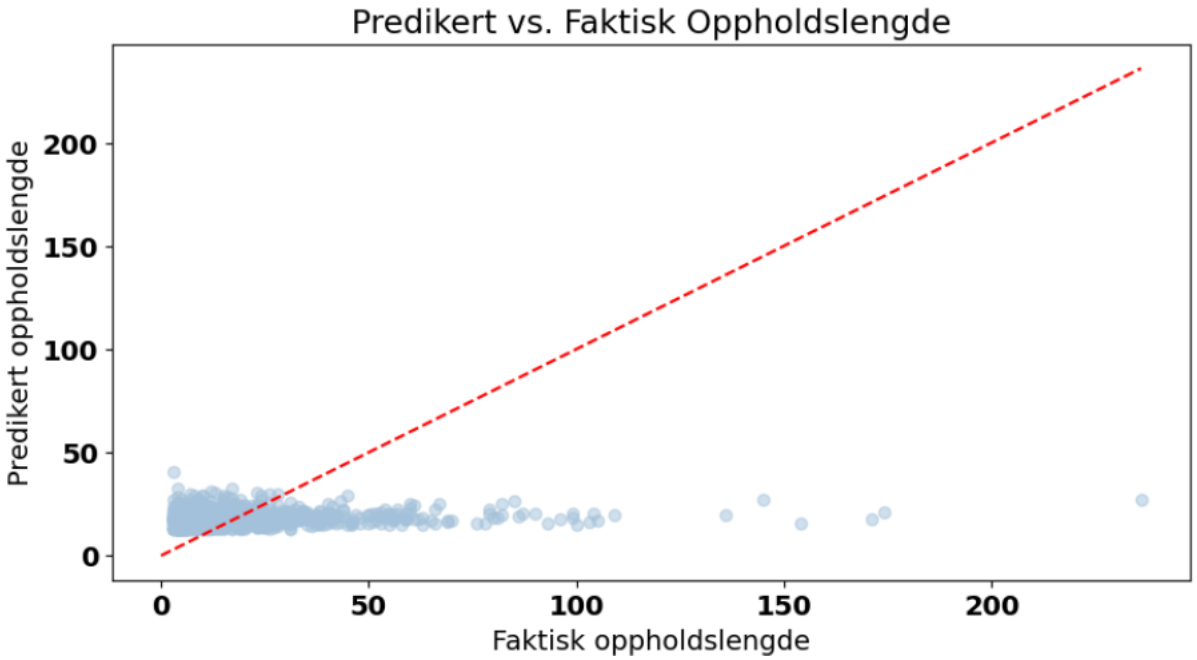
Variabelutvinning- og modelleringsprosessen avdekket overraskelser og interessante funn. I min første tilnærming til dataanalysen valgte jeg å erstatte negative oppholdslengder med medianoppholdslengde før jeg hadde delt i trenings-, validerings og testdatasett. Denne tilnærmingen resulterte i en generaliseringsevne som var betraktelig redusert sammenliknet med prediksjonsevnen på valideringsdataene.

I den endelige tilnærmingen fjerner jeg de negative oppholdslengdene etter at datasettet var delt. Dette førte til en marginalt dårligere prediksjonsevne på valideringsdataene, men en forbedret generaliseringsevne på testdataene. Dette kan trolig skyldes at når medianverdier erstatter negative verdier før datainndeling, introduseres skjevhet i treningsdatasettet som ikke er til stede i testdatasettet. Det er derimot overraskende at kun 6 negative oppholdslengder kan ha så stor innvirkning på RMSE.

I figur 5.2.4 var en oversikt over uteliggere i de fysiologiske variablene. I utgangspunktet valgte jeg å opprette et 95% konfidensintervall for å filtrere bort uteliggere, i håp om å redusere støy i datasettene. Denne tilnærmingen resulterte derimot i en nedgang i både prediksjonsevnen og generaliseringsevnen til modellene. Det er tydelig at uteliggerverdiene gir et realistisk bilde av helsetilstanden til de alvorlig syke pasientene, og bør derfor beholdes under modelltreningen.

7.3 Modellens styrker og svakheter

For å kunne forbedre modellens prediksjonsevne er det viktig å ha oversikt over når modellen presterer godt, og når den ikke gjør det. Figur 6.2.2 viser de predikerte oppholdslengdene sammenliknet med de faktiske oppholdslengdene, der den stiplede røde linjen representerer perfekt prediksjon.



Figur 6.2.2: Figuren viser den faktiske oppholdslengden til pasienter opp mot modellens predikerte oppholdslengde.

Figuren gir oss viktig informasjon om modellens styrker og svakheter. De predikerte oppholdslengdene varierer mellom `13-43 dager`. Det er tydelig at modellen underestimerer oppholdene når de faktiske verdiene blir høye. Det tyder på at modellen er dårlig på å forstå tilstanden til pasienter som er alvorlig syke, og krever omfattende

behandling. Denne underestimeringen er resultat av at modellen er trent på data der alvorlig syke er underrepresentert.

På den positive siden gir modellen mer troverdige resultater for flertallet av pasientene. Den faktiske median oppholdslengden er `11 dager`, mens den predikerte median oppholdslengden er `17.4 dager`**.** Dette tyder på at modellen generelt overestimerer oppholdslengden til pasienter med mer moderate symptomer. Disse resultatene indikerer at det er nødvendig å utføre mer omfattende analyser på pasienter med langvarige opphold for å forstå hva som kjennetegner denne gruppen pasienter, og tilpasse modellen deretter.

Ved praktisk anvendelse av modellen kan underestimering av alvorlig syke pasienters opphold ha flere alvorlige konsekvenser. For det første kan det føre til at pasientene blir utskrevet for tidlig, noe som øker sannsynligheten for komplikasjoner, gjeninnleggelse eller død. Dette fører til en dårligere opplevelse som pasient, og medfører at sykehuset allokerer ressursene sine ineffektivt. Over tid kan dette være kostnadsineffektivt, og skape negative ringvirkninger i helsevesenet. Av den grunn krever modellen bearbeiding og videreutvikling for å kunne behandle en bredere populasjon, og kunne anvendes i praksis.

7.4 Modellens tilfredstillelse og anvendelighet

Den simple grunnlinjemodellen predikerer at alle utfall vil være lik medianverdien til den uavhengige variabelen, uten å ta hensyn til mønstre i datasettet. Tidligere analyse viser at Random Forest Regressor presterer bedre enn grunnlinjemodellen, grunnet dens evne til å identifisere sammenhenger og optimalisering av hyperparametere. Det er tilfredsstillende at en kompleks regresjonsmodell er bedre på å kategorisere og finne mønstre blant pasienter i populasjonen.

Anvendeligheten av modellen er begrenset, derimot. Som figur 6.2.2 illustrerte, anvender modellen godt på pasienter med moderat sykdomsalvorlighet, men sliter med å fange opp mønstrene hos de mest alvorlig syke pasientene. Det viser seg at modellen kun klarer å gjøre tjueseks korrekte prediksjoner, noe som ikke er optimalt. Likevel er ikke målet med prediksjonene nødvendigvis å treffe eksakt med antall dager, men å minimere feilmarginen, noe modellen klarer til en viss grad.

Om man klarer å kategorisere pasienter basert på helseparametere, kan modellen være praktisk anvendelig på de pasientene med moderate til milde symptomer. For at modellen skal være praktisk anvendelig på et sykehus med en bredere populasjon, må modellen tilpasses til at den oppgir mer nøyaktige prediksjoner uavhengig av sykdomstype og alvorlighetsgrad.

7.5 Forslag til forbedringer med ubegrenset tid

For å forbedre modellenes ytelse på valideringsdataene, og testdataene, gitt ubegrenset tid, ville jeg finjustert databehandlingen, og utforsket flere hyperparametere. Videre kunne jeg undersøkt flere avanserte metoder innen feature engineering, som for eksempel Principal Component Analysis (PCA), for å redusere antall variabler og forenkle dataanalysen. Dette kan ivareta balansen mellom ytelse på valideringsdataene og testdataene, samtidig som modellen klarer å tilpasse seg uteliggere mer effektivt. Ved fremtidig datainnsamling kan det være relevant å filtrere inn alvorlig syke i datasettet, for å øke representativiteten, og forbedre generaliseringsevnen på en bredere gruppe pasienter.

En modell som er bedre tilpasset hele populasjonen, og som yter godt uansett helseparametere, kan være med på å effektivisere helsevernet. Dersom maskinlæringsmodellen reduserer avviket mellom faktiske og predikerte oppholdslengder, kan sykehuset allokere ressursene sine bedre, noe som fører til bedre pasientbehandling og reduserte kostnader.

8 Praktisk bruk av modellen

I denne delen vil jeg teste den praktiske anvendelsen av modellen på nye datasett og vurdere muligheten for interaktiv prediksjon på en nettside.

8.1 Prediksjon på `sample_data`

Mappen `sample_data` inneholder sykehusdata for helt nye pasienter. Datastrukturen er identisk som pasientdataene i `raw_data`, med unntak av oppholdslengden som mangler. Det er opp til modellen å predikere denne.

Databehandlingen av de nye dataene følger samme fremgangsmåte som i rådataen. Datasettene kombineres til en DataFrame og lagres som `sample_data_encoded`. Modellen brukes til å predikere oppholdslengde for hver pasient. Den predikerte oppholdslengden lagres i variabelen `predikert_oppholdslengde`. Det komplette datasettet, inkludert den predikerte oppholdslengden, lagres i filen `predicions.csv`.

8.2 Nettside prediksjon av oppholdslengde

Nettsiden for prediksjon av pasienters oppholdslengde basert på angitt pasientdata er bygget i `app.py`. For å integrere maskinlæringsmodellen i applikasjonen brukes biblioteket `pickle` til å eksportere modellen inn i applikasjonen.

På nettsiden kan brukere, som for eksempel sykepleiere og leger, velge egendefinert data for alle variablene i datasettet. Deretter predikerer modellen oppholdslengden for pasienten. Det kan være nyttig på et sykehus, da modellen kan predikere oppholdslengde på kun få sekunder. Modellen kan være med på å effektivisere prosesser i helsevesenet, redusere kostnader relatert til pasientopphold, og potensielt redde liv.

For simpelhetens skyld har jeg inkludert en knapp, Median verdier, som automatisk fyller inn medianverdiene for alle numeriske variabler. Nettsiden håndterer også eventuelle brukerfeil.

Referanser

SUPPORT2. UCI Machine Learning Repository. (2024). https://archive.ics.uci.edu/dataset/880/support2

Blaser, N. (2023). Nello Blaser. https://blasern.github.io/

GeeksforGeeks. (2024a, April 9). *Gradient boosting vs Random Forest*. https://www.geeksforgeeks.org/gradient-boosting-vs-random-forest/#basic-algorithm-of-gradient-boosting-vs-random-forest.

GeeksforGeeks. (2024b, June 5). *What is elasticnet in Sklearn?* https://www.geeksforgeeks.org/what-is-elasticnet-in-sklearn/

What is Generalization in Machine Learning? (2024). *What is generalization in machine learning?*. RudderStack. https://www.rudderstack.com/learn/machine-learning/generalization-in-machine-learning/