Laget av Nello Blaser

Contents

[**1**  **Introduksjon** 2](#_Toc181110158)

[1.1 Beskrivelse av datasettene 2](#_Toc181110159)

[**2**  **Innledende dataforberedelse** 3](#_Toc181110160)

[2.1 Klargjøring av data 3](#_Toc181110161)

[2.1.1 Demografiske data 3](#_Toc181110162)

[2.1.2 Fysiologiske data 4](#_Toc181110163)

[2.1.3 Sykehusalvorlighetsdata 4](#_Toc181110164)

[2.2 Kombinering av datasett 4](#_Toc181110165)

[2.3 Databehandling 5](#_Toc181110166)

[2.4 Inndeling av trenings-, validerings- og testdata 5](#_Toc181110167)

[**3** **Dummy-kodifisering** 6](#_Toc181110168)

[**4** **Utforskende dataanalyse – statistikk** 7](#_Toc181110169)

[4.1 Statistiske mål: Gjennomsnitt, standardavvik, kvantiler og mer 7](#_Toc181110170)

[4.2 Korrelasjonsmatrise (vurder og fjerne) 8](#_Toc181110171)

[**5** **Utforskende dataanalyse - visualisering** 8](#_Toc181110172)

[5.1 Demografiske variabler 9](#_Toc181110173)

[5.1.1 Alder 9](#_Toc181110174)

[5.1.2 Kjønn 11](#_Toc181110175)

[5.1.3 Inntekt 12](#_Toc181110176)

[5.2 Helserelaterte variabler 13](#_Toc181110177)

[5.2.1 Oppholdslengde 13](#_Toc181110178)

[5.2.2 Sykdomskategorier blant etnisiteter 14](#_Toc181110179)

[5.2.3 Korrelasjon mellom fysiologiske variabler 15](#_Toc181110180)

[5.2.4 Korrelasjon mellom variabler i sykdomsalvorlighet 16](#_Toc181110181)

[5.2.5 Fysiologiske data og uteliggere 18](#_Toc181110182)

[5.2.6 Sammenligning av overlevelsesestimat: 2 vs. 6 måneder 20](#_Toc181110183)

[**6** **Modellering** 21](#_Toc181110184)

[6.1 Modellutvalg 22](#_Toc181110185)

[6.1.1 Grunnlinjemodell 22](#_Toc181110186)

[6.1.2 Gradient Boosting Regressor 23](#_Toc181110187)

[6.1.3 Random Forest Regressor 24](#_Toc181110188)

[6.1.4 Elastic net 25](#_Toc181110189)

[6.1.4 Generaliseringsevne 25](#_Toc181110190)

[References 28](#_Toc181110191)

# **1 Introduksjon**

**Formålet** med denne oppgaven er å anvende alt materialet lært i INF161 til å bygge en maskinlæringsmodell som skal predikere sykehusoppholdet til pasienter. Denne modellen vil benytte pasientopplysninger fra fire ulike datasett, som sammen gir en helhetligforståelse av pasientens helse og personlige bakgrunn.

## 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 benyttes 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.** Demografiske data 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

Det fysiologiske datasettet inneholder en betydelig andel NaN-verdier, noe som kan skape utfordringer i senere faser, spesielt under modellutvikling og predikasjon av oppholdslengde. For å unngå tap av verdifull informasjon ved fjerning av datapunkter, erstattes NaN-verdiene med anbefalte eksempelverdier hentet fra en pålitelig kilde: <https://archive.ics.uci.edu/dataset/880/support2>.

### 2.1.3 Sykehusalvorlighetsdata

Dette datasettet kommer i et annet format enn de andre filene, 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. Vi inspiserer dataene ved å bruke **.info().** En rekke de numeriske kolonnene er av typen «object», som representerer tekstdata. I maskinlæringsmodellen ønsker vi flest mulig kolonner av numeriske verdier, da dette forenkler trening av modellene. For å konvertere disse kolonnene til numeriske verdier, bruker vi metoden .**to\_numeric()**.

Jeg identifiserer også 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 fjerne de negative verdiene før inndeling av trenings-, validerings- og testdata for å sikre at datasettet inneholder meningsfulle data før visualisering og modellering. Dette 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. 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 testdata og valideringsdata. 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 dataene er klare og pålitelige før de brukes i videre analyser, som ved bruk av maskinlæringsmodeller.

## 4.1 Statistiske mål: Gjennomsnitt, standardavvik, kvantiler og mer

Vi undersøker statistiske målene i treningsdata ved å bruke metoden **.describe().** 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.

🡪 forklare at du ikke velger å fjerne verdiene under visualisering av uteliggere. Det kan være spesielle medisinske årsaker som forsaker ekstremverdiene, og jeg velger av den grunn å ta hensyn til dem.

## 4.2 Korrelasjonsmatrise (vurder og fjerne)

A screenshot of a computer screen

Description automatically generatedEn korrelasjonsmatrise er et nyttig verktøy for å undersøke sammenhenger mellom variablene i datasettet. Den kan hjelpe med å identifisere potensielle årsaksforhold og avdekke skjulte mønstre som ellers er vanskelige å oppdage. Følgende er en liten utskrift av korrelasjonsmatrisen.

Korrelasjonsmatrisen for treningsdataene består av 63 x 63 = 3969 verdier. De fleste av disse verdiene viser lav korrelasjon. Likevel legger jeg merke til enkelte variabler som korrelerer sterkere med hverandre. Et eksempel er variablene *sykdom underkategori lung cancer* og *kreft metastatic*, som har en korrelasjonsverdi på 0.665739 Dette kan være kritiske funn, da det er viktig å eliminere variabler som er sterkt korrelerte. For høye verdier kan føre til overestimater i maskinlæringsmodellen, noe som kan svekke modellens prediksjonsevne.

# 

# **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 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

**A graph of a graph

Description automatically generated with medium confidence**Alder har ofte sammenheng med sykdomsalvorlighet og oppholdslengde å gjøre da kroppen og immunforsvaret blir svakere med alderen. Å få oversikt over aldersfordelingen blant pasientene kan være aktuelt fordi forskjellige aldersgrupper kan ha varierende behandlingsbehov.

Figur 5.1.1: Histogram som illustrerer aldersfordelingen basert på kjønn.

Fra histogrammet ser det ut som det er en stor andel pasienter i alderen 60-80 år, både menn og kvinner. Vi merker oss at det er minst en pasient registrert med negativ alder.

**A graph of different colored squares

Description automatically generated with medium confidence**

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

Denne verdien er bemerkelsesverdig og må tas hensyn til i databehandling for at modellene skal yte best mulig. Videre kan fordelingen over oppholdslengde for ulike aldersgrupper være aktuelt å undersøke for å allokere ressurser bedre og effektivisere behandling.

Som vi ser, er oppholdslengden på sykehuset kortest blant de eldste pasientene. Dette forventet, da dødeligheten blant de eldste ofte er høyest som følge av svekket immunforsvar og flere helseproblemer. Det er imidlertid verdt å merke seg at forskjellene ikke er dramatiske. Dette tyder på at selv om alder er en faktor, finnes det andre variabler som også spiller en betydelig rolle på oppholdslengden. For å få en bedre forståelse av disse faktorene, vil vi undersøke videre for å undersøke hvordan andre demografiske og fysiologiske variabler påvirker oppholdslengden.

### 5.1.2 Kjønn

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

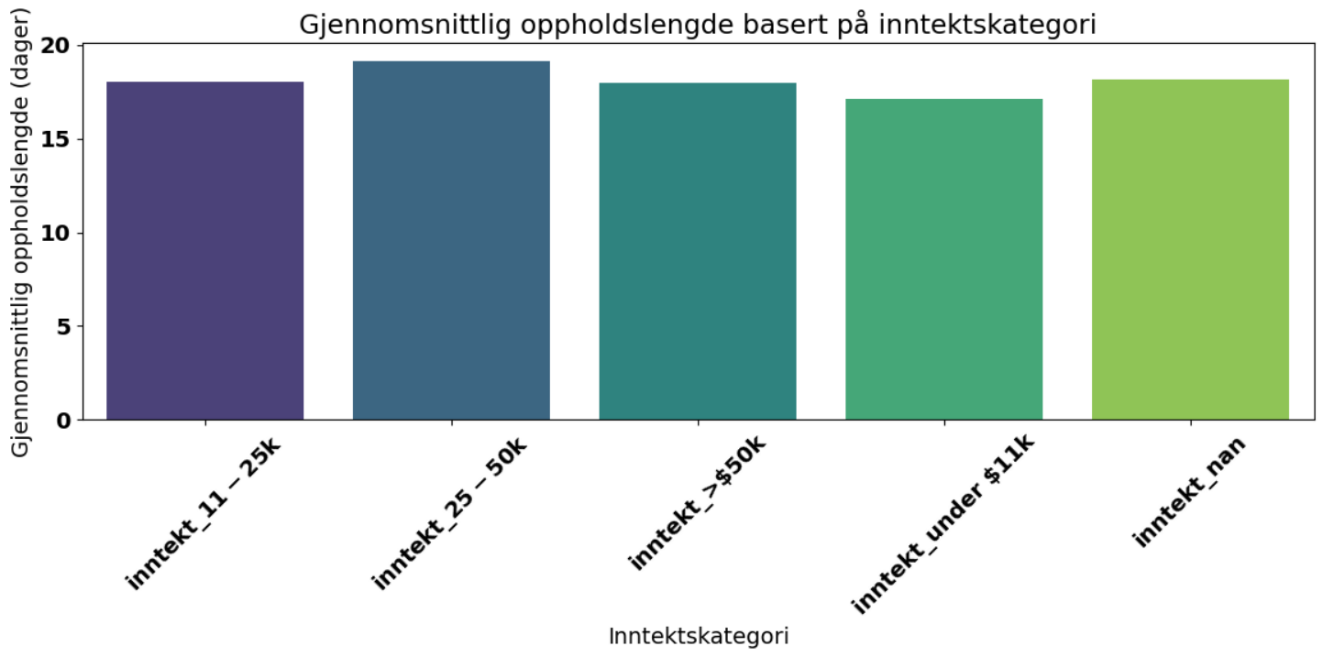
A screenshot of a graph

Description automatically generated

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 videre om andre sosioøkonomiske faktorer har innvirkning på helseutfall.

### 5.1.3 Inntekt

Sosioøkonomiske variabler, som inntekt, kan ha innvirkning på sykdomsforekomster og alvorlighetsgrad. Hypotese min 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: Stolpediagrammet illustrerer gjennomsnittlig oppholdslengde basert på inntektskategori.

Diagrammet viser at det er marginale forskjeller i oppholdslengde på tvers av inntektskategoriene. 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.

## 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 er på omtrent 18 dager, og det er relativt lav varians. Likevel er det flere interessante uteliggere, blant annet flere 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.

A graph of a graph

Description automatically generated with medium confidenceBoksplottet viser en interessant observasjon: minst én pasient har en negativ oppholdslengde. Dette er ikke realistisk, så jeg bytter derfor ut alle negative verdier for medianverdien for oppholdslengde i treningssettet.

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

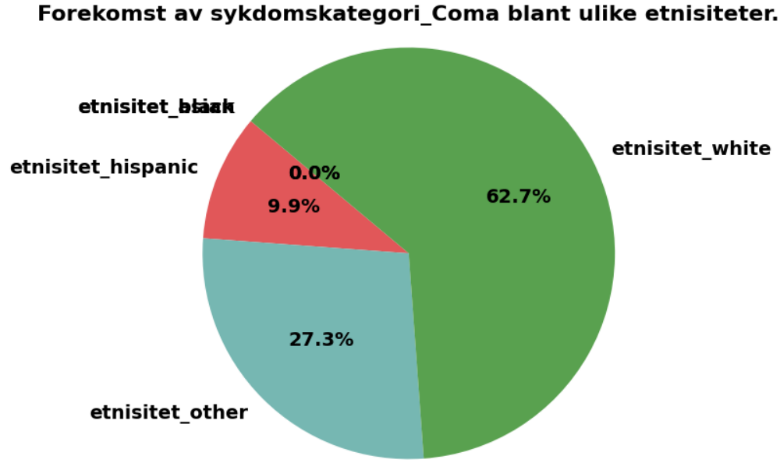
### 

### 5.2.2 Sykdomskategorier blant etnisiteter

A pie chart with numbers and text

Description automatically generatedA pie chart with text on it

Description automatically generatedÅ undersøke sykdomskategorier blant pasienter med ulike etnisiteter er viktig for å forstå om visse etnisiteter har økt sårbarhet 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.

A pie chart with different colored circles

Description automatically generated

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

Noen resultater er verdt å merke seg, blant annet andelen av **etnisitet\_white** som faller under sykdomskategori kreft og koma. Dette foreslår mulige sammenhenger mellom etnisiteter og sykdomskategori. Slike funn kan hjelpe sykehuset å predikere sykehusoppholdet opp mot sykdomskategoriene 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 de ulike fysiologiske variablene. 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 høyt korrelerte variabler. 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.

**A chart with different colored squares

Description automatically generated with medium confidence**

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). Dette er variabler som muligens bør elimineres for å forhindre «overfitting» i modellen, og sørge for at den gir mer nøyaktige resultater.

### 5.2.4 Korrelasjon mellom variabler i sykdomsalvorlighet

A screenshot of a data table

Description automatically generatedFor å unngå overtilpasning i modellen, er det viktig å få oversikt over korrelasjon mellom variablene i sykdomsalvorlighetsdataene. 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.

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 viktig for å unngå overtilpasning, og forbedrer modellens evne til å predikere oppholdslengde.

### 5.2.5 Fysiologiske data og uteliggere

A group of graphs with numbers

Description automatically generated with medium confidenceFor 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 treningsdataen inneholder mye uteliggere, kan det føre til dårlig generaliseringsevne, og økt risiko for overtilpasning. Nedenfor er boksplott over variablene i fysiologiske data, inkludert de fire vitaltegnene blodtrykk, hjertefrekvens, respirasjonsfrekvens og kroppstemperatur.

Figur 5.2.4: Boksplottene gir oversikt over diverse statistiske mål blant fysiologiske variabler, inkludert identifisering av 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 mange uteliggere. Dette kan tyde på at enkelte pasienter har ekstreme verdier som følge av alvorlig sykdom. Disse verdiene bør vurderes nøye, men jeg velger å ikke fjerne uteliggerne i de fysiologiske dataene, 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 blodtrykket 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 være 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

A comparison of a graph

Description automatically generated with medium confidence  
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. 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 data som generaliserer godt til nye data, usette data. Vi ønsker å velge den beste modellen for datasettet og evaluere modellens evne til å predikere sykehusopphold på testdata. (Blaser, 2023). Evalueringen vil bli utført ved bruk av rot middelkvadratfeil (RMSE), som forteller hvor mye modellens predikerte verdier avviker fra faktiske verdier.

Formelen for RMSE er:

Her er de predikerte oppholdslengdene, og er de faktiske oppholdslengdene. N er antall pasienter, som tilsier antall observerte oppholdslengder.

Første steg for å bygge maskinlæringsmodeller er å laste inn maskinlæringsbiliotektet **sckit-learn (sklearn)**. Biblioteket inneholder verktøy for å utvikle, trene og effektivisere maskinlæringsmodeller.

## 6.1 Modellutvalg

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 predikasjoner. I denne seksjonen skal jeg begrunne mine valg av modeller, inkludert en grunnlinjemodell, Gradient Boosting Regressor, Random Forest Regressor og Elastic Net, **og hvorfor disse modellene presterer godt på sykehusdata. 🡪 svar på denne!!!**

### 6.1.1 Grunnlinjemodell

Å definere en grunnlinjemodell er en essensiell del av modellutviklingsprosessen. Hensikten med denne modellen er at den skal bruke enkle imputasjonsteknikker, 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.

Modellen opprettes ved en **Dummyregressor** som gjennomfører prediksjoner ved å imputere medianverdi av oppholdslengden i **y\_train**, uavhengige av input. Alle prediksjoner på **X\_val** vil derfor være medianverdier, da dette er den eneste strategien modellen er trent på. Avslutningsvis sammenliknes de faktiske verdiene i **y\_val** med prediksjonene.

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

### 

### 6.1.2 Gradient Boosting Regressor

Gradient Boosting Regressor er en regresjonsmodell som bygger trærne sekvensielt, der 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 predikasjonsevne, men for mange kan føre til overtilpasnin.
* **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. Når dataene deles inn i trenings-, validerings- og testdata settes **random\_state=42**. For å få konsekvens og reproduserbarhet spesifiserer jeg den samme verdien i modellene mine også. Hvis ikke kan modellen gi uventende prediksjoner.

Den beste RMSE-verdien for Gradient Boosting var **20.036**, 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.

A graph of a number of objects

Description automatically generated with medium confidenceDen beste RMSE-verdien i Random Forest Regressor var **19.998**, noe som er en forbedring fra både Gradient Boosting Regressor, som hadde en RMSE på 20.036, og grunnlinjemodellen. Variasjonen i RMSE illustrerer hvordan ulike modeller presterer ulikt.

Figur 6.1.3: Variablenes innvirkning på modellens prediksjoner.

### 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, noe vi har bevist er tilfelle i korrelasjonsmatrisene (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 var **20.085**, som er betydelig bedre enn grunnlinjemodellen.

|  |  |
| --- | --- |
| **Modell** | **RMSE** |
| Random Forest Regressor | 19.998797 |
| Gradient Boosting Regressor | 20.036169 |
| Elastic Net | 20.084848 |
| Grunnlinjemodell | 21.241185 |

Tabell 6.1.4: Sammenlikning av RMSE for valgte 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.

Ut fra tabellen ser vi at Random Forest Regressor har den beste prediksjonsevnen, noe som kan tilskrives dens evne til å håndtere ikke-lineære relasjoner og multikolinearitet effektivt. Grunnlinjemodellen presterer dårligst, som er forventet på grunn av dens enkle struktur, som ikke tar hensyn til de komplekse sammenhengene i dataene.

### 

### 6.1.4 Generaliseringsevne

Generaliseringsevnen beskriver modellens evne til å gi nøyaktige predikasjoner 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. En modell kan for eksempel være overtilpasset treningsdataene, og dermed være dårlig egent til å utføre predikasjoner 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 19.99 på treningsdataene. Imidlertid, når modellen brukes på testdataene, resulterer det i en RMSE på 19.21, noe som indikerer marginal forbedring i generaliseringsevne. Dette viser at modellen presterer godt på treningsdataene og klarer å generalisere til usett data. Den marginale forskjellen mellom ytelse på treningsdataene og generaliseringsevne antyder at modellen er godt tilpasset uten tegn til overtilpassing.

For å forbedre modellenes ytelse på treningsdata, og generaliseringsevnen, gitt ubegrenset tid, ville jeg finjustert databehandlingen, og utforsket flere hyperparametere. Jeg kunne også undersøkt flere metoder innen feature engineering, som for eksempel Principal Component Analysis (PCA), for å redusere antall variabler og forenkle dataanalysen. Dette ville bidra til at modellen ikke overtilpasser seg treningsdataene, samtidig som den yter godt på usett data.

## 6.2 Diskusjon av resultater

6.3 Overraskende resultater

6.2 Analyse av resultater 🡪 på denne delen er det viktig å svare i samsvar med prosjektrapportering og prosjektbeskrivelse.

* Tilfredsstillende resultat 🡪 modell bedre enn baselinemodell
* Viktig kilde for å diskutere resultater. <https://www.geeksforgeeks.org/gradient-boosting-vs-random-forest/#performance-of-gradient-boosting-vs-random-forest>

Se blant annet: «handling Overfitting of Gradient Boosting vs Random Forest» og «Performance of Gradient Boosting vs Random Forest».

6.3 Overraskende resultater

Når resultater skal diskuteres:

**NB:** Modellutvelgelse er en viktig del av oppgaven og vil bli vurdert deretter. Før du anvender maskinlæringsalgoritmer, bør du alltid vurdere (og rapportere) hvilke resultater du forventer. Når du har brukt maskinlæringsalgoritmer, bør du alltid kommentere hvor godt resultatene samsvarer med forventningene dine.

**NB pt. 2:** Ved å sette forskjellige seeds får man drastiske endringer i RMSE. Det bør derfor argumenteres for hvorfor RMSE ikke er altfor betydelig. Heller viktig å være konsekvent. Også argumentere for hvilke imputasjonsteknikker/kryssvalideringsteknikker osv som kunne forbedret RMSE.

Har prøvd å fjerne inntektskategorier og etnisiteter som ifølge analysen ikke skulle ha betydelig innvirkning på oppholdslengden. Dette førte til høyere/dårligere RMSE. Hva kan gurnnen til dette være? DISKUTER. Diskuter med SEED også.

# References

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/>

1. Baseline modell (enkel, men realistisk)
2. Bruk av pipeline for fleksibel modellkjøring.

* Å ha en pipeline gjør prosessen for databehandling og modellering enklere og mer organisert.
* Pipeline konfigureres med plasserholdere for imputering (strat), skalering (scaler) og modellvalg (model). Dette gjør det mulig å bytte ut ulike deler dynamisk i RandomizedSearchCV.
* Koden utforsker både SimpleImputer (for å bruke mean og median) og KNNImputer som imputeringsteknikker.

1. Hyperparameter-tuning med RandomizedSearchCV

* RandomizedSearchCV utfører et systematisk søk etter de beste hyperparametrene i GradienBoostingRegerssor, inkludert valg av imputering, antall estimatorer, læringsrate og tre-dybde. Dette gjør modellen bedre tilpasset datasettene.

Sjekke for beste modell (inkludere R^2)???

1. Sjekker generaliseringsevne på testdata

* Det er viktig å evaluere hvordan den valgte modellen presterer på usett data for å vurdere generaliseringsevne.
  + Kan bruke krystallisasjon på treningssettet for å redusere overfitting.