Rapport – prosjekt INF161

Målet for prosjektet er å bygge en maskinlæringsmodell for å predikere den forventete lengden på sykehusoppholdet per pasient. Modellen vil bruke pasientopplysninger i form av fysiologiske, demografiske og sykdomsalvorlighetsdata på tvers av ni sykdomskategorier for å predikere forventet oppholdslengde. Målet er å anslå lengden på sykehusopphold for nye pasienter basert på disse variablene.

**Tilnærming:**

Tilnærmingen til dette prosjektet var å begynne å lese inn rådata fra de ulike filene om demografi, fysiologisk data og sykehusdata. Videre kombinerte jeg filene i et felles datasett som videre ble delt inn i trenings, validerings og testdata. Videre vil jeg visualisere korrelasjonen mellom ulike variabler og oppholdslengde for å

**Treningsdata:**

For treningsdatasettet var det en utfordring at det fantes både numeriske og kategoriske variabler. For visualisering trengs numerisk data så derfor måtte jeg transformere de kategoriske variablene til numeriske. Først brukte jeg get\_dummies ( ) funksjonen for å gjøre om eksempelvis variabelen kreft som opprinnelig hadde verdiene «yes», og «no», til to nye variabler «kreft\_no», og «kreft\_yes», med tallverdiene 0 og 1 som verdier. Dette funket som ønsket, men ettersom målet er en maskinlæringsmodell vil get\_dummies( ) være problematisk å bruke av flere grunner.

1. Inkonsekvente kolonner over ulike datasett hvor ulike kolonner kan dannes for samme kategori av data som skaper problemer når modellen skal predikere oppholdslengde
2. Get\_dummies() klarer ikke å behandle nye, usette kategorier
3. Get\_dummies() er ikke kompatibelt med Scikit-learn pipelines

Derfor brukte jeg heller OneHotEncoder som er mer fleksibelt og sikrer konsekvente kolonner over flere datasett, har alternativer for å behandle usette data og integrerer i pipelines.

**Manglende data:**

Undersøker først hvor mange manglende verdier det er i hver kolonne, før jeg begynner å fjerne data som ikke gir mening. Fjerner derfor alle pasienter som har negativ oppholdslengde og duplikater. I tillegg velger jeg en terskelverdi på 30%, og fjerner alle kolonner som har mer enn 30% manglende verdier. Fortsetter så med å fjerne urelevante data for oppholdslengde som vil forvirre modellen og føre til overfitting, dermed fjerner jeg etnisitet, inntekt og utdanning. Pasienter hvor oppholdslengde mangler fjernes også.

**Visualisering av treningsdata:**

Velger ut ulike modeller for visualisering. Lager et enkelt scatterplot med en variabel i sammenheng med oppholdslengde. Valgte her kroppstemperatur bare for å ha et eksempel. Ser også på et barplot med de topp 10 sterkeste korrelasjonene av variabler med oppholdslengde. Dette får jeg ved å bruke en korrelasjonsmatrise «corr( )». Ønsker å visualisere de variablene som har størst korrelasjon med oppholdslengde, både på positiv side og negativ side. I tillegg visualiserer jeg fullstendig korrelasjonsmatrise.

**Trening av modeller og imputering:**

Trener først en baseline modell og ser på RMSE som jeg får på tilnærmet lik 20.6. Forsøker så å trene en Random Forest modell og sorteres variablene i et plot etter viktighet for modellen. Dette gir meg et innblikk i hvilke variabler som er viktigst for modellen og som modellen hovedsakelig bruker for å predikere oppholdslengde.

Videre ønsker jeg å imputere for manglende verdier i X\_train og forsøker flere ulike strategier i kombinasjon med ulike modeller for å sammenligne.

Sjekker først R^2 score som er et mål på hvor godt modellen forklarer variasjon i data. Ser her at jeg får ganske lave verdier (1 er en perfekt modell), og det høyeste jeg får er ved gjennomsnittsimputering (0,1257)

Videre sjekker jeg MSE for både Random Forest og lineær regresjon med 6 ulike imputeringsstrategier. Som resultat får jeg at å fjerne all manglende date er den beste imputeringsstrategien både for lineær regresjon og Random Forest med MSE på henholdsvis 105,12 og 117,04.

Sjekker videre gjennomsnittsimputering for flere ulike modeller, og ser at ved denne imputering så gir GradientBoosting best resultat, altså lavest MSE på 357,47, dette er svært høyt når målet er å predikere antall dager og standardavviket er på nesten ett år. Ser på ulike modeller ved KNN imputering og får igjen at GradientBoosting er den beste modellen med MSE på 363,6 som igjen er svært høyt.

Undersøker videre hvilke variabler som har høyest korrelasjon for å se hva som blir vektlagt mest av modellene. Ser her at pasient\_id og kjønn har høye koreelasjoner som forvirrer modellen, ting som demens er også mulig ikke relevant for oppholdslengde, men modellen finner høy korrelasjon som tyder påå overfitting. At modellene får såpass høy MSE tyder på overfitting og for mye data som forvirrer modellen og imputeringsstrategiene.

Videre tester jeg accuracy på ulike modeller med ulike imputeringsstrategier. Den beste modellen er Gjennomsnitt + Logistisk regresjon med en accuracy på 0.097 på valideringssettet. Tester igjen med flere modeller og får at den beste modellen da er knn + svm med en accuracy på 0.104 på valideringssettet.

**Variabelutvinning:**

For variabelutvinning tar jeg utgangspunkt i verdier som kan skape forvirring for modellen og forsøker å sammenfatte disse på best mulig måte. Til å begynne med lagde jeg en ny kolonne kalt «kreft» hvor 1 betyr at pasienten har kreft og 0 betyr at pasienten ikke har kreft. Dette gjør jeg for å forkorte kolonnene kreft\_yes og kreft\_no til 1 enkelt kolonne. I tillegg legger jeg inn en kolonne for å signalisere om pasienten har høy hjertefrekvens eller ikke (hjertefrekvens over 100) da dette har større betydning for pasientens fysiologiske tilstand. Velger også å lage en ny kolonne som sammenfatter hjertefrekvens og respirasjonsfrekvens da dette igjen kan ha stor betydning for pasientens helse. I tillegg velger jeg å slå sammen flere av de fysiologiske sykdomskategoriene til en felles som jeg kaller total sykdomsbelastning. Gjør dette både for X\_train og X\_val da begge datasett må ha samme antall kolonner med like navn for at jeg kan kunne trene og bruke en modell til å predicte y\_pred utfra X\_val.