Rapport

# Titanic katastrofen- ECON 3170

## *Introduksjon*

I denne oppgaven skal vi - ved hjelp av ulike maksinlæringsmodeller - forsøke å predikere om passasjerer på Titanic overlevde eller ikke. Datasettet er hentet fra kaggle, og oppgaven er inspirert av Kaggles maksinlæringskonkurranse (Kaggle). Vi innleder med å innspisere datasettet, og forsøke å få oversikt over de ulike variablene, og omfanget av manglende verdier. Vi vil så forsøke å behandle de manglende verdiene, og tilpasse de slik at de blir enklere å bruke i treningen av maskinlæringsmodellene. Videre vil vi inspisere datasettet ved hjelp av plots for å forsøke å se sammenhengen mellom ulike variabler, og andelen overlevende. Vi vil så tilpasse parametere og hyperparametere for å forsøke å få så presise modeller som mulig. Til slutt vil vi implementere modellene i et dashboard hvor en kan finne overlevelsessansynlighet for en selvvalgt karakter.

## Pakker

I prosjetet kommer vi hovedsaklig til å benytte oss av to pakker: tidyverse og tidymodels. Begge pakkene er samlinger av mange ulike pakker. Vi kommer i hovedsak til å bruke Tidyverse til manipulering og visualisering av data, og Tidymodels til maskinlæring.

library(tidyverse)   
library(tidymodels)

## Data

titanic <- suppressMessages(read\_csv("Titanic-Dataset.csv"))  
glimpse(titanic[1,])

Rows: 1  
Columns: 12  
$ PassengerId <dbl> 1  
$ Survived <dbl> 0  
$ Pclass <dbl> 3  
$ Name <chr> "Braund, Mr. Owen Harris"  
$ Sex <chr> "male"  
$ Age <dbl> 22  
$ SibSp <dbl> 1  
$ Parch <dbl> 0  
$ Ticket <chr> "A/5 21171"  
$ Fare <dbl> 7.25  
$ Cabin <chr> NA  
$ Embarked <chr> "S"

#NOTE: “suppressMessages”, for å unngå output melding.

Datasettet består av 891 observasjoner med 12 variabler, og er en oversikt over et utvalg av passasjerene ombord på Titanic. Første kolonne er PassengerId som gir hver observasjon en unik ID. Den andre kolonnen er Survival; en binær variabel som indikerer om passasjeren overlevde (1) eller ikke (0). Tredje kolonne er Pclass som forteller hvilken klasse passasjeren var på. Fjerde kolonne, Name, viser navnene til hver enkelt passasjer. Kolonne fem, Sex, er en kategorisk variabel som betegner kjønnet til passasjeren. Den sjette kolonnen, Age, er en kontinuerlig variabel som gir alderen til passasjeren. Kolonne sju, SibSp, er en variabel som viser antallet søsken eller ektefeller ombord på Titanic. Kolonne åtte, Parch viser antallet foreldre eller barn ombord.Nienede kolonne, Ticket, viser bilettnummeret til passasjeren Tiende kolonne, Fare, betegner prisen passasjeren brukte på biletten. Kolonne elleve, Cabin, er lugarnummeret til passasjeren. Embarked, den tolvte kolonnen er en kategorisk variabel som viser hvor passasjeren gikk på. hvor “C” er Cherbourg, “Q” er Queenstown og “S” er Southampton.

Kilde: (Kaggle)

head(titanic)

# A tibble: 6 × 12  
 PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin  
 <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <chr>  
1 1 0 3 Braund… male 22 1 0 A/5 2… 7.25 <NA>   
2 2 1 1 Cuming… fema… 38 1 0 PC 17… 71.3 C85   
3 3 1 3 Heikki… fema… 26 0 0 STON/… 7.92 <NA>   
4 4 1 1 Futrel… fema… 35 1 0 113803 53.1 C123   
5 5 0 3 Allen,… male 35 0 0 373450 8.05 <NA>   
6 6 0 3 Moran,… male NA 0 0 330877 8.46 <NA>   
# ℹ 1 more variable: Embarked <chr>

Pclass og Fare virker veldig spennede, de fleste har nok sett filmen og der blir tydlig fremstilt at hvis du er i første klasse har du en større sjanse for å overleve. Samme med Fare, hvis du har betalt mye for biletten indikerer dette en høyere klasse som kan hinte til en større sannsynlighet for å overleve. Cabin og Ticket er nok også verdier som har en sterk tilknytting med hvilken klasse du er i og hvor mye du betalte. Likevel blir det vannskelig å skulle si om noen overlever ut ifra hvilket bilettnummer man har. Ved første øyekast ser det ut som at Cabin variablen har mange manglende verdier. Hva den faktiske lugaren var er nok kanskje ikke altfor interessant, men hvilket dekk den ligger på kunne vært av stor interesse. Ettersom at dekket har mye å si for om du klarte å komme deg ut i tide. Alder og kjønn er igjen ganske interessante variabler. Vi kjenner jo alle til “Kvinner og barn først”. Her vil det nok være interessant å behandle dataen for å fremmheve om vedkommende er et barn eller ikke. Et annet aspekt kan også være om personen blir ansett som gammel eller ikke, ettersom at det ikke er utenkelig at eldre får prioritet under evakuering av skipet. Hvilken betydning navnet har for selve overlevelsen er vanskelig å si, men er trolig ikke av noe særlig stor betydning. På en annen side kan titler være nyttig for å estimere alderen til observasjoner hvor alderen mangler, ettersom man kan anta at titler som Master., Mr., Mrs., og Miss., følger en viss aldersfordeling. Embarked er igjen variabel det blir litt vanskelig å si noe om, ettersom at det er kun tre forskjellige steder, som egentlig ikke burde ha særlig stor betydning. På den andre siden kan det være en korrelasjon mellom hvor man gikk om bord og hvilken klasse man er i. Parch og Sibsp er en ganske interessante variabler, fordi det ikke er utenkelig at familiestørrelse og overlevelse kan ha en sammenheng. PassengerId er i grunnen ikke en veldig relevant variabel, og er trolig mer for å referere til den enkelte passasjer, og har etter all sannsynlighet ingenting med overlevelse å gjøre.

*Inspirasjon (Allohvk)*

## Utforsking av dataen

### Behandling av manglende verdier(NA`s)

skimr::skim(titanic) |>   
 select(skim\_variable, n\_missing)

# A tibble: 12 × 2  
 skim\_variable n\_missing  
 <chr> <int>  
 1 Name 0  
 2 Sex 0  
 3 Ticket 0  
 4 Cabin 687  
 5 Embarked 2  
 6 PassengerId 0  
 7 Survived 0  
 8 Pclass 0  
 9 Age 177  
10 SibSp 0  
11 Parch 0  
12 Fare 0

Vi bruker funksjonen skim fra skimr pakken for å skaffe en oversikt over dataen. Det mest interresante i utskriften er kolonnen “n\_missing” som viser hvor mange manglende verdier det er per variabel. Dersom vi ser på variabelen “Cabin” ser vi at den har 687 manglende verdier. Det er rimelig å anta at selve lugarnummeret ikke er så viktig for å predikere overlevelse, og at variabler som Pclass da vil være til større hjelp. Dersom man hadde gjort noen antagelser om hvordan dekkene var organisert kunne en brukt for eksempel Pclass og Fare til å forsøke å estimere lugarnummeret, men ettersom dette vil bli noe upresist og kunne resultert i uheldige utslag på prediksjonene - som følge av et lite datasett - velger vi istedenfor å bare fjerne hele variabelen.

titanic <- titanic |>   
 select(-Cabin)

Variabelen Age mangler 177 verdier, noe som kan være problematisk ettersom vi tenker at alder kan være viktig en viktig indikator for om noen overlevde eller ikke. Siden datasettet ikke er så stort velger vi å forsøke å fylle de manglende verdiene med gjennomsnittsalderen til passasjerer som deler tittel, og reiser med samme klasse (Pclass). Ved å dele inn etter både tittel og klasse forsøker vi å unngå at for mange observasjoner får den samme alderen, noe som kan bidra til å påvirke presisjonen til modellene.

Inspirasjonen er hentet fra artikkelen til *Allohvk*

#Lager lister til å fylle inn verdiene  
master.mean <- list()   
mr.mean <- list()  
mrs.mean <- list()  
miss.mean <- list()  
  
  
#Setter opp en for-loop til å finne gjennomsnittet for alle med en gitt tittel, i en gitt klasse  
for (i in 1:3){  
 for (j in c("Master", "Mr.", "Mrs.", "Miss")) {  
 if (j == "Master"){  
 master.mean[[i]] <- titanic |>   
 filter(grepl("Master.", Name, fixed = TRUE), Pclass == i, Sex == "male") |>   
 summarise(mean\_age = mean(Age, na.rm = TRUE)) |>   
 pull(mean\_age)  
 } else if(j == "Mr."){  
 mr.mean[[i]] <- titanic |>   
 filter(grepl("Mr.", Name, fixed = TRUE), Pclass == i, Sex == "male") |>   
 summarise(mean\_age = mean(Age, na.rm = TRUE)) |>   
 pull(mean\_age)  
 } else if (j == "Mrs.") {  
 mrs.mean[[i]] <- titanic |>   
 filter(grepl("Mrs.", Name, fixed = TRUE), Pclass == i, Sex == "female") |>   
 summarise(mean\_age = mean(Age, na.rm = TRUE)) |>   
 pull(mean\_age)  
 } else if(j == "Miss") {  
 miss.mean[[i]] <- titanic |>   
 filter(grepl("Miss.", Name, fixed = TRUE), Pclass == i, Sex == "female") |>   
 summarise(mean\_age = mean(Age, na.rm = TRUE)) |>   
 pull(mean\_age)  
 }  
 }  
}  
  
  
#Fyller inn de manglende verdiene  
for (i in 1:3){  
 titanic <- titanic |>   
 mutate(Age = ifelse(grepl("Mr.", Name, fixed = T) & Pclass == i & is.na(Age), mr.mean[[i]], Age)) |>   
 mutate(Age = ifelse(grepl("Miss", Name, fixed = T) & Pclass == i & is.na(Age), miss.mean[[i]], Age)) |>   
 mutate(Age = ifelse(grepl("Mrs.", Name, fixed = T) & Pclass == i & is.na(Age), mrs.mean[[i]], Age)) |>   
 mutate(Age = ifelse(grepl("Master", Name, fixed = T) & Pclass == i & is.na(Age), master.mean[[i]], Age))  
}

Vi fyller inn de manglende verdiene ved å ta utgangspunkt i titlene - Master, Mr. , Mrs. , Miss. - og finner gjennomsnittsalderen for klassen de tilhører, og fyller dette inn i datasettet. Ettersom det eksisterer flere titler som vi ikke dekket sjekker vi om vi finner flere manglende verdier, og forsøker samme metode for å sette inn verdiene.

#Sjekker om det er fler manglende verider  
titanic |>   
 filter(is.na(Age))

# A tibble: 1 × 11  
 PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare  
 <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl>  
1 767 0 1 Brewe, Dr. A… male NA 0 0 112379 39.6  
# ℹ 1 more variable: Embarked <chr>

#Finner gjennomsnittet for tittel gitt klasse  
mean.dr.p1 <- titanic |>   
 filter(grepl("Dr.", Name, fixed = TRUE) & Pclass == 1) |>   
 summarise(round(mean(Age, na.rm = TRUE))) |>   
 pull()  
  
#Setter inn for verdien  
titanic <- titanic |>   
 mutate(Age = ifelse(is.na(Age), mean.dr.p1, Age))

Fra utskriften til skimr funksjonen ser vi at det nå kun mangler 2 verdier for variabelen Embarked. Vi undersøker disse nærmere:

titanic |>   
 filter(is.na(Embarked))

# A tibble: 2 × 11  
 PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare  
 <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl>  
1 62 1 1 Icard, Miss.… fema… 38 0 0 113572 80  
2 830 1 1 Stone, Mrs. … fema… 62 0 0 113572 80  
# ℹ 1 more variable: Embarked <chr>

Vi ser at de manglende verdiene er for to kvinner, som begge overlevde, og som reiste med første klasse. For å opprettholde variasjonen i datasettet, og bevare flest mulig av observasjonene vi ønsker å predikere, vil vi forsøke å beholde observasjonene ved å fylle inn en verdi for Embarked.

titanic |>   
 filter(Sex == "female" & Pclass == 1) |>   
 group\_by(Embarked, Pclass) |>  
 summarise(mean\_survived = mean(Survived), antall = n())

# A tibble: 4 × 4  
# Groups: Embarked [4]  
 Embarked Pclass mean\_survived antall  
 <chr> <dbl> <dbl> <int>  
1 C 1 0.977 43  
2 Q 1 1 1  
3 S 1 0.958 48  
4 <NA> 1 1 2

Det vi kan tolke fra utskriften er at de fleste kvinnene som reiste med førsteklasse overlevde, uavhengig av hvor de gikk ombord. Det vil derfor være rimelig å fylle de manglende verdiene med et tilfeldig trekk mellom “C” og “S”, ettersom disse var de vanligste stedene å gå ombord.

tilfeldig\_embarked <- sample(c("C", "S"), size = 1)  
  
titanic <- titanic |>   
 mutate(Embarked = ifelse(is.na(Embarked),  
 yes = tilfeldig\_embarked,  
 no = Embarked))  
  
paste("Det er",sum(is.na(titanic)), "NAs datasettet")

[1] "Det er 0 NAs datasettet"

### Visualisering av data

Når vi nå har bearbeidet dataene ønsker vi å utforske de ytterligere, for å få et bedre inntrykk av sammenhengene i dataene.

titanic |>   
 group\_by(Survived) |>   
 summarise(Antall = n())

# A tibble: 2 × 2  
 Survived Antall  
 <dbl> <int>  
1 0 549  
2 1 342

Vi ser her at det kun er 342 passasjerer som overlevde, og at det var 549 passasjerer som omkom. Vi ønsker først å se på hvordan antall overlevende fordeler seg på klassen de reiste med. Antallet overlevende ser ut til å være ganske likt mellom klassene. Å se på andelen overlevende fordelt på klasse kan derfor gi et bedre inntrykk.

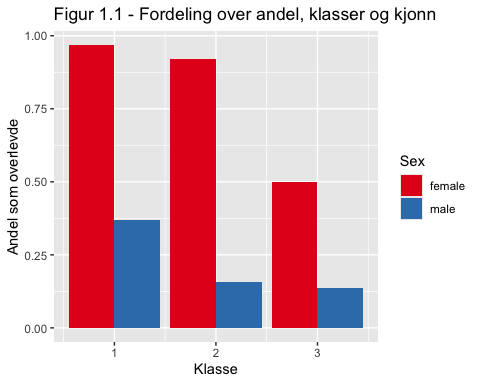
titanic |>   
 group\_by(Pclass) |>   
 summarise(Overlevelsesandel = sum(Survived)/n())

# A tibble: 3 × 2  
 Pclass Overlevelsesandel  
 <dbl> <dbl>  
1 1 0.630  
2 2 0.473  
3 3 0.242

#### Kjønn og overlevlese

Vi ser av utskriften at andelen som overlevde er betydelig høyere for klasse 2 og 1 enn for klasse 2. Videre ønsker vi å undersøke betydningen til kjønn. Vi finner Overlevelsesandelen, fordelt på kjønn, fordelt på klasse.

library(tidyverse)  
  
titanic |>   
 group\_by(Pclass,Sex) |>   
 summarise(Overlevelsesandel = sum(Survived)/n()) |>   
 ggplot(aes(x = Pclass, y = Overlevelsesandel, fill = Sex)) +  
 geom\_col(position = position\_dodge()) +  
 ylab("Andel som overlevde") +  
 xlab("Klasse") +  
 ggtitle("Figur 1.1 - Fordeling over andel, klasser og kjonn") +  
 scale\_fill\_brewer(palette = "Set1")

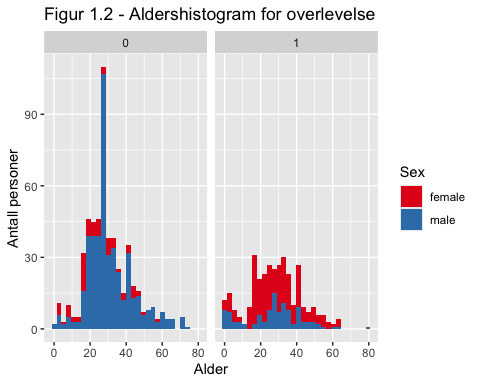


Av diagrammet kan en se at det er en markant forskjell på overlevelsesandelen for kvinner og menn, for alle klasser. En kan også se at observasjonen fra utskriften også holder her, men at effekten er større for menn enn for kvinner. Vi ser at andelen for kvinner som overlevde i tredje klasse er større enn andelen for menn som overlevde i første klasse. Dette gir mening med tanke på at kvinner og barn var prioritert - “Kvinner og barn først”. Vi ønsker derfor å se på hvordan alder, og kjønn fordeler seg på overlevelse.

Vi plotter derfor et histogram som viser aldersfordelingen, og kjønnsfordelingen for døde og overlevende.

titanic |>   
 ggplot(aes(x = Age, fill = Sex)) +  
 geom\_histogram() +  
 facet\_wrap(vars(Survived)) +  
 scale\_fill\_brewer(palette = "Set1") +  
 ggtitle("Figur 1.2 - Aldershistogram for overlevelse") +  
 xlab("Alder") +  
 ylab("Antall personer")

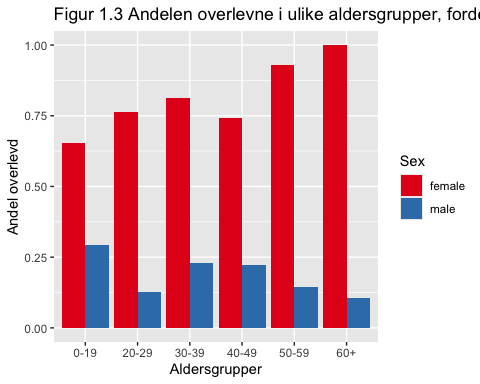
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



Histogrammet bekrefter de tidligere observasjonene. Vi ser at histogrammet for overlevende har en tyngre hale for Alder < 18, noe som indikerer at det var en større overlevelse blant barn og unge. Dette styrker hypotesen om at kjønn og alder er tett korrelert med sannsynligheten for å overleve.

For å vise denne effekten tydeligere deler vi inn i aldersgrupper

titanic |>   
 mutate(  
 Agegroup = cut(  
 Age,   
 breaks = c(-Inf,20, 30, 40, 50, 60, Inf),   
 labels = c("0-19", "20-29", "30-39", "40-49", "50-59", "60+"))) |>   
 group\_by(Agegroup,Sex) |>   
 summarise(Andel = sum(Survived)/n())|>  
 ggplot(aes(x = Agegroup, y = Andel, fill = Sex)) +  
 geom\_col(position = "dodge") +  
 scale\_fill\_brewer(palette = "Set1") +   
 xlab("Aldersgrupper") +  
 ylab("Andel overlevd") +  
 ggtitle("Figur 1.3 Andelen overlevne i ulike aldersgrupper, fordelt på kjønn")

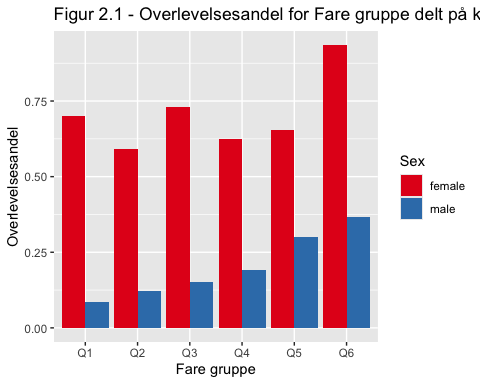


Vi ser her at tendensen fra tidligere - at kvinner i større grad overlever en menn - består. Vi ser at alder har en svak sammenheng med overlevelse, men at denne sammenhengen er ulik for kvinner og menn. For kvinner er det en stigende sammenheng, med unntak av aldersgrupper 41-50, som går litt ned. Grunnen for dette kan være at kvinner mellom 0 og 20, enten selv var barn - og dermed ble prioritert - eller at de var unge barnløse kvinner som hadde prioritet etter barn, og kvinner med barn slik at de trekker overlevelsesandelen ned. Kvinner i gruppen 41-50 kan , pågrunn av alder, antas å reise uten barn, og ble derfor nedprioritert til fordel for barn, kvinner med barn og eldre kvinner. For menn ser man at den høyeste overlevelsesraten er for menn mellom 0-20, noe som er plausibelt ettersom barn ble prioritet. Den laveste overlevelsesraten er for menn i alderen 20-31, videre kan en se at det ikke er særlig stor sammenheng mellom alder og overlevelse, men at det er en synkende tendens for aldre over 30.

#### Fare, Embarked og Overlevelse

Videre ønsker vi å undersøke sammenhengen mellom prisen på billett, og overlevelse. Deler derfor variabelen fare inn i 6 grupper, og finner overlevelsesandelen per gruppe.

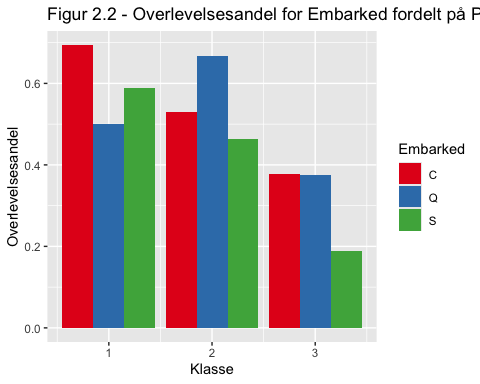
titanic |>   
 filter(Fare < 200) |>   
 mutate(Fare\_group = ntile(Fare, 6)) |>  
 mutate(Fare\_group = factor(Fare\_group, labels = c("Q1", "Q2", "Q3", "Q4", "Q5", "Q6"))) |>   
 group\_by(Fare\_group, Sex) |>   
 summarise(andel = sum(Survived)/n()) |>   
 ggplot(aes(x = Fare\_group, y = andel, fill = Sex)) +   
 geom\_col(position = "dodge") +  
 scale\_fill\_brewer(palette = "Set1") +  
 xlab("Fare gruppe") +  
 ylab("Overlevelsesandel") +  
 ggtitle("Figur 2.1 - Overlevelsesandel for Fare gruppe delt på kjønn")



Utvalget har noen uteliggende observasjoner som vi har fjernet for å få en tydeligere visualisering, vi deler så utvalget inn i seks ulike grupper, og viser overlevelsesratene for personene i de ulike Fare gruppene, og deler inn i kjønn. Vi ser av denne visualiseringen at det ikke er noe tydelig sammenheng for kvinner hva gjelder hvilken Fare gruppe de var i. For menn ser vi at det er en jevn økning i overlevelsesandelen når faregruppen øker.

Til sist ønsker vi å se på hvordan overlevelsesandelen for passasjerene på de ulike ombordstigningene er.

titanic |>   
 group\_by(Embarked, Pclass) |>   
 summarise(andel\_overlevde = sum(Survived)/ n()) |>   
 ggplot(aes(x = Pclass, y = andel\_overlevde, fill = Embarked )) +  
 geom\_col(position = position\_dodge()) +  
 scale\_fill\_brewer(palette = "Set1") +  
 ggtitle("Figur 2.2 - Overlevelsesandel for Embarked fordelt på Pclass") +  
 xlab("Klasse") +  
 ylab("Overlevelsesandel")



Vi ser her at det ikke er noe tydelig mønster for overlevelsesandel, og hvor passasjerene gikk ombord. Dette er også rimelig med tanke på at hvor en gikk om bord trolig ikke påvirker ens sjanse til å overleve.

#### Konklusjon fra datavisualisering

Vi kan se av grafene at det i hovedsak er kjønn og hvilken klasse en tilhører som har sterkest sammenheng med overlevelsen. Vi ser at hvor mye biletten kostet har noe å si for overlevelse, men denne sammengen er sterkest for menn, og at hvor en gikk på ikke har så mye å si. Alder har mest å si for menn.

## Nye variabler

Vi ser at den kan være hensiktsmessig å formalisere noen av variablene, og lage nye variabler. Dette kan forbedrer modellens ytelse, gjør resultatene lettere å forstå og kan gi ny innsikt.

Vi velger å lage en dummy variabel for om man er mindreåring eller ikke, en for hvor stor familie personen reiser med, og antall personer per billett. Inspirasjon hentet (Donges, 2018).

titanic <- titanic |>   
 add\_count(Ticket, name = "Person\_per\_ticket") |>   
 mutate(Minor = ifelse(Age < 18, 1, 0),  
 Family\_size = SibSp+Parch,  
 Alone = ifelse(Family\_size == 0, 1, 0)) |>   
 select(-c(Name, Ticket, SibSp, Parch, PassengerId)) |>   
 mutate(Pclass = as.factor(Pclass),  
 Alone = as.factor(Alone),  
 Survived = as.factor(Survived),  
 Minor = as.factor(Minor),  
 Sex = as.factor(Sex),  
 Embarked = as.factor(Embarked))

# Maskinlæring

Før vi tilpasser modellene vi ønsker å bruke, deler vi datasettet inn i trening og test sett. Dataene deles inn i en 80/20 fordeling hvor variabelen Survived skal være jevnt fordelt.

set.seed(3170)  
titanic\_split<- initial\_split(titanic, prop = .8, strata = Survived)  
titanic\_train <- training(titanic\_split)  
titanic\_test <- testing(titanic\_split)  
titanic\_split

<Training/Testing/Total>  
<712/179/891>

Lager en recepie:

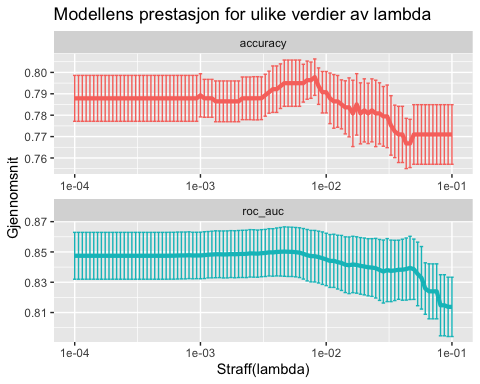
titanic\_recipe <-   
 recipe(Survived ~ ., data = titanic\_train) |>  
 step\_dummy(Sex, Embarked, Alone, Pclass, Minor)

## LASSO

Vi setter opp lasso modellen og tuner den. Plottet viser hvordan accuracy og roc\_auc utvikler seg for ulike verdier av lambda.

lasso\_model <-   
 logistic\_reg(penalty = tune(), mixture = 1) |>   
 set\_engine("glmnet") |>   
 set\_mode("classification")  
  
  
wflow\_lasso <-   
 workflow() |>   
 add\_recipe(titanic\_recipe) |>   
 add\_model(lasso\_model)  
  
folds <- vfold\_cv(titanic\_train, 5, strata = Survived)  
  
penalty\_grid <- grid\_regular(penalty(range = c(-4, -1)), levels = 100)  
  
doParallel::registerDoParallel() # Parallelprogramering for raskere tune  
  
lasso\_tune <-   
 wflow\_lasso |>   
 tune\_grid(resamples = folds, grid = penalty\_grid)  
  
lasso\_tune |>  
 collect\_metrics() |>   
 filter(.metric != "brier\_class") |>   
 ggplot(aes(penalty, mean, color = .metric)) +  
 geom\_errorbar(aes(  
 ymin = mean - std\_err,  
 ymax = mean + std\_err  
 )) +  
 geom\_line(size = 1.5) +  
 facet\_wrap(~.metric, scales = "free", nrow = 2) +  
 scale\_x\_log10() +  
 theme(legend.position = "none") +  
 ggtitle("Modellens prestasjon for ulike verdier av lambda")+  
 xlab("Straff(lambda)") +  
 ylab("Gjennomsnit")

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.



Bruker den beste roc\_auc verdien i modellen og tilpasser modellen med denne verdien.

beste\_lamda <- select\_best(lasso\_tune, metric = "roc\_auc")  
beste\_lamda

# A tibble: 1 × 2  
 penalty .config   
 <dbl> <chr>   
1 0.00464 Preprocessor1\_Model056

lasso\_fit <- finalize\_workflow(wflow\_lasso, beste\_lamda) |>   
 fit(data = titanic\_train)

Kjører prediksjoner med de nye parameterne

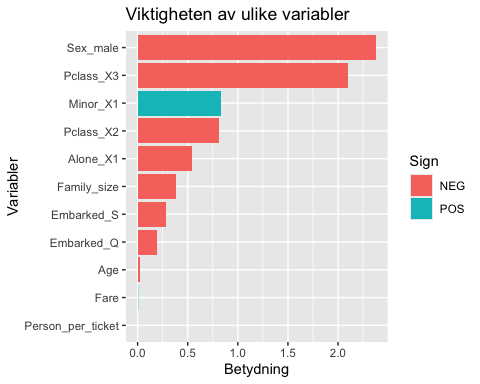
predict(lasso\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test) |>   
 accuracy(truth = Survived, estimate = .pred\_class)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 accuracy binary 0.832

auc\_lasso <-   
 predict(lasso\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_auc(Survived, .pred\_0) |>   
 pull(.estimate)  
  
lasso\_roc <-   
 predict(lasso\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_curve(Survived, .pred\_0)

Finner de hvilke variabler som er viktigst for modellen.

lasso\_fit |>   
 extract\_fit\_parsnip() |>   
 vip::vi() |>   
 ggplot(aes(y = reorder(Variable, Importance),   
 x = Importance, fill = Sign)) +  
 geom\_col() +  
 theme() +   
 ggtitle("Viktigheten av ulike variabler") +  
 xlab("Betydning") +  
 ylab("Variabler")



Konstruerer en forvirringsmatrise med prediksjonene fra modellen

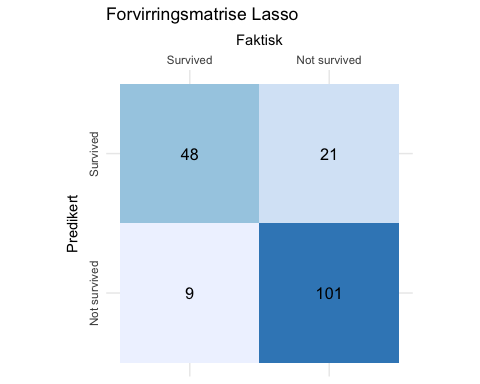
library(cvms)  
  
prediksjoner\_lasso <- predict(lasso\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test)  
  
forvirringsmatrise\_lasso <- prediksjoner\_lasso |>   
 conf\_mat(Survived, .pred\_class)  
  
riktigmatrise\_lasso <- as.tibble(forvirringsmatrise\_lasso$table) |>   
 mutate(Prediction = ifelse(Prediction == 0, "Not survived", "Survived")) |>   
 mutate(Truth = ifelse(Truth == 0, "Not survived", "Survived"))

Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
ℹ Please use `as\_tibble()` instead.  
ℹ The signature and semantics have changed, see `?as\_tibble`.

plot\_confusion\_matrix(riktigmatrise\_lasso, "Prediction", "Truth", "n",   
 add\_normalized = F,  
 add\_row\_percentages = F,  
 add\_col\_percentages = F) +  
 xlab("Faktisk") +  
 ylab("Predikert") +  
 ggtitle("Forvirringsmatrise Lasso")

Warning in plot\_confusion\_matrix(riktigmatrise\_lasso, "Prediction", "Truth", :  
'ggimage' is missing. Will not plot arrows and zero-shading.

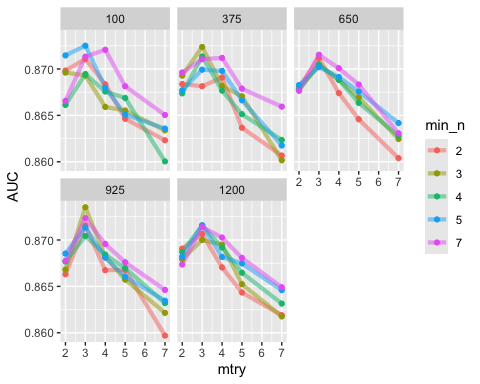
Warning in plot\_confusion\_matrix(riktigmatrise\_lasso, "Prediction", "Truth", :  
'rsvg' is missing. Will not plot arrows and zero-shading.



## Random forest

Definerer modellen, lager en workflow med samme recipe, definerer en grid og tilpasser hyperparametere.

set.seed(3170)  
  
rf\_mod <- rand\_forest(  
 mtry = tune(),  
 trees = tune(),  
 min\_n = tune()  
) |>   
 set\_engine("ranger", importance = "impurity") |>   
 # For å kunne bruke Vip pakken  
 set\_mode("classification")  
  
  
  
rf\_wflow <- workflow() |>   
 add\_recipe(titanic\_recipe) |>   
 add\_model(rf\_mod)  
  
  
rf\_grid <- grid\_regular(  
 mtry(range = c(2, 7)),  
 min\_n(range = c(2,7)),  
 trees(range = c(100,1200)),  
 levels = 5  
)  
  
  
doParallel::registerDoParallel()  
  
tune\_rf <- tune\_grid(  
 rf\_wflow,  
 resamples = folds,  
 grid = rf\_grid  
)  
  
tune\_rf |>   
 collect\_metrics() |>   
 filter(.metric == "roc\_auc") |>   
 mutate(min\_n = factor(min\_n)) |>   
 ggplot(aes(mtry, mean, color = min\_n)) +  
 geom\_line(alpha = 0.5, size = 1.5) +  
 geom\_point() +  
 labs(y = "AUC") +  
 facet\_wrap(vars(trees))



Vi ser her hvilke kombinasjoner som gir høyest treffsikkerhet.

Bruker resultatene fra tilpassingen til å trene modellen, og kjøre prediksjoner med den trente modellen.

set.seed(3170)  
beste\_mtry\_min\_tree <- select\_best(tune\_rf, metric = "roc\_auc")  
  
rf\_fit <- finalize\_workflow(rf\_wflow, beste\_mtry\_min\_tree) |>   
 fit(titanic\_train)  
  
  
pred.rf <- predict(rf\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test) |>   
 accuracy(Survived, .pred\_class)  
pred.rf

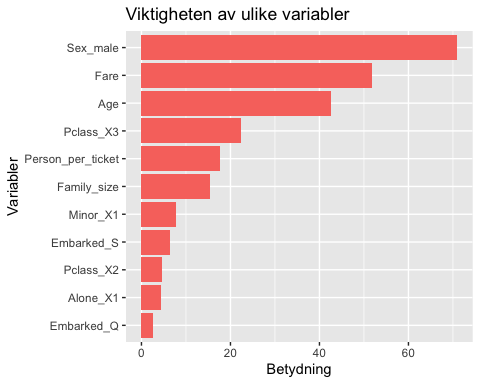
# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 accuracy binary 0.844

Finner AUC og ROC for prediksjonen

auc\_rf <-   
 predict(rf\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_auc(Survived, .pred\_0) |>   
 pull(.estimate)  
  
rf\_roc <- predict(rf\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_curve(Survived, .pred\_0)

Finner hvilke variabler som er viktige for prediksjonen

rf\_fit |>   
 extract\_fit\_parsnip() |>   
 vip::vi() |>   
 ggplot(aes(y = reorder(Variable, Importance), x = Importance, fill = "#F8766D")) +  
 geom\_col() +  
 theme(legend.position = "none") +   
 ggtitle("Viktigheten av ulike variabler") +  
 xlab("Betydning") +  
 ylab("Variabler")



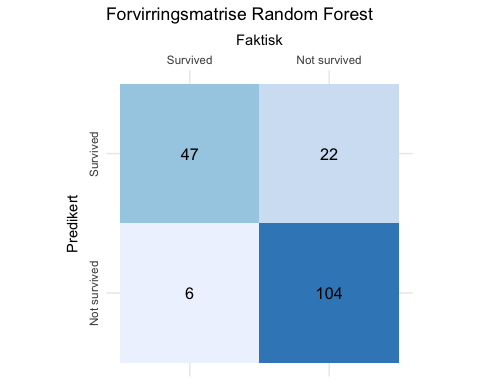
Denne modellen trekker frem - i likhet med LASSO - at kjønn er viktigst for prediksjonen. I kontrast til LASSO vekter random forest modellen variabler som Fare og Age høyt, og bryr seg mindre om klassen til passasjeren.

Konstruerer så en forvirringsmatrise for Random Forest modellen

prediksjoner\_rf <- predict(rf\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test)  
  
forvirringsmatrise\_rf <- prediksjoner\_rf |>   
 conf\_mat(Survived, .pred\_class)  
  
riktigmatrise\_rf <- as.tibble(forvirringsmatrise\_rf$table) |>   
 mutate(Prediction = ifelse(Prediction == 0, "Not survived", "Survived")) |>   
 mutate(Truth = ifelse(Truth == 0, "Not survived", "Survived"))  
  
plot\_confusion\_matrix(riktigmatrise\_rf, "Prediction", "Truth", "n",   
 add\_normalized = F,  
 add\_row\_percentages = F,  
 add\_col\_percentages = F) +  
 xlab("Faktisk") +  
 ylab("Predikert") +  
 ggtitle("Forvirringsmatrise Random Forest")

Warning in plot\_confusion\_matrix(riktigmatrise\_rf, "Prediction", "Truth", :  
'ggimage' is missing. Will not plot arrows and zero-shading.

Warning in plot\_confusion\_matrix(riktigmatrise\_rf, "Prediction", "Truth", :  
'rsvg' is missing. Will not plot arrows and zero-shading.



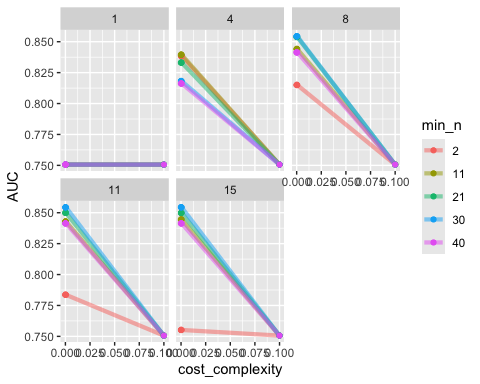
## Decision Tree

Følger de samme prosedyrene som ved tidligere modeller

decision\_model <- decision\_tree(  
 cost\_complexity = tune(),  
 tree\_depth = tune(),  
 min\_n = tune()) |>   
 set\_engine("rpart") |>   
 set\_mode("classification")  
  
  
  
decision\_wflow <- workflow() |>   
 add\_recipe(titanic\_recipe) |>   
 add\_model(decision\_model)  
  
decision\_tree\_params <- parameters(decision\_model) |>   
 update(  
 cost\_complexity = cost\_complexity(),  
 tree\_depth = tree\_depth(),  
 min\_n = min\_n()  
 )

Warning: `parameters.model\_spec()` was deprecated in tune 0.1.6.9003.  
ℹ Please use `hardhat::extract\_parameter\_set\_dials()` instead.

decision\_grid <- grid\_regular(  
 decision\_tree\_params,  
 levels = 5  
)  
  
doParallel::registerDoParallel()  
  
tune\_decision <- tune\_grid(  
 decision\_wflow,  
 resamples = folds,  
 grid = decision\_grid,  
)  
  
tune\_decision |>   
 collect\_metrics() |>   
 filter(.metric == "roc\_auc") |>   
 mutate(min\_n = factor(min\_n)) |>   
 ggplot(aes(cost\_complexity, mean, color = min\_n)) +  
 geom\_line(alpha = 0.5, size = 1.5) +  
 geom\_point() +  
 labs(y = "AUC") +  
 facet\_wrap(vars(tree\_depth))



Lar tidymodels ta seg av ranges på ulike parameterer. Noe som vi ikke gjorde for random forest eller lasso.

best\_cost\_depth\_min <- select\_best(tune\_decision, metric = "roc\_auc")  
best\_cost\_depth\_min

# A tibble: 1 × 4  
 cost\_complexity tree\_depth min\_n .config   
 <dbl> <int> <int> <chr>   
1 0.0000000001 8 30 Preprocessor1\_Model086

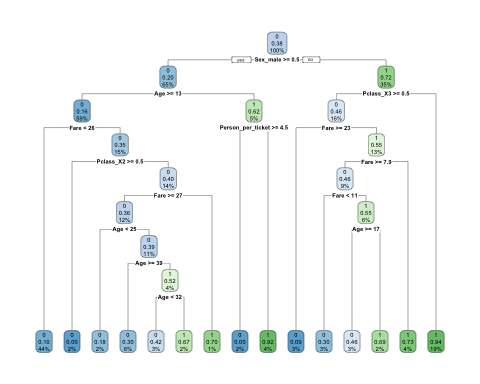
decision\_fit <- finalize\_workflow(decision\_wflow, best\_cost\_depth\_min) |>   
 fit(titanic\_train)  
  
  
predict(decision\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test) |>   
 accuracy(Survived, .pred\_class)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 accuracy binary 0.827

Dårligste Accuarcy så langt

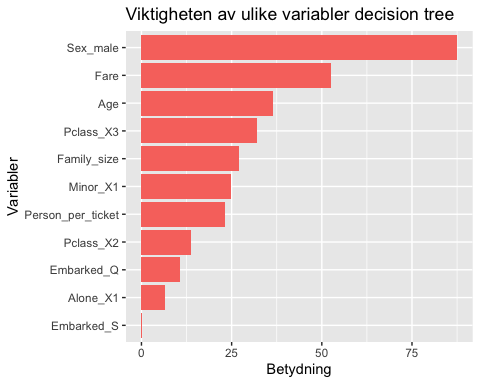
auc\_decision <-   
 predict(decision\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_auc(Survived, .pred\_0) |>   
 pull(.estimate)  
  
decision\_roc <- predict(decision\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_curve(Survived, .pred\_0)

library(rpart)  
library(rpart.plot)  
  
decision\_final\_fit <- finalize\_workflow(decision\_wflow, best\_cost\_depth\_min) |>   
 last\_fit(titanic\_split)  
  
final\_tree <- extract\_workflow(decision\_final\_fit)  
  
final\_tree |>   
 extract\_fit\_engine() |>   
 rpart.plot(roundint = FALSE)



For å bruke funkjsonen til rpart, må vi fitte på en annen måte, vi kunne ha brukt denne metodene med last\_fit gjennom hele, men vi syntes det var greiere å kun jobbe med fit, isteden for å kjøre last\_fit

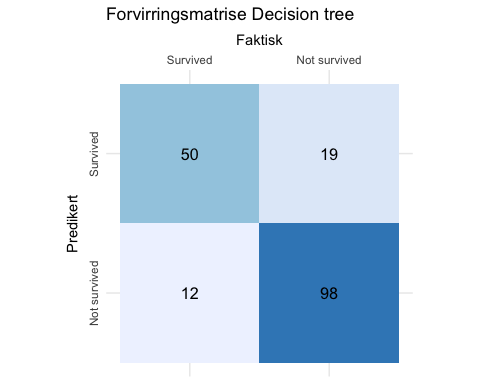
decision\_fit |>   
 extract\_fit\_parsnip() |>   
 vip::vi() |>   
 ggplot(aes(y = reorder(Variable, Importance), x = Importance, fill = "#F8766D")) +  
 geom\_col() +  
 theme(legend.position = "none") +   
 ggtitle("Viktigheten av ulike variabler decision tree") +  
 xlab("Betydning") +  
 ylab("Variabler")



prediksjoner\_decision <- predict(decision\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test)  
  
forvirringsmatrise\_decision <- prediksjoner\_decision |>   
 conf\_mat(Survived, .pred\_class)  
  
riktigmatrise\_decision <- as.tibble(forvirringsmatrise\_decision$table) |>   
 mutate(Prediction = ifelse(Prediction == 0, "Not survived", "Survived")) |>   
 mutate(Truth = ifelse(Truth == 0, "Not survived", "Survived"))  
  
plot\_confusion\_matrix(riktigmatrise\_decision, "Prediction", "Truth", "n",   
 add\_normalized = F,  
 add\_row\_percentages = F,  
 add\_col\_percentages = F) +  
 xlab("Faktisk") +  
 ylab("Predikert") +  
 ggtitle("Forvirringsmatrise Decision tree")

Warning in plot\_confusion\_matrix(riktigmatrise\_decision, "Prediction", "Truth",  
: 'ggimage' is missing. Will not plot arrows and zero-shading.

Warning in plot\_confusion\_matrix(riktigmatrise\_decision, "Prediction", "Truth",  
: 'rsvg' is missing. Will not plot arrows and zero-shading.



# Extreme Gradient Boosting Tree

xg\_model <- boost\_tree(  
 trees = 1000,  
 tree\_depth = tune(),  
 learn\_rate = tune(),  
 mtry = tune(),  
 min\_n = tune(),  
 loss\_reduction = tune(),  
 sample\_size = tune()  
 )|>   
 set\_engine("xgboost") |>   
 set\_mode("classification")  
  
  
xg\_wflow <- workflow() |>   
 add\_recipe(titanic\_recipe) |>   
 add\_model(xg\_model)  
  
xg\_grid <- grid\_latin\_hypercube(  
 tree\_depth(),  
 learn\_rate(),  
 finalize(mtry(), titanic\_train),  
 min\_n(),  
 loss\_reduction(),  
 sample\_size = sample\_prop(),  
 size = 50  
)

Warning: `grid\_latin\_hypercube()` was deprecated in dials 1.3.0.  
ℹ Please use `grid\_space\_filling()` instead.

doParallel::registerDoParallel()  
  
tune\_xg <- tune\_grid(  
 xg\_wflow,  
 resamples = folds,  
 grid = xg\_grid,  
)  
  
best\_param\_xg <- select\_best(tune\_xg, metric = "roc\_auc")  
best\_param\_xg

# A tibble: 1 × 7  
 mtry min\_n tree\_depth learn\_rate loss\_reduction sample\_size .config   
 <int> <int> <int> <dbl> <dbl> <dbl> <chr>   
1 6 9 11 0.0991 0.00000154 0.483 Preprocessor1\_Mo…

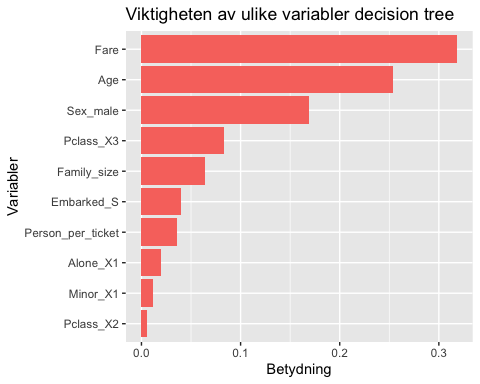
grid: <https://juliasilge.com/blog/xgboost-tune-volleyball/,> blir litt vanskelig å plotte

xg\_fit <- finalize\_workflow(xg\_wflow, best\_param\_xg) |>   
 fit(titanic\_train)  
  
predict(xg\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test) |>   
 accuracy(Survived, .pred\_class)

# A tibble: 1 × 3  
 .metric .estimator .estimate  
 <chr> <chr> <dbl>  
1 accuracy binary 0.860

auc\_xg <-   
 predict(xg\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_auc(Survived, .pred\_0) |>   
 pull(.estimate)  
  
xg\_roc <- predict(xg\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_curve(Survived, .pred\_0)

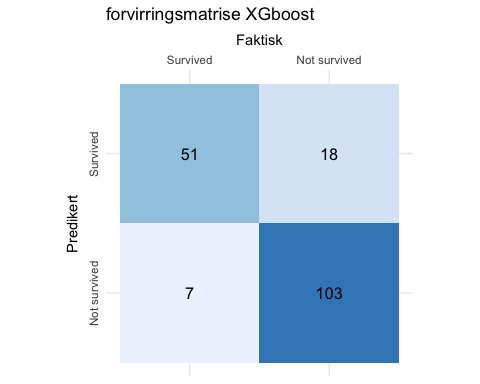
xg\_fit |>   
 extract\_fit\_parsnip() |>   
 vip::vi() |>   
 ggplot(aes(y = reorder(Variable, Importance), x = Importance, fill = "#F8766D")) +  
 geom\_col() +  
 theme(legend.position = "none") +   
 ggtitle("Viktigheten av ulike variabler decision tree") +  
 xlab("Betydning") +  
 ylab("Variabler")



prediksjoner\_xg <- predict(xg\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test)  
  
forvirringsmatrise\_xg <- prediksjoner\_xg |>   
 conf\_mat(Survived, .pred\_class)  
  
riktigmatrise\_xg <- as.tibble(forvirringsmatrise\_xg$table) |>   
 mutate(Prediction = ifelse(Prediction == 0, "Not survived", "Survived")) |>   
 mutate(Truth = ifelse(Truth == 0, "Not survived", "Survived"))  
  
plot\_confusion\_matrix(riktigmatrise\_xg, "Prediction", "Truth", "n",   
 add\_normalized = F,  
 add\_row\_percentages = F,  
 add\_col\_percentages = F) +  
 xlab("Faktisk") +  
 ylab("Predikert") +  
 ggtitle("forvirringsmatrise XGboost")

Warning in plot\_confusion\_matrix(riktigmatrise\_xg, "Prediction", "Truth", :  
'ggimage' is missing. Will not plot arrows and zero-shading.

Warning in plot\_confusion\_matrix(riktigmatrise\_xg, "Prediction", "Truth", :  
'rsvg' is missing. Will not plot arrows and zero-shading.



# Multi Layer Percepton

Starter med å definere modell, og lage en workflow.

mlp.model <- mlp(hidden\_units = tune(), penalty = tune(), epochs = tune()) |>   
 set\_engine("nnet") |>   
 set\_mode("classification")  
  
mlp\_wflow <- workflow() |>   
 add\_model(mlp.model) |>   
 add\_recipe(titanic\_recipe)

Henter så ut hyperparameterne og implempenterer de i en grid ved hjelp av grid\_latin\_hypercube med 50 nivåer

mlp\_params <- extract\_parameter\_set\_dials(mlp\_wflow)  
meterics <- metric\_set(roc\_auc, accuracy, brier\_class)  
  
grid <- grid\_latin\_hypercube(  
 mlp\_params,  
 size = 50  
)  
  
doParallel::registerDoParallel()  
  
mlp\_tune <- tune\_grid(  
 mlp\_wflow,   
 resamples = folds,   
 grid = grid,  
 metrics = meterics,  
 control = control\_grid(save\_pred = TRUE)  
)

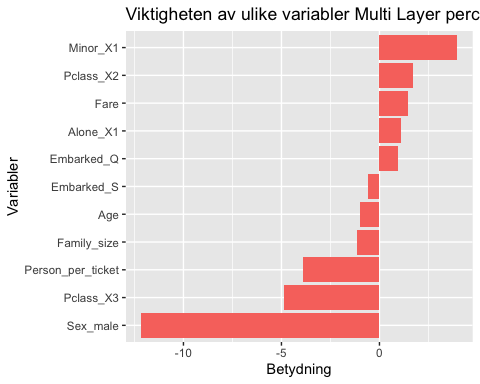
Henter deretter ut de beste parameterne, og tilpasser modellen

logistic\_param.reg <- select\_best(mlp\_tune, metric = "roc\_auc") |>  
 select(-.config)  
  
fn.mlp\_wflow <- mlp\_wflow |>   
 finalize\_workflow(logistic\_param.reg)  
  
fn.mlp\_fit <- fn.mlp\_wflow |>   
 fit(titanic\_train)

Kjører prediksjoner, og plotter ROC-AUC.

auc\_mlp <-   
 predict(fn.mlp\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_auc(Survived, .pred\_0) |>   
 pull(.estimate)  
  
mlp\_roc <- predict(fn.mlp\_fit, titanic\_test, type = "prob") |>   
 bind\_cols(titanic\_test) |>   
 roc\_curve(Survived, .pred\_0)

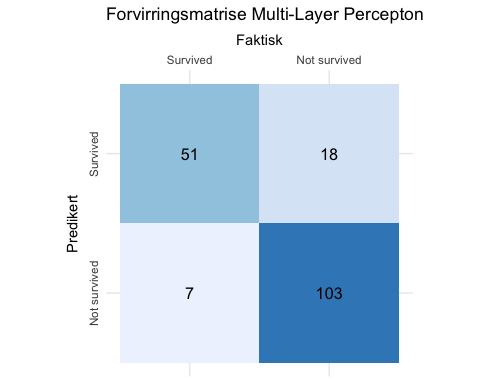
library(NeuralNetTools)  
  
fn.mlp\_fit |>   
 extract\_fit\_parsnip() |>   
 vip::vi() |>   
 ggplot(aes(y = reorder(Variable, Importance), x = Importance, fill = "#F8766D"))+  
 geom\_col() +  
 theme(legend.position = "none") +   
 ggtitle("Viktigheten av ulike variabler Multi Layer percepton") +  
 xlab("Betydning") +  
 ylab("Variabler")



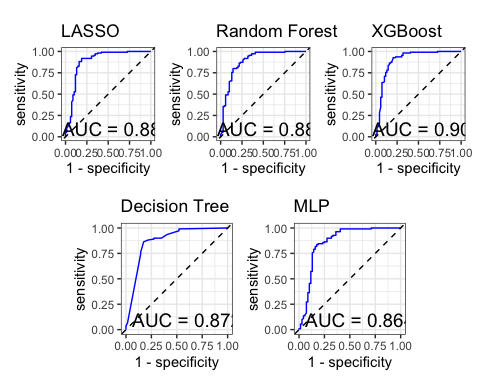
prediksjoner\_mlp <- predict(fn.mlp\_fit, titanic\_test) |>   
 bind\_cols(titanic\_test)  
  
forvirringsmatrise\_xg <- prediksjoner\_xg |>   
 conf\_mat(Survived, .pred\_class)  
  
riktigmatrise\_xg <- as.tibble(forvirringsmatrise\_xg$table) |>   
 mutate(Prediction = ifelse(Prediction == 0, "Not survived", "Survived")) |>   
 mutate(Truth = ifelse(Truth == 0, "Not survived", "Survived"))  
  
plot\_confusion\_matrix(riktigmatrise\_xg, "Prediction", "Truth", "n",   
 add\_normalized = F,  
 add\_row\_percentages = F,  
 add\_col\_percentages = F) +  
 xlab("Faktisk") +  
 ylab("Predikert") +  
 ggtitle("Forvirringsmatrise Multi-Layer Percepton")

Warning in plot\_confusion\_matrix(riktigmatrise\_xg, "Prediction", "Truth", :  
'ggimage' is missing. Will not plot arrows and zero-shading.

Warning in plot\_confusion\_matrix(riktigmatrise\_xg, "Prediction", "Truth", :  
'rsvg' is missing. Will not plot arrows and zero-shading.



library(patchwork)  
  
plot\_roc\_curve <- function(roc\_data, auc\_value, title) {  
 roc\_data |>   
 ggplot(aes(x = 1 - specificity, y = sensitivity)) +   
 geom\_path(col = "blue") +  
 geom\_abline(slope = 1, linetype = "dashed") +  
 coord\_equal() +  
 annotate("text", x = 0.6, y = 0.1,   
 label = paste("AUC =", round(auc\_value, 3)), size = 5) +  
 theme\_bw() +  
 ggtitle(title)  
}  
  
  
lasso\_plot <- plot\_roc\_curve(lasso\_roc, auc\_lasso, "LASSO")  
rf\_plot <- plot\_roc\_curve(rf\_roc, auc\_rf, "Random Forest")  
decision\_plot <- plot\_roc\_curve(decision\_roc, auc\_decision, "Decision Tree")  
xg\_plot <- plot\_roc\_curve(xg\_roc, auc\_xg, "XGBoost")  
mlp\_plot <- plot\_roc\_curve(mlp\_roc, auc\_mlp, "MLP")  
  
  
(lasso\_plot | rf\_plot | xg\_plot) / (decision\_plot | mlp\_plot)



# Evaluering

En ROC-AUC kurve viser forholdet mellom sann positiv og falsk positiv, og gir derfor et godt grunnlag for sammenlikning på tvers av modellene vi har valgt å bruke. (Kuhn, M. & Silge, J., 2023) Desto høyere areal det er imellom kurven og den stiplede linjen - lavere andel av falsk positiv - desto bedre presterer modellen.

Vi ser at alle modellene presterer relativt bra, men at de beste modellene er LASSO-regresjon, Random Forest (RF) og Extreme Gradient Boosting Tree (XGB). Dette har sammenheng med at disse tre modellene bruker liknende metoder. LASSO er i hovedsak en regresjonsmodell som legger til en “penalty” altså en straff for å unngå at modellen overtilpasses og dermed fanger opp tilfeldig støy i dataen. RF lager flere ulike beslutningstrær ved å dele datasettet inn i flere ulike kombinasjoner - derav ensamblemodell. I binære klassifiseringsoppgaver bruker den det resultatet som flest trær gir. XGB kan sies å være en kombinasjon av disse modellene, fordi den bruker liknende metode som RF, ved å lage flere beslutningstrær som den bruker til å optimalisere tapsfunksjonen, i tillegg til å bruke samme regularisering som LASSO (L1) .

De to siste modellene Desicion Tree (DT) og Multi-Layer Percepton (MLP) er to forksjellige modeller som presterer dårligere enn LASSO, RF, og XGB. Desicion Tree lager et beslutningstree av hele treningsettet, og oppretter ulike “grener” for å forsøke å klassifisere en observasjon. Dette gjør at modellen er lett å tolke, og det er enkelt å følge hvilke avveininger den gjør. Problemet med denne metoden er at den lett kan fange opp tilfeldig støy i dataen som gjør den sårbar for overtilpassning. MLP er på en annen side en mye mer kompleks modell som benytter seg av nevrale nettverk for å gjøre prediksjoner, dette gjør det mer komplisert å forstå hvilke faktorer som påvirker hverandre. På grunn av kompleksiteten til modellen krever den at tilpassingen av parameterne er utført nøye og riktig. Kravet til en nøyere tilpasning av parametere er trolig grunnen til at vår modell presterer dårligst, “grid-en” vi har satt opp er relativt simpel, og vi har benyttet oss av samme recipe som for de andre modellene, noe som kan gjøre at dataene er vanskeligere å behandle for modellen, som også gjør den mer upresis.

# Konklusjon

For å konkludere kan vi se at modellene som fungerer best er modeller som klarer å håndtere tilfeldig støy i datasettet godt. De to modellene som presterer høyest er ensemblemodeller, dette kan være fordi beslutningstrærne ikke blir like påvirket av de kontinuerlige variablene som Age og Fare, ettersom grenene i beslutningstrærne trolig vil gruppere disse selv i motsetning til LASSO som i større grad tar hensyn til de eksakte verdiene. XGB modellen presterer best, antagelig fordi denne også implementerer reguleringsmetoder som også brukes i lineære modeller.

Vi har også benyttet oss av ressurser som forelesningsark, forelesningsnotater, og seminaroppgaver fra faget ECON3170

* Kaggle. *Titanic - Machine Learning from Disaster*. Kaggle. Hentet 1. november 2024 fra <https://www.kaggle.com/competitions/titanic>
* Allohvk. *Titanic - Advanced EDA*. Kaggle. Hentet 1. november 2024 fra <https://www.kaggle.com/code/allohvk/titanic-advanced-eda?scriptVersionId=77739368>
* Donges, N. (2018). *Predicting the survival of Titanic passengers*. Towards Data Science. Hentet 4. november 2024 fra <https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8>
* M, Kuhn. & J, Silge. *Tidy modelling with R*