

CHURN ANALYSE



Forfattere:

Line A. Adolph, Maria B. A. Hitz, Maria Cristiana Maxim, Martin Bindner, Abdikadir A. M. H. Omar

1. interne eksamensprojekt

Vejleder: Simon Bjerrum Eilersen

Dato: 9. maj 2025

Antal tegn: xx.xxx

Indholdsfortegnelse

1	Resumé	4
2	Indledning	4
3	Problemstilling	5
4	Problemformulering	5
4.1	Underspørgsmål	6
5	Afgrænsning	6
5.1	AI-chatbots og anvendelse af ChatGPT	6
5.2	Datagrundlag	7
5.3	Modellens omfang og valg af algoritmer	7
5.4	Systemintegration	7
5.5	Juridiske og etiske vurderinger	8
6	Definitioner	8
7	Analyse	9
7.1	Dataforståelse og fordeling	9
7.2	Egenskaber ved virksomheder med høj churn	10
7.3	Feature engineering	13
7.4	Modelperformance	14
7.5	Variable importance og indsigt	15
8	Juridiske og etiske overvejelser	16
8.1	Introduktion	16
8.2	Behandlingsgrundlag og dataminimering	17
8.3	Transparens og oplysningspligt	17
8.4	Pseudonymisering og adgangsforhold	17

8.5	Datasikkerhed og organisatorisk ansvar	18
8.6	Etiske overvejelser og ansvarlig anvendelse	18
8.7	Delkonklusion	19
9	Anbefaling	19
9.1	Medlemspleje prioriteres over for virksomheder uden kontakt, eventdeltagelse eller modtaget hjælp, da disse faktorer er stærkt associeret med churn.	20
9.2	Løbende opdatering af modellen sikres ved at integrere churn-værktøjet i Business Viborgs CRM eller medlemsdatabase, så nye data automatisk indgår i fremtidige analyser. .	20
9.3	Etisk og transparent kommunikation om brugen af data indgår i medlemsdialogen for at styrke tilliden og sikre overholdelse af GDPR.	20
10	Konklusion	20
11	Literaturliste	22
12	Bilagsoversigt	23

1 Resumé

Business Viborg arbejder for at skabe optimale rammer for erhvervslivet i Viborg Kommune og har en ambition om at nå 700 medlemmer i 2025. Medlemsafgang truer imidlertid både organisationens økonomiske fundament og dens rolle som erhvervspolitisk talerør. For at imødekomme denne udfordring er der i dette projekt udviklet en datadrevet prototype, der forudsiger churn og identificerer centrale risikofaktorer på medlemsniveau.

Løsningen kombinerer brugervenlig formidling med avanceret maskinlæring i et R-baseret workflow. Seks modeller blev afprøvet, hvor Random Forest blev valgt som slutmodel på baggrund af gennemsigthed og forklaringskraft. Feature engineering inddrager bl.a. kontaktfrekvens, eventdeltagelse og modtaget erhvervshjælp.

Modellen er operationaliseret i et interaktivt dashboard, som understøtter medlemskonsulenternes opsøgende arbejde. Projektet er udviklet med respekt for GDPR og dataetik og illustrerer, hvordan en lokal medlemsorganisation med begrænsede ressourcer kan anvende data strategisk og ansvarligt til at styrke fastholdelsen og engagementet blandt sine medlemmer.

2 Indledning

Business Viborg er en medlemsorganisation, der arbejder målrettet for at skabe optimale vilkår for erhvervslivet i Viborg Kommune. Med over 600 medlemsvirksomheder udgør organisationen en væsentlig aktør i det lokale erhvervsøkosystem – både som netværksfacilitator, vidensformidler og politisk interessevaretagere. Ifølge chefkonsulent Michael Freundlich er ambitionen at nå 700 medlemmer og en omsætning på 2,9 mio. kr. i 2025.

Men når virksomheder forlader organisationen, reduceres ikke blot indtægtsgrundlaget – også Business Viborgs netværkskapital og politiske legitimitet svækkes. Derfor er det afgørende at få indsigt i, hvilke faktorer der øger risikoen for udmeldelse, og hvordan man kan arbejde proaktivt med medlemsfastholdelse.

I dette projekt udvikles der en datadrevet løsning, der kombinerer teknisk analyse med brugervenlig indsigt og som respekterer både juridiske og etiske rammer. Målet er at styrke medlemskonsulenternes beslutningsgrundlag og understøtte en mere effektiv og målrettet medlemspleje.

3 Problemstilling

Business Viborg er registreret under branchekoden 26104793, og arbejder målrettet for at skabe optimale rammer for erhvervslivet i Viborg Kommune. Som en medlemsorganisation med over 600 virksomheder i ryggen, er relationerne til medlemskredsen helt afgørende, både for at dele viden, styrke netværk og skabe lokal vækst. I forbindelse med præsentationen af Business Viborg udtalte chefkonsulent Michael Freundlich: “Vores mål for 2025 er at nå 700 medlemmer og en omsætning på 2,9 mio. kr.”

Men når virksomheder melder sig ud, mister Business Viborg ikke kun en indtægt, men også værdifulde forbindelser, politisk legitimitet og mulighed for at gøre en forskel for erhvervslivet i området. For at handle proaktivt ønsker Business Viborg at få bedre indsigt i, hvad der driver churn og hvem der er i risikozonen.

Derfor skal der udvikles et datadrevet værktøj, som kombinerer teknisk analyse med brugervenlig indsigt. Et værktøj, der gør det muligt for både medlemskonsulenter og ledelse at træffe kloge beslutninger og handle i tide med respekt for både dataetik og jura.

4 Problemformulering

Hvordan kan Business Viborg analysere og anvende medlemsdata til at udvikle et beslutningsunderstøttende dashboard, der forudsiger churn og forklarer centrale risikofaktorer – baseret på relevante maskinlæringsmetoder og med inddragelse af etiske og juridiske overvejelser?

4.1 Underspørgsmål

Eksplorativ analyse (EDA)

Beskriv hvilke mønstre og karakteristika kendetegner de virksomheder, der forlader Business Viborg?

Modelvalg og performance

Hvordan kan forskellige machine learning-modeller anvendes til at forudsige churn i Business Viborgs kontekst, og hvilke modeller er mest velegnede?

Datavisualisering

Hvordan kan resultater og churn-indsigter formidles via et brugervenligt dashboard, som understøtter daglig opsøgende indsats for medlemskonsulenter og ledelse?

Etik og jura

Hvilke juridiske krav (fx GDPR) og etiske overvejelser bør indgå i udviklingen og brugen af et churn-forudsigelsesværktøj baseret på medlemsdata?

5 Afgrænsning

I udviklingen af en datadrevet churn-model for Business Viborg er det nødvendigt at foretage en række metodiske og praktiske afgrænsninger for at sikre projektets gennemførlighed og fokus. Følgende underafsnit præciserer, hvordan projektets omfang er afgrænset i forhold til teknologisk anvendelse, datagrundlag, modeller, systemintegration og juridiske vurderinger.

5.1 AI-chatbots og anvendelse af ChatGPT

ChatGPT 4.0 har været anvendt som et understøttende værktøj i forbindelse med idéudvikling, sproglig formulering og grammatisk korrektur. Modellen har alene fungeret som et supplement i arbejdet med

tekstbaserede opgaver og har ikke erstattet selvstændig analyse, faglig vurdering eller besvarelse af projektets problemformulering. Chatbotten er således ikke anvendt til at generere indhold i den analytiske eller metodiske del af projektet.

5.2 Datagrundlag

Projektet baserer sig udelukkende på det datasæt, der er stillet til rådighed af Business Viborg. Datasættet indeholder oplysninger om medlemskab, virksomhedsdemografi, branchetilknytning, kontaktaktivitet, eventdeltagelse samt ydet rådgivning. Alle data er pseudonymiserede og begrænset til et afgrænset tidsrum. Dette kan påvirke modellens generaliserbarhed over tid og dens evne til at indfange nyere tendenser i medlemsadfærd.

5.3 Modellens omfang og valg af algoritmer

Formålet med projektet er at udvikle en forklarlig og anvendelig prototype frem for en produktionsklar løsning. Der er derfor ikke foretaget omfattende hyperparameter-tuning for alle modeller. Seks modeller er testet – herunder Support Vector Machine, Random Forest og XGBoost – og performance er evalueret på baggrund af F1-score og AUC som de primære metrikker. Fokus har været på at finde en balance mellem prædiktiv nøjagtighed og forklaringskraft.

5.4 Systemintegration

Den udviklede løsning er implementeret som en webbaseret prototype i R og er ikke integreret med Business Viborgs interne systemer, såsom CRM- eller medlemsdatabaser. Modellen kan tilgås og anvendes lokalt gennem RStudio Cloud eller ved afvikling på en dedikeret server, men kræver manuel opdatering af data. Fremtidig integration og automatisering er oplagte skridt i en potentiel videreudvikling.

5.5 Juridiske og etiske vurderinger

Projektet indeholder en overordnet vurdering af de juridiske og etiske rammer med fokus på dataminimering, transparens og behandlingsgrundlag i henhold til GDPR. Der er ikke foretaget en fuld juridisk gennemgang, og tekniske løsninger som adgangsstyring, kryptering og samtykkehåndtering er ikke implementeret i prototypen. Disse aspekter betragtes som en integreret del af en eventuel implementeringsfase og bør afklares i samarbejde med relevante juridiske rådgivere og systemansvarlige.

6 Definitioner

I dette afsnit defineres centrale begreber og forkortelser anvendt gennem rapporten:

Churn: Når en virksomhed ophører med sit medlemskab i Business Viborg. I datasættet angives dette som en binær variabel, hvor 1 betyder churn og 0 betyder fortsat medlemskab.

Churn-model: En prædiktiv model, der estimerer sandsynligheden for, at en virksomhed cherner. Den er baseret på historiske medlemsdata og konstruerede forklaringsvariable.

Feature Engineering: Fremstilling af nye forklarende variable fra eksisterende data, som styrker modellens evne til at forudsige churn. Eksempler inkluderer medlemsanciennitet, kontaktaktivitet og deltagelse i arrangementer.

MeetingLength: Længden af det seneste dokumenterede møde med en virksomhed, målt i minutter. Bruges som indikator for relationens styrke.

har_haft_kontakt: En binær indikator for, om virksomheden har haft kontakt med Business Viborg (f.eks. møder, telefonopkald eller rådgivning).

deltaget_i_event: Binær variabel der angiver, om virksomheden har deltaget i mindst ét event i analyseperioden.

hjælp_kategori: En kategorisk variabel, der angiver typen af erhvervsfaglig støtte virksomheden har modtaget. Kategorierne er fx Strategi Udvikling, Organisation og Ledelse, Jura og Struktur m.fl.

medlem_antal_år: Antal år virksomheden har været medlem, beregnet som forskellen mellem analysedato og oprettelsesdato.

Machine Learning (ML): En metode til at bygge modeller, der kan lære mønstre i data og forudsige fremtidige hændelser. I projektet er ML anvendt til churn-forudsigelse.

Random Forest: En ML-algoritme, der kombinerer mange beslutningstræer for at skabe en robust og forklarlig model. Valgt som slutmodel i projektet.

ROC AUC: Et mål for modellens evne til at adskille churnere og ikke-churnere. En værdi tæt på 1 indikerer høj prædiktiv nøjagtighed.

F_meas (F1-score): Et samlet præstationsmål, som balancerer præcision og recall – særligt velegnet ved skæve datasæt.

Dashboard: Et interaktivt visualiseringsværktøj, der præsenterer churn-risici og medlemsindsigter på en overskuelig måde til brug i den daglige medlemspleje.

GDPR: EU's databeskyttelsesforordning. Projektet tager højde for centrale principper som dataminimering, transparens og legitimt behandlingsgrundlag.

7 Analyse

7.1 Dataforståelse og fordeling

Analysen tager udgangspunkt i et datasæt bestående af 2.966 medlemsvirksomheder tilknyttet Business Viborg, som har et selvstændigt P-nummer. Datasættet afspejler en betydelig variation med hensyn til virksomhedsstørrelse, branchetilhørsforhold og interaktionsniveau med organisationen.

En indledende fordeling afslører, at virksomheder uden dokumenteret kontakt eller deltagelse i arrangementer har markant højere churn-rate. Denne observation antyder, at fraværet af kontakt og engagement kan være centrale indikatorer for medlemsophør.

7.2 Egenskaber ved virksomheder med høj churn

For bedre at forstå hvordan forskellige former for engagement påvirker medlemsstatus, har vi kombineret to centrale variabler: kontakt med Business Viborg og deltagelse i events. Dette giver fire grupper, som varierer i deres relation til organisationen. Figuren nedenfor viser tydeligt, hvordan virksomheder med både kontakt og eventdeltagelse i langt højere grad fastholdes som medlemmer, mens fravær af begge faktorer er stærkt forbundet med churn.

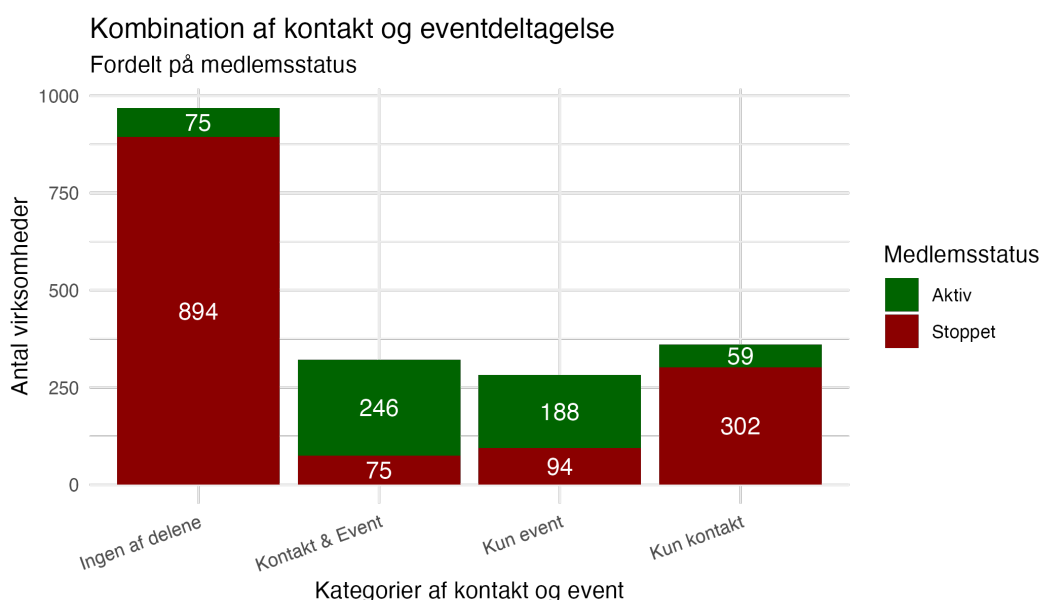


Figure 1: Fravær af begge faktorer er tæt forbundet med udmeldelse, mens dobbelt engagement viser stærk fastholdelse.

Virksomheder med begrænset kontakt til organisationen, lav deltagelse i arrangementer og uden dokumenteret interaktion har generelt en markant højere risiko for at opsige deres medlemskab. Dette underbygges af figuren nedenfor, der viser de fem postnumre med den højeste gennemsnitlige churn-risiko. Her ses, at geografiske områder med lav tilknytning til det centrale område (8800 Viborg) udviser særlig høj churn-sandsynlighed.

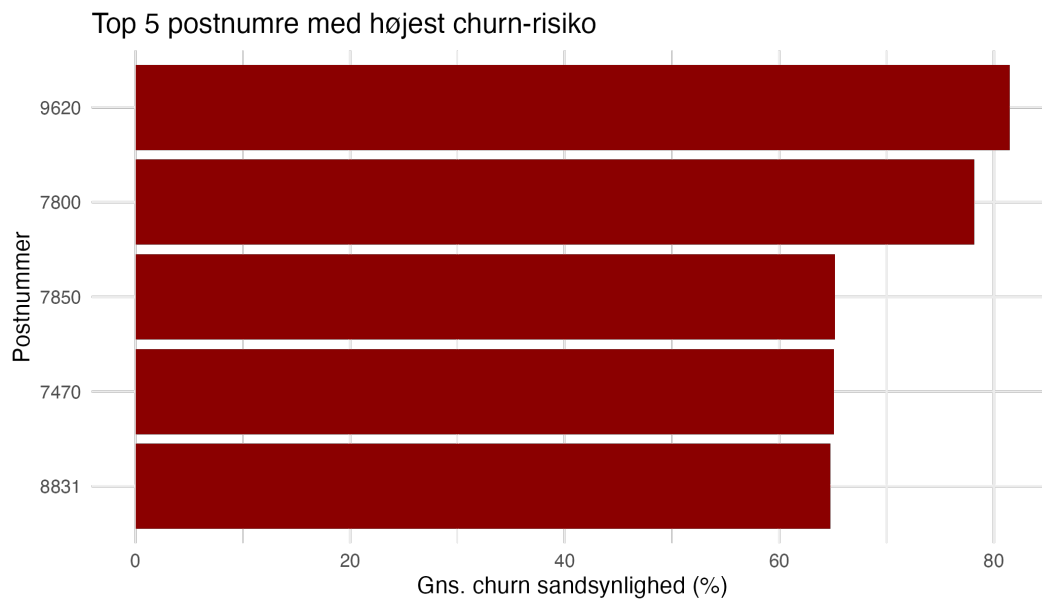


Figure 2: Geografisk afstand fra Viborgs centrum ser ud til at spille en rolle i udmeldelsestendens. 9620 Aalestrup, 7800 Skive, 7850 Stoholm, 7470 Karup, 8831 Løgstrup

Også på brancheniveau er der tydelige forskelle. Nogle brancher er kendetegnet ved begrænset interaktion med organisationen og har derfor en højere sandsynlighed for medlemsophør. Figuren herunder visualiserer de fem brancher med den højeste gennemsnitlige churn-risiko, hvilket bekræfter tendensen.

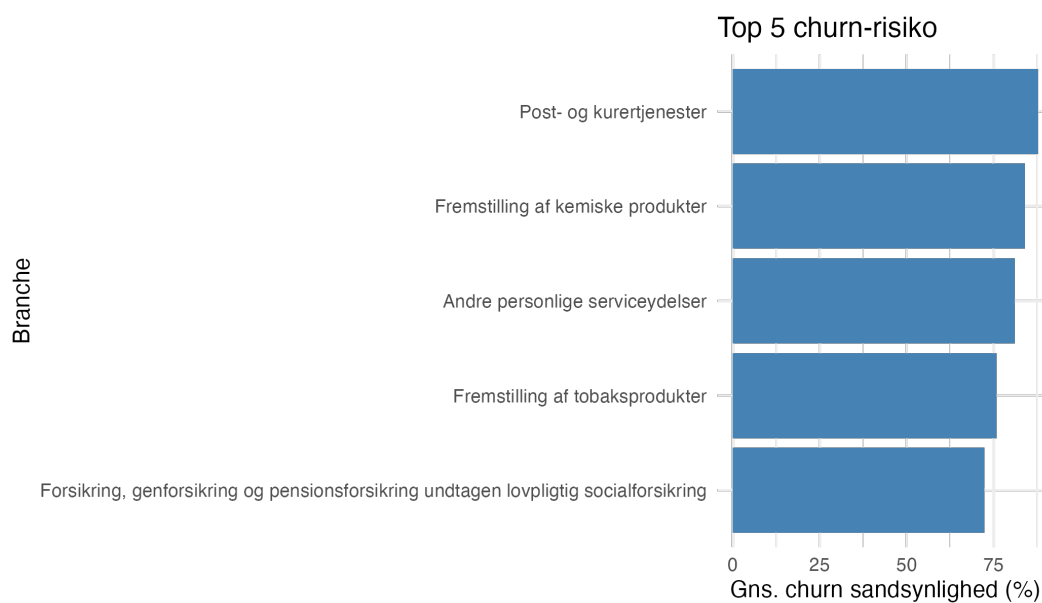


Figure 3: Brancher med lav netværkssværdi og specialiserede ydelser udviser generelt højere risiko for medlemsophør.

Omvendt findes der brancher, hvor medlemmerne i langt højere grad fastholdes. Disse brancher har ofte en mere stabil tilknytning og deltager aktivt i organisationens tilbud. Den følgende visualisering viser de fem brancher, hvor medlemmerne i størst omfang forbliver tilknyttet – hvilket indikerer, at der her eksisterer en stærkere relation og et større udbytte af medlemskabet.

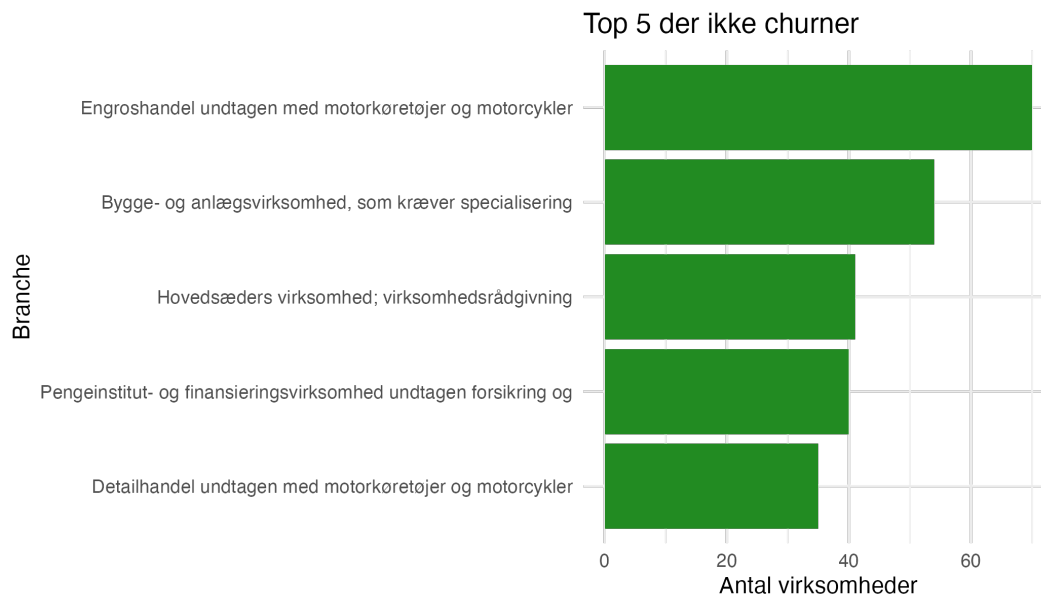


Figure 4: Branchenes høje fastholdelse kan skyldes stærkere relationer og oplevet værdi af netværket.

I de fem brancher hvor churnrisikoen er lille, er antallet af virksomheder højt. Dette kunne understøtte teorien om at netværksværdi har høj prioritet, blandt dem der er medlemmer af Business Viborg.

7.3 Feature engineering

På baggrund af ovenstående mønstre blev der konstrueret nye forklarende variabler for at styrke modellernes prædiktioner. De mest centrale inkluderer:

- `medlem_antal_år`: længden af medlemskab målt i år
- `har_haft_kontakt`: binær indikator for, om der har været nogen form for kontakt
- `deltaget_i_event`: binær indikator for eventdeltagelse

Disse variabler blev udledt på baggrund af domæneviden og eksplorativ analyse, og bidrog væsentligt til forbedret modelperformance.

7.4 Modelperformance

Seks machine learning modeller blev afprøvet: Support Vector Machine (SVM), XGBoost, Random Forest, logistisk regression, K-nearest Neighbors (KNN) og Naive Bayes.

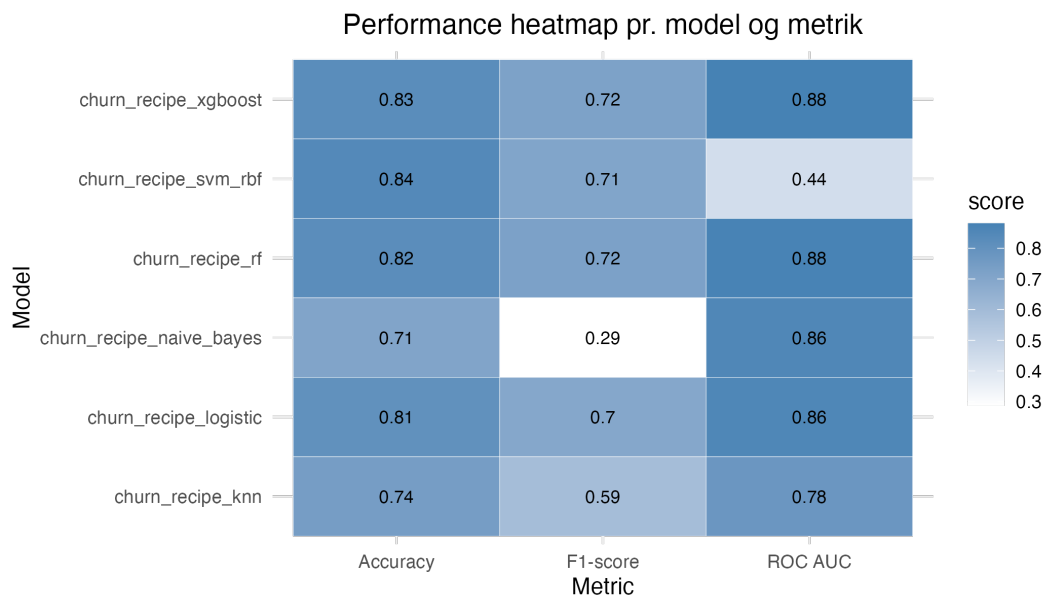


Figure 5: Farveintensitet viser performance – mørkere felter angiver højere score. Random Forest og XGBoost scorer generelt højt, mens Naive Bayes udviser lav F1-score.

På trods af XGBoost gode resultater blev Random Forest valgt som slutmodel. Dette skyldes modellens kombination af prædiktiv styrke og modelgennemsigtighed, hvilket gør den mere anvendelig i en praktisk kontekst. XGBoost blev fravalgt grundet behovet for yderligere parameteroptimering, som ikke var formålstjenligt inden for projektets rammer. Vi valgte den mest simple af de to modeller, da deres metrikker lå meget tæt.

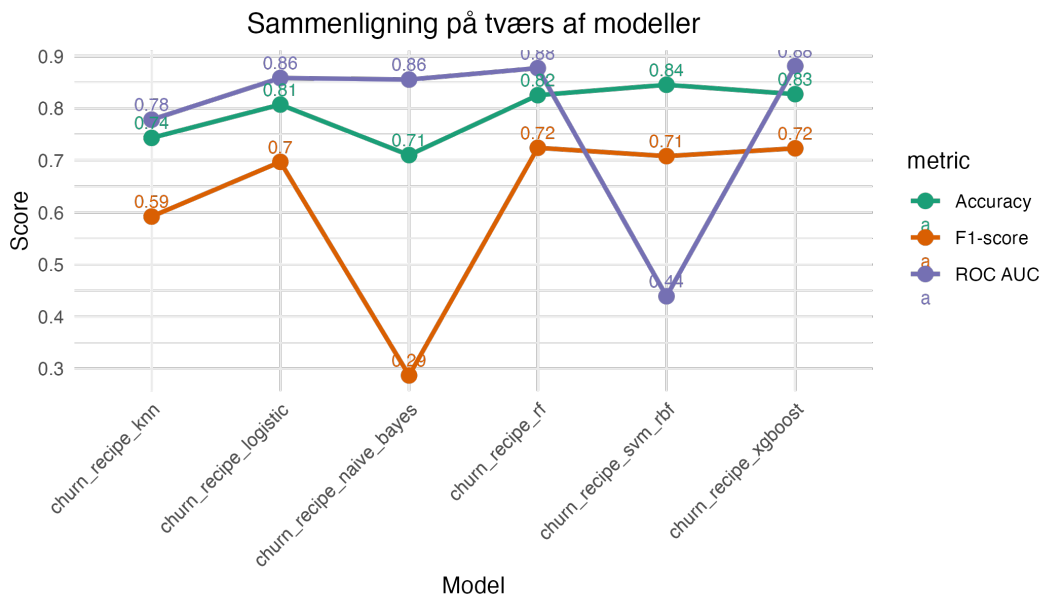


Figure 6: Lineplot over modelperformance. Random Forest og XGBoost opnår høj score på alle tre metrikker, hvilket underbygger deres styrke som robuste og præcise modeller.

7.5 Variable importance og indsigt

Ved hjælp af `vip()`-pakken blev de mest betydningsfulde variable i Random Forest-modellen identificeret. De fire vigtigste prædiktorer var:

- `deltaget_i_event_Ja`
- `deltaget_i_event_Nej`
- `MeetinLenght`
- `Employees`

Disse variable udgør tilsammen et stærkt grundlag for at forstå churn-mekanismer i Business Viborgs medlemsbase og bekræfter den eksplorative analyses fund.

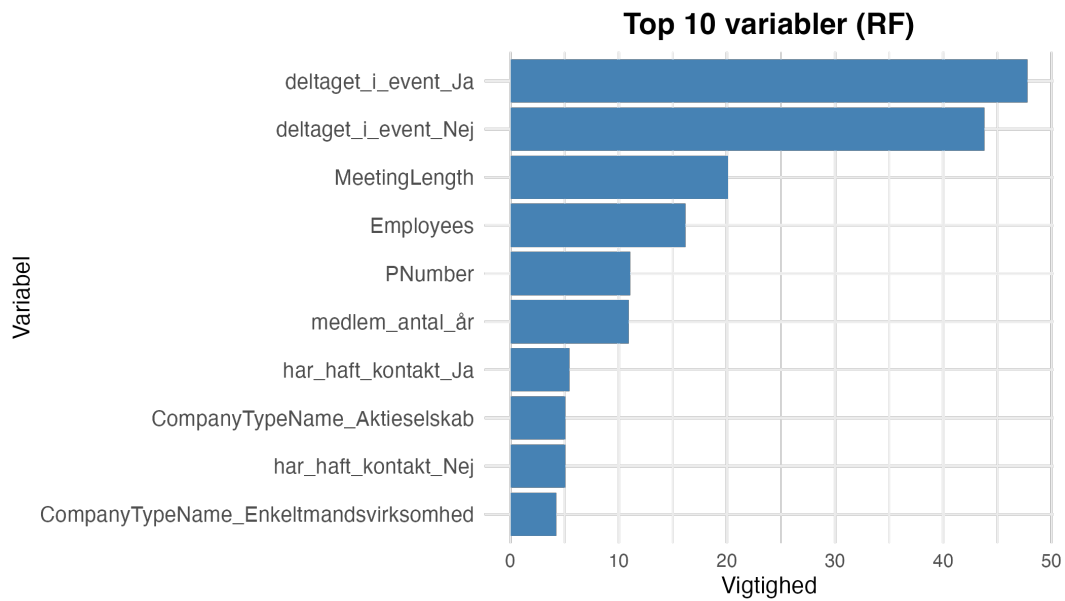


Figure 7: Eventdeltagelse, mødelængde og antal ansatte er blandt de mest betydningsfulde prædiktorer for churn.

8 Juridiske og etiske overvejelser

8.1 Introduktion

Udviklingen af en maskinlæringsbaseret churn-model for Business Viborg involverer behandling af medlemsdata, som i flere tilfælde kan tilknyttes identificerbare virksomheder og kontaktpersoner. Det er derfor afgørende, at både juridiske krav og etiske principper integreres som en central del af udviklingsprocessen. Dette afsnit belyser de væsentligste krav i henhold til EU's databeskyttelsesforordning (GDPR) samt centrale dataetiske hensyn, der bør overvejes i forbindelse med implementeringen af modellen.

8.2 Behandlingsgrundlag og dataminimering

Ifølge artikel 6 i GDPR må personoplysninger kun behandles, hvis der foreligger et lovligt behandlingsgrundlag. I denne kontekst vurderes det, at Business Viborg lovligt kan basere databehandlingen på den legitime interesse (artikel 6, stk. 1, litra f). Formålet – at fastholde medlemmer og styrke medlemsrelationer – vurderes som sagligt, nødvendigt og proportionalt i forhold til de registreredes forventninger.

Det er dog væsentligt, at formålet med databehandlingen er klart defineret og dokumenteret. Hvis data senere ønskes anvendt til fx automatiseret profilering eller målrettet markedsføring, skal formålet genvurderes, og samtykke kan blive nødvendigt.

Samtidig er det afgørende, at behandlingen lever op til GDPR's princip om dataminimering (artikel 5). Kun data, der er relevante og nødvendige i forhold til churn-prediktion, må inddrages. Følsomme eller overflødige oplysninger skal enten udelades eller anonymiseres, og modellen bør løbende evalueres for at sikre overensstemmelse med dette princip.

8.3 Transparens og oplysningspligt

Business Viborg er forpligtet til at informere sine medlemmer om, hvordan deres data anvendes. Denne oplysningspligt følger af artikel 13 og 14 i GDPR og indebærer, at medlemmerne skal oplyses om formål, behandlingsgrundlag, deres rettigheder samt hvordan de kan gøre indsigelse. Disse informationer bør være let tilgængelige og formidles i et klart og forståeligt sprog, fx via privatlivspolitikken eller velkomstmateriale.

Transparens bidrager ikke blot til juridisk overholdelse, men også til at opbygge tillid og styrke relationen til medlemmerne.

8.4 Pseudonymisering og adgangsforhold

Datasættet, der er anvendt til udvikling af modellen, er pseudonymiseret for analytikere. Det betyder, at personhenførbare oplysninger er fjernet eller maskeret, men ikke fuldt anonymiseret. Internt i Business

Viborg vil det fortsat være muligt at identificere enkelte virksomheder eller personer.

Det betyder, at alle krav i GDPR fortsat er gældende. Organisationen skal sikre passende adgangsstyring, begrænse adgangen til identificerbare data, og dokumentere hvilke medarbejdere har adgang til hvad. Endvidere skal der være klare retningslinjer for, hvordan data må bruges, og hvordan utilsigtet identifikation undgås.

8.5 Datasikkerhed og organisatorisk ansvar

I henhold til artikel 32 i GDPR skal Business Viborg etablere passende tekniske og organisatoriske foranstaltninger for at beskytte personoplysninger mod uautoriseret adgang, tab eller misbrug. Dette inkluderer: - Kryptering og adgangskontrol - Intern logning af dataadgang - Uddannelse af medarbejdere i datasikkerhed - Regelmæssig evaluering af sikkerhedspolitikker - Eventuelle databehandleraftaler med eksterne samarbejdspartnere

En systematisk tilgang til datasikkerhed er afgørende – ikke blot af juridiske årsager, men også for at opretholde tillid til organisationens datapraksis.

8.6 Ethiske overvejelser og ansvarlig anvendelse

Ud over de juridiske krav bør Business Viborg også forholde sig aktivt til de etiske implikationer ved at anvende en churn-model. Dataetiske principper foreslået af bl.a. Dataetisk Råd anbefaler, at teknologiske løsninger skal:

- Sætte mennesket i centrum
- Undgå diskrimination og skævvridning
- Skabe gennemsigtighed og forklarelige beslutninger

Det er vigtigt, at modellen ikke bruges til at stigmatisere bestemte medlemsgrupper eller segmenter. Anvendelsen af churn-risiko bør altid ledsages af kritisk refleksion og inddragelse af menneskelig dømmekraft i den endelige beslutning om handling. Medlemsdialogen bør være præget af forståelse og imødekommenhed – ikke automatiseret kategorisering.

8.7 Delkonklusion

Sammenfattende vurderes det, at Business Viborgs churn-projekt kan gennemføres i overensstemmelse med GDPR og etiske retningslinjer, forudsat at visse betingelser overholdes. Der skal foreligge et klart og dokumenteret behandlingsgrundlag, medlemmerne skal informeres tydeligt og rettidigt, og datasættet skal reduceres til relevante oplysninger i henhold til formålet. Desuden skal organisationen sikre passende tekniske og organisatoriske sikkerhedsforanstaltninger og være opmærksom på, at data fortsat er personhenførbare, selvom de fremstår pseudonymiserede. Afslutningsvis bør Business Viborg betragte dette projekt som et skridt mod mere datadrevet medlemsservice – men også som en mulighed for at styrke sin rolle som en ansvarlig og gennemsigtig aktør i det lokale erhvervsliv. Det forudsætter, at juridiske forpligtelser og dataetiske principper ikke ses som forhindringer, men som en integreret del af en moderne og tillidsbaseret organisation.

9 anbefaling

På baggrund af analysen anbefales det, at Business Viborg anvender den udviklede churn-model som et beslutningsunderstøttende værktøj i det opsøgende medlemsarbejde. Modellen kan identificere virksomheder med høj risiko for udmeldelse og derved muliggøre en mere målrettet og proaktiv indsats fra medlemskonsulenterne. Det foreslås, at: ## Dashboards og churn-risiko indgår som fast element i konsulenterne arbejdsrutiner og prioritering af medlemmer.

9.1 Medlemspleje prioriteres over for virksomheder uden kontakt, eventdeltagelse eller modtaget hjælp, da disse faktorer er stærkt associeret med churn.

9.2 Løbende opdatering af modellen sikres ved at integrere churn-værktøjet i Business Viborgs CRM eller medlemsdatabase, så nye data automatisk indgår i fremtidige analyser.

9.3 Etisk og transparent kommunikation om brugen af data indgår i medlemsdialogen for at styrke tilliden og sikre overholdelse af GDPR.

10 Konklusion

Dette projekt har demonstreret, hvordan Business Viborg kan anvende medlemsdata og maskinlæring til at forudsige churn og styrke den strategiske medlemspleje. Med udgangspunkt i en analyse af 2.966 medlemsvirksomheder og afprøvning af seks modeller blev der udviklet en forklarlig og prædiktiv Random Forest-model ($F1 = 0,71$, $AUC = 0,86$). Modellen integreres i et dashboard, som giver medlemskonsulenterne mulighed for at prioritere opsøgende indsats på et datadrevet grundlag.

Analysen viser, at variabler relateret til kontakt, eventdeltagelse og modtaget erhvervshjælp er blandt de vigtigste drivere for fastholdelse – i tråd med både tidligere analyser og forretningsmæssig intuition. Løsningen er udviklet med respekt for GDPR og dataetik og udgør et realistisk og skalerbart værktøj for medlemsorganisationer, der ønsker at arbejde mere strategisk med churn.

Det vurderes, at modellens præcision kunne forbedres yderligere, hvis der var adgang til oplysninger om, hvornår udmeldelser fandt sted (Timestamp). Denne information kunne give indblik i, hvordan churn-risikoen udvikler sig over tid og bidrage til en endnu mere målrettet indsats.

Projektet besvarer dermed problemformuleringen ved at kombinere teknisk modeludvikling med brugervenlig formidling og ansvarlig dataanvendelse. Perspektiverende åbner det op for en bredere anvendelse af datadrevne beslutningsstøttesystemer i mindre organisationer med store medlemsmæssige ambitioner.

11 Literaturliste

AI

OpenAI. (2025). ChatGPT (4.0). <https://chatgpt.com/>

Bøger

WWW-dokumenter

Undervisningsmaterialer

12 Bilagsoversigt

- Bilag 1: GDPR – Europa-Parlamentets og Rådets Forordning (EU) 2016/679

Konsolideret og officielt dokument om databeskyttelse i EU. Anvendes som juridisk referenceramme i projektets afsnit om etiske og juridiske overvejelser.

Tilgængelig via: <https://eur-lex.europa.eu/legal-content/DA/TXT/?uri=CELEX%3A32016R0679>

- Bilag 2: x
- Bilag 3: x
- Bilag 4: x
- Bilag 5: x
- Bilag 6: x

```
# -----
# 1. Load data
# -----

# Indlæser alle nødvendige datasæt
meetings <- readRDS("data/meetings.rds")
events <- readRDS("data/events.rds")
event_participants <- readRDS("data/event_participants.rds")
company_contacts <- readRDS("data/company_contacts.rds")
all_contact <- readRDS("data/all_contact.rds")
all_companies <- readRDS("data/all_companies.rds")
old_projects <- readRDS("data/old_projects.rds")
```

```
# -----
# 2. Merge datasets
# -----

# -----
# 2.1: Fjern dubletter og behold første registrering pr. virksomhed
# -----

meetings_unique <- meetings |>
  group_by(CompanyId) |>
  summarise(across(everything(), first))    # Første møde pr. virksomhed

events_unique <- events |>
  group_by(Cvr) |>
  summarise(across(everything(), first))    # Første event pr. virksomhed
```



```

event_participants_unique <- event_participants |>
  group_by(Cvr) |>
  summarise(across(everything(), first))    # Første deltagerinfo pr. virksomhed

# -----
# 2.2: Saml alle datasæt med left_join og ryd op i dubletter
# -----

merged_df <- all_companies |>
  left_join(company_contacts, by = "CompanyId") |>    # Join kontaktpersoner
  left_join(all_contact, by = "contactId") |>        # Join kontaktinfo
  left_join(meetings_unique, by = "CompanyId") |>    # Join mødedata
  rename(Cvr = "z_companies_1_CVR-nummer_1") |>      # Omdøber kolonnen til
                                                    # "Cvr", så den matcher med events
  left_join(events_unique, by = "Cvr") |>            # Join eventinfo
  left_join(event_participants_unique, by = "Cvr") |> # Join deltagerinfo
  select(-ends_with(".y"), -ends_with(".x"))         # Fjerner dublet-kolonner

# -----
# 2.3: Klargør datasæt: fjern anonyme oplysninger og omdøb kolonnenavne
# -----

# Fokus: Unikke virksomheder via PNumber (produktionsenhedsnummer)
# Det giver os 2966 unikke observationer.

merged_df <- merged_df |>
  select(-z_companies_1_Firmanavn_1, -z_contacts_1_Email_1)

```

```

# Fjerner anonymiserede data

# Standardiser kolonnenavne for overskuelighed
colnames(merged_df) <- c(
  "BusinessCouncilMember", "CompanyDateStamp", "CompanyId", "CompanyType",
  "CVR", "Employees", "PostalCode", "CompanyTypeName", "PNumber", "Country",
  "NACECode", "CompanyStatus", "AdvertisingProtected", "ContactId",
  "CompanyOwnerId", "ContactLastUpdated", "TitleChanged", "LocationChanged",
  "CreatedBy", "MeetingLength", "Firstname", "UserRole", "Initials",
  "EventExternalId", "EventPublicId", "Description", "LocationId",
  "MaxParticipants", "EventLength", "EventId"
)

# -----

# 2.4: Fjern dubletter og irrelevante kolonner
# Udfyld manglende værdier i eventkolonner med "Ingen event"
# -----

# Beholder unikke virksomheder, fjerner irrelevante kolonner,
# og udfylder NA i eventdata
merged_unique <- merged_df |>
  distinct(PNumber, .keep_all = TRUE) |> # Beholder én række pr. PNumber
  select(-TitleChanged, -LocationChanged, -CreatedBy, -Firstname,
    # Fjerner irrelevante variabler
    -UserRole, -Initials, -ContactLastUpdated) |>
  mutate(across( # Erstatte NA i event-kolonner med "Ingen event"
    c(MeetingLength, EventExternalId, EventPublicId, Description,
      LocationId, MaxParticipants, EventLength, EventId),

```

```

    ~ if_else(is.na(.), "Ingen event", as.character(.))
  ))

# Rens MeetingLength og konverter til numerisk (fjern " mins")
merged_unique <- merged_unique |>
  mutate(
    MeetingLength = ifelse(MeetingLength == "Ingen event", "0 mins",
                           MeetingLength),
    MeetingLength = as.numeric(str_remove(MeetingLength, " mins"))
  )

# -----
# 2.5: Splitter NACECode i kode og beskrivelse,
# fjern original kolonne og NA-rækker
# -----
merged_unique <- merged_unique |>
  mutate(
    Employees = if_else(is.na(Employees), "Ukendt", as.character(Employees)),
    # NA -> "Ukendt"

    NACECode = if_else(is.na(NACECode), "Ukendt", as.character(NACECode)),
    # NA -> "Ukendt"

    Nacecode = if_else(NACECode == "Ukendt", "Ukendt",
                       str_extract(NACECode, "[0-9]+")),
    # Hent kode
    Nacebranche = if_else(NACECode == "Ukendt", "Ukendt",
                          str_remove(NACECode, "[0-9]+\\s*"))
  )

```

```

# Hent branche
) |>
select(-NACECode) |> # Fjerner original NACECode-kolonne
na.omit()            # Fjerner rækker med NA-værdier

# -----
# 2.6: Tjek for tilbageværende NA-værdier
# -----
# colSums(is.na(merged_unique))

# -----
# 2.7: Gem det rensede datasæt til senere brug
# -----
saveRDS(merged_unique, "merged_unique.rds")

# -----
# 2.8: Merge old_projects (frivillig) med virksomhedsdata
# -----

# Omdøb SMVContactId til ContactId
old_projects <- old_projects |>
  rename(ContactId = SMVContactId) # Omdøb kolonne for at matche join

# Gem kolonnenavne fra old_projects (ekskl. ContactId)
old_project_cols <- setdiff(names(old_projects), "ContactId")
cols_to_fill <- setdiff(old_project_cols, c("Id", "SMVCompanyId", "SharedWith"))

```

```

# Join med merged_unique og erstat NA med "Tom"
merged_unique_old_projects <- merged_unique |>
  left_join(old_projects, by = "ContactId") |> # Merger på ContactId
  select(-Id, -SMVCompanyId, -SharedWith) |> # Fjerner unødvendige kolonner
  mutate(across(all_of(cols_to_fill),
    ~ if_else(is.na(.), "Tom", as.character(.)))) |> # NA → "Tom"
  distinct(PNumber, .keep_all = TRUE) # Behold unikke virksomheder

# -----

# 2.9: Tjek for NA-værdier i det udvidede datasæt
# -----

# colSums(is.na(merged_unique_old_projects))

merge_datasets <- merged_unique_old_projects

```

```

# -----

# 3. Cleaning data
# -----

# -----

# 3.1: Første kig på datastrukturen
# Giver et hurtigt overblik over variabelnavne, typer og eksempel-værdier
# -----

# glimpse(merge_datasets)

# -----

# 3.2: Tæl hvor mange NA (manglende værdier) der findes i hver kolonne
# Dette er nyttigt for at forstå, hvor der evt. skal renses eller imputeres

```

```

# -----
# Tjekker for manglende værdier (NA) i alle variabler
na_count <- merge_datasets |>
  summarise(across(everything(), ~ sum(is.na(.)))) |>
  pivot_longer(everything(), names_to = "variable", values_to = "na_count")

# -----
# 3.3: Rensning af kolonnenavne
# Fjerner forstyrrende elementer som tal, specialtegn og mellemrum
# Gør kolonnenavne nemmere at bruge i videre analyser og modeller
# -----
# Rydder op i variabelnavne: fjerner tal, specialtegn og whitespace
names(merge_datasets) <- names(merge_datasets) |>
  str_remove("^([0-9]+_1*\\s*)") |>      # Fjerner startende tal/1-taller
  str_replace_all("[ /\\-]+", "_") |>   # Erstatte mellemrum og specialtegn med _
  str_replace_all("_+", "_") |>        # Fjerner dobbelte underscores
  str_remove("_$") |>                  # Fjerner underscore i slutningen
  str_trim()                           # Trim whitespace

# Udskriver de rensede kolonnenavne
# print(names(merge_datasets))

# -----
# 3.4: Fjern irrelevante kolonner (ID'er og tekniske felter)
# Disse kolonner bruges ikke i analysen og fjernes derfor fra datasættet
# -----
clean_data <- merge_datasets |>
  dplyr::select(-ContactId, -CompanyOwnerId, -EventExternalId,

```

```

      -EventPublicId, -LocationId, -Tekstfelt, -CompanyType)

# -----
# 3.5: Erstatning og konvertering af værdier
# - Tekst som "Tom", "Ukendt" og "Ingen event" → NA
# - NA i tekstfelter bliver til "Ukendt"
# - NA i tal bliver til 0
# - Udvalgte kolonner konverteres til numerisk format
# -----

clean_data <- clean_data |>
  mutate(
    across(
      c(CVR, Nacecode, PostalCode, PNumber, MaxParticipants,
        EventLength, Employees), ~ as.numeric(ifelse(.x %in% c(" ", "", "Tom",
          "Ukendt", "Ingen event"), NA, .x))
    ),
    across(where(is.character), ~ replace_na(.x, "Ukendt")), # Tekst: NA →
    # "Ukendt"
    across(where(is.numeric), ~ replace_na(.x, 0))           # Tal: NA → 0
  )

# -----
# 3.6: Konverter dato-kolonner til rigtig datoformat
# Vigtigt hvis man senere skal beregne fx forskel i tid
# -----

CompanyDateStamp <- as.Date(clean_data$CompanyDateStamp, format = "%Y-%m-%d")
Kontaktdato      <- as.Date(clean_data$Kontaktdato, format = "%Y-%m-%d")

```

```
# -----  
# 3.7: # Viser datastruktur efter rensning  
# -----  
# glimpse(clean_data)
```

```
# -----  
# 4. Feature Engineering  
# -----  
  
# -----  
# 4.1: # Viser datastruktur efter rensning  
# -----  
  
# glimpse(clean_data) # Bruger glimpse til at få et hurtigt overblik over data  
  
# -----  
# 4.2: Opretter en ny variabel, der beregner hvor mange år  
# en virksomhed har været medlem. Vi bruger CompanyDateStamp (oprettelsesdato)  
# og beregner forskellen til dags dato.  
# -----  
feature_engineering <- clean_data |>  
  mutate(  
    medlem_antal_år = round(  
      as.numeric(difftime(Sys.Date(), as.Date(CompanyDateStamp),  
        units = "days")) / 365,
```



```

0
)
)

# -----
# 4.3: Rensning af Employees-kolonnen (antal ansatte).
# Nogle gange kan tal være formateret med punktummer (f.eks. "1.000")
# eller mellemrum (f.eks. "1 000").
# Disse fjernes, så kolonnen kan konverteres til numerisk format
# -----

feature_engineering <- feature_engineering |>
  mutate(
    Employees = Employees |>
      str_replace_all("\\.", "") |>      # Fjerner punktummer
      str_replace_all("\\s+", "") |>    # Fjerner mellemrum
      as.numeric()                      # Konverterer til tal
  )

# -----
# 4.4: Oversættelse af virksomhedstyper til mere læsbare formater
# Eksempel: "A/S" bliver til "Aktieselskab"
# -----

feature_engineering <- feature_engineering |>
  mutate(
    CompanyTypeName = str_replace_all(CompanyTypeName, "A/S", "Aktieselskab"),
    CompanyTypeName = str_replace_all(CompanyTypeName, "ApS", "Anpartsselskab"),
    CompanyTypeName = str_replace_all(CompanyTypeName, "IVS", "Iværksætterselskab"),

```

```

CompanyTypeName = str_replace_all(CompanyTypeName, "P/S", "Partnerselskab"),
CompanyTypeName = str_replace_all(CompanyTypeName, "K/S", "Kommanditselskab")
)

# -----
# 4.5: Tilføj branchebetegnelse baseret på NACE-koder
# NACE er en standard for brancheklassifikation (fx "01 Landbrug")
# Vi bruger de første to cifre til at matche mod en lookup-tabel med branchenavne
# -----
nace_lookup <- read_delim("data/nace_branchenavne.csv", delim = ";") |>
  select(KODE, TITEL) |>
  rename(Nace_kort = KODE, Branche_navn = TITEL)

```

Rows: 1732 Columns: 10

-- Column specification -----

Delimiter: ";"

chr (6): KODE, TITEL, GENERELLE_NOTER, INKLUDERER, INKLUDERER_OGSÅ, EKSKLUDERER

dbl (2): SEKvens, NIVEAU

lgl (2): PARAGRAF, MÅLEENHED

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

# Tilføj branchebetegnelse baseret på Nacecode og fjern overflødige kolonner
# Lav en ny kolonne med de første to cifre af Nacecode
feature_engineering <- feature_engineering |>
  mutate(Nace_kort = substr(Nacecode, 1, 2)) |> # Udtrækker de to første cifre
  select(-Nacebranche) |> # Fjerner den gamle kolonne
  left_join(nace_lookup, by = "Nace_kort") |> # Slår op i brancheregister

```

```

mutate(
  Branche_navn = replace_na(Branche_navn, "Ukendt"),
  # Hvis ingen match, brug "Ukendt"
  Branche_navn = as.factor(Branche_navn)
  # Gør den klar til ML (kategorisk)
) |>
select(-Nacecode, -Nace_kort) |>      # Fjerner unødvendige kolonner
relocate(Branche_navn, .after = PNumber) # Flytter Branche_navn efter PNumber

# -----
# 4.6: Opretter 2. feature/variabel - har virksomheden haft kontakt?
# Vi kigger på flere kolonner og vurderer:
# hvis mindst én ikke er "Tom", så har der været kontakt
# -----

feature_engineering <- feature_engineering |>
mutate(
  har_haft_kontakt = if_else(
    Virksomhedsbesøg != "Tom" | Telefonkontakt != "Tom" |
    Konsulent_Navn != "Tom" | Notat != "Tom" | Kontaktdato != "Tom",
    "Ja", "Nej")
  ) |>
select(-Virksomhedsbesøg, -Telefonkontakt, - Konsulent_Navn,
      -Notat, -Kontaktdato)

```

```

# -----
# 4.7: Opretter 3. feature/variabel - har virksomheden deltaget i event?
# Hvis EventLength er større end 0, siger vi "Ja", ellers "Nej"
# -----

feature_engineering <- feature_engineering |>
  mutate(deltaget_i_event = if_else(as.numeric(EventLength) > 0, "Ja", "Nej"))

# -----
# 4.8: Skaber kategorier der viser virksomhedens behov for hjælp
# Her grupperes TRUE/FALSE-kolonner i temaer som Strategi, Jura, Økonomi osv.
# Den viser, hvilken overordnet type hjælp virksomheden har modtaget.
# -----

feature_engineering <- feature_engineering |>
  # Sørg for at konvertere kolonnerne til logiske værdier (TRUE/FALSE)
  mutate(across(matches("^\\d+_1"), ~ .x != "FALSE" & .x != "Tom")) |>
  mutate(

# Opretter en enkelt variabel, der kategoriserer virksomheden baseret på de
# 8 områder
  hjælp_kategori = case_when(

# Hvis virksomheden har søgt hjælp til strategi/emner som
# forretningsidé, produkt osv.
  (as.logical(Kundeportefølje) | as.logical(Forretningsmodel) |
    as.logical(Forretningsidé) | as.logical(Produktportefølje))

```

```

    ~ "Strategi Udvikling",

# Hvis fokus har været på markedsføring, branding eller PR
    (as.logical(Markedsføring) | as.logical(Branding) |
     as.logical(Kommunikation_og_PR)) ~ "Marketing og Kommunikation",

# Hvis der er søgt hjælp til salg, eksport eller markedsposition
    (as.logical(Salg) | as.logical(Eksport) |
     as.logical(Markedsposition)) ~ "Salg og Eksport",

# Hvis der har været fokus på ledelse, netværk eller organisation
    (as.logical(Medarbejdere) | as.logical(Netværk) |
     as.logical(Samarbejdspartnere) | as.logical(Ejer_og_bestyrelse))
     ~ "Organisation og Ledelse",

# Hvis det handler om økonomi, finansiering eller fonde
    (as.logical(Økonomistyring) | as.logical(Finansiering) |
     as.logical(Kapitalfond) | as.logical(Vækstfonden) |
     as.logical(Innovationsfonden)) ~ "Økonomi og Finansiering",

# Hvis det handler om daglig drift, it-systemer eller forretningsgange
    (as.logical(Leverance_og_projektstyring) | as.logical(IT_systemer) |
     as.logical(Faciliteter) |
     as.logical(Forretningsgange)) ~ "Drift og Systemer",

# Hvis fokus er på jura, ejerskifte mv.
    (as.logical(Juridiske_forhold) |
     as.logical(Ejerskifte_og_generationsskifte)) ~ "Jura og Struktur",

```

```

# Hvis der er søgt støtte gennem offentlige ordninger
(as.logical(EU_Kontoret_i_DK_Interreg) | as.logical(Erhvervshuset) |
 as.logical(FN_1) |
 as.logical(Andre_nationale_ordninger)) ~ "Støtteordninger",

# Tilføjelse af de nye kategorier
(as.logical(Uddannelse_kompetenceudvikling) |
 as.logical(Vidensordninger) |
 as.logical(IV_Vejledning) |
 as.logical(Virksomhedsbesøg_Virksomhed_under_3_år) |
 as.logical(I_Værkstedet) |
 as.logical(Klippekort_Udleveret) |
 as.logical(Væksthjul_Screening) |
 as.logical(Agro_Business_Park) |
 as.logical(Konsulent_virksomhed_uden_for_Kommunen_DK) |
 as.logical(Lokal_konsulent_eller_virksomhed) |
 as.logical(Indenrigsministeriet_The_Trade_Council) |
 as.logical(Produktudviklin)) ~ "Andre Hjælpeordninger",

TRUE ~ "Ingen specifik hjælp"
)
) |>

# Ryd op ved at fjerne de originale variabler der er brugt til grupperingen
select(-c(
  Kundeportefølje, Forretningsmodel, Forretningsidé, Produktportefølje,
  Markedsføring, Branding, Kommunikation_og_PR,
  Salg, Eksport, Markedsposition,
  Medarbejdere, Netværk, Samarbejdspartnere, Ejer_og_bestyrelse,

```

```

Økonomistyring, Finansiering, Kapitalfond, Vækstfonden, Innovationsfonden,
Leverance_og_projektstyring, IT_systemer, Faciliteter, Forretningsgange,
Juridiske_forhold, Ejerskifte_og_generationsskifte,
EU_Kontoret_i_DK_Interreg, Erhvervshuset, FN_1, Andre_nationale_ordninger,
Uddannelse_kompetenceudvikling, Vidensordninger, IV_Vejledning,
Virksomhedsbesøg_Virksomhed_under_3_år, I_Værkstedet,
Klippekort_Udleveret, Væksthjul_Screening, Agro_Business_Park,
Konsulent_virksomhed_uden_for_Kommunen_DK, Lokal_konsulent_eller_virksomhed,
Indenrigsministeriet_The_Trade_Council, Produktudviklin
))

# Tjek resultatet
# glimpse(feature_engineering)

# -----
# 4.9: Behold kun aktive virksomheder
# -----

feature_engineering <- feature_engineering |>
  filter(CompanyStatus %in% c("Aktiv", "NORMAL")) |>
  dplyr::select(-CompanyDateStamp, -CompanyId, -CVR, -Country,
    -CompanyStatus, -AdvertisingProtected, -MaxParticipants, -Description,
    -EventLength, -EventId, -Andet) # Sletter de kolonner vi ikke vil bruge

# -----
# 4.10: Tilføj churn-kolonne
# Opretter ny kolonne kaldet 'churn', viser om virksomheden er stoppet som medlem.
# Hvis BusinessCouncilMember er TRUE (virksomheden er medlem), sættes churn = 0

```

```

# Hvis BusinessCouncilMember er FALSE (virksomheden har forladt fællesskabet),
# sættes churn = 1
# -----
feature_engineering <- feature_engineering |>
  mutate(churn = if_else(BusinessCouncilMember == TRUE, 0, 1)) |>
  select(-BusinessCouncilMember)

# -----
# 4.11: Konverterer udvalgte kolonner til faktorer,
# som er nødvendigt for ML-modeller
# En faktor er en kategorisk variabel - dvs. den indeholder en begrænset mængde
# unikke værdier (kategorier). # Eksempler på faktorer: postnumre, ja/nej,
# virksomhedsformer (ApS, A/S, IVS osv.)
# I maskinlæring skal sådanne kolonner være faktorer,
# så algoritmerne forstår dem som kategorier og ikke som tekst.
# -----
feature_engineering <- feature_engineering |>
  mutate(
    CompanyTypeName = as.factor(CompanyTypeName),
    har_haft_kontakt = as.factor(har_haft_kontakt),
    deltaget_i_event = as.factor(deltaget_i_event),
    hjælp_kategori = as.factor(hjælp_kategori),
    PostalCode = as.factor(PostalCode),
    churn = as.factor(churn)
  )

# -----

```



```
# 4.12: Gem det færdigbehandlede datasæt til senere analyse eller modellering
```

```
# -----
```

```
write_rds(feature_engineering, "data/feature_engineered_data.rds")
```

```
# Helper-funktion: Henter og opdaterer .rds-filer fra de andre branches
```

```
# =====
```

```
# 5. Eksplorativ Dataanalyse (EDA)
```

```
# =====
```

```
#Medlemsstatus
```

```
# 5.1 Søjlediagram: Fordeling af medlemsstatus
```

```
# 5.2 Boxplot: Medlemsstatus vs. år som medlem
```

```
# 5.3 Histogram: Fordeling af medlemsår
```

```
#Adfærd
```

```
# 5.4 Søjlediagram: Eventdeltagelse blandt aktive medlemmer
```

```
# 5.5 Søjlediagram: Kombination af kontakt og eventdeltagelse vs. medlemsstatus
```

```
# 5.6 Søjlediagram: Hjælpekategorier blandt aktive deltagere vs. ikke-deltagere
```

```
#Baggrundssata: Branche og geografi
```

```
# 5.7 Søjlediagram: Branchefordeling blandt aktive medlemmer
```

```
# 5.8 Søjlediagram: Eventdeltagelse fordelt på branche
```

```
# 5.9 Søjlediagram: Hjælpekategorier fordelt på branche
```

```
# 5.10 Søjlediagram: Eventdeltagelse pr. postnummer
```

```
#Engagement og korrelationer
```

```

# 5.11 Søjlediagram: Gennemsnitlig mødelængde pr. branche
# 5.12 Boxplot: Mødelængde vs. eventdeltagelse
# 5.13 Boxplot: Mødelængde vs. medlemsstatus
# 5.14 Histogram: Fordeling af mødelængder
# 5.15 Korrelationsmatrix: Numeriske variable

# -----
# Indlæs data
# -----

featured <- readRDS("data/feature_engineering.rds")

# -----
# Funktion: Identificer outliers med IQR-metoden
# -----

# Vi bruger interkvartil-afstanden (IQR) til at identificere outliers.
# Observationer udenfor [Q1 - 1.5*IQR, Q3 + 1.5*IQR] regnes som outliers.

find_outliers <- function(x) {
  iqr <- IQR(x, na.rm = TRUE)
  lower <- quantile(x, 0.25, na.rm = TRUE) - 1.5 * iqr
  upper <- quantile(x, 0.75, na.rm = TRUE) + 1.5 * iqr
  x < lower | x > upper
}

# -----
# 5.1 Søjlediagram: Fordeling af medlemsstatus
# -----

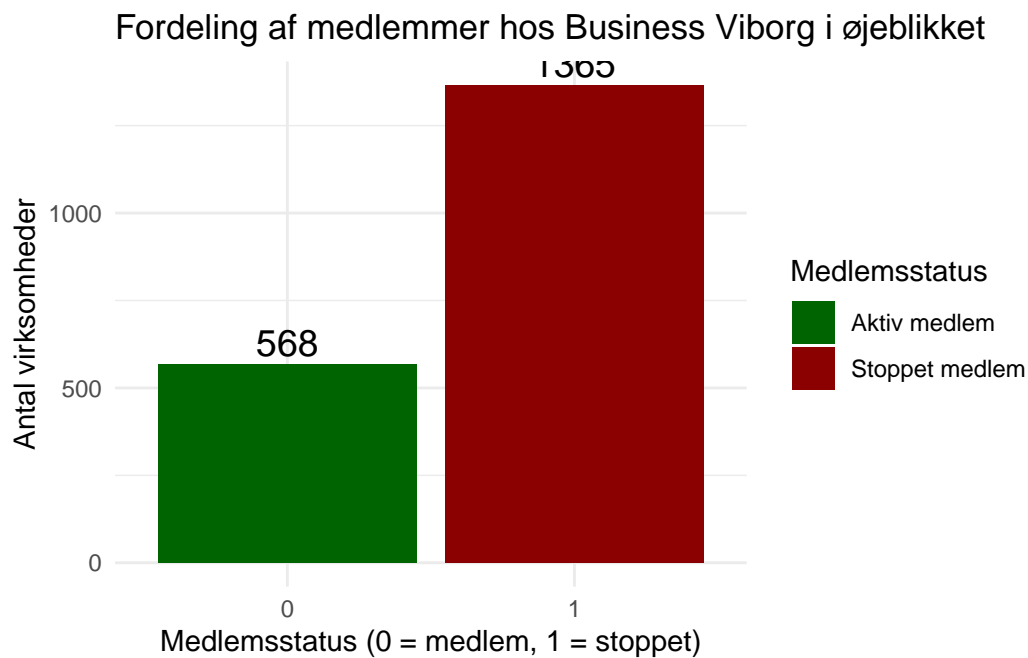
# En simpel søjlediagram der viser, hvor mange virksomheder der er

```

```
# henholdsvis aktive og stoppepe.

featured |>
  ggplot(aes(x = factor(churn), fill = factor(churn))) +
  geom_bar() +
  geom_text(stat = "count", aes(label = ..count..), vjust = -0.3, size = 5) +
  scale_fill_manual(
    values = c("0" = "darkgreen", "1" = "darkred"),
    labels = c("0" = "Aktiv medlem", "1" = "Stoppet medlem"),
    name = "Medlemsstatus"
  ) +
  labs(
    title = "Fordeling af medlemmer hos Business Viborg i øjeblikket",
    x = "Medlemsstatus (0 = medlem, 1 = stoppet)",
    y = "Antal virksomheder"
  ) +
  theme_minimal()
```

Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
 i Please use `after_stat(count)` instead.



```
ggsave("images/EDA_1_fordeling_medlemmer.png", width = 7, height = 4, dpi = 300)

# -----
# 5.2 Boxplot: Medlemsstatus vs. år som medlem
# -----
# Vi tjekker hvor længe virksomheder har været medlem afhængigt af medlemsstatus
# og tilføjer info om outliers, gennemsnit og antal.

featured <- featured |>
  mutate(outlier_medlemsår = find_outliers(memlem_antal_år))

nøgletal <- featured |>
  group_by(churn) |>
  summarise(
    antal = n(),
    gennemsnit = round(mean(memlem_antal_år, na.rm = TRUE), 1),
```

```

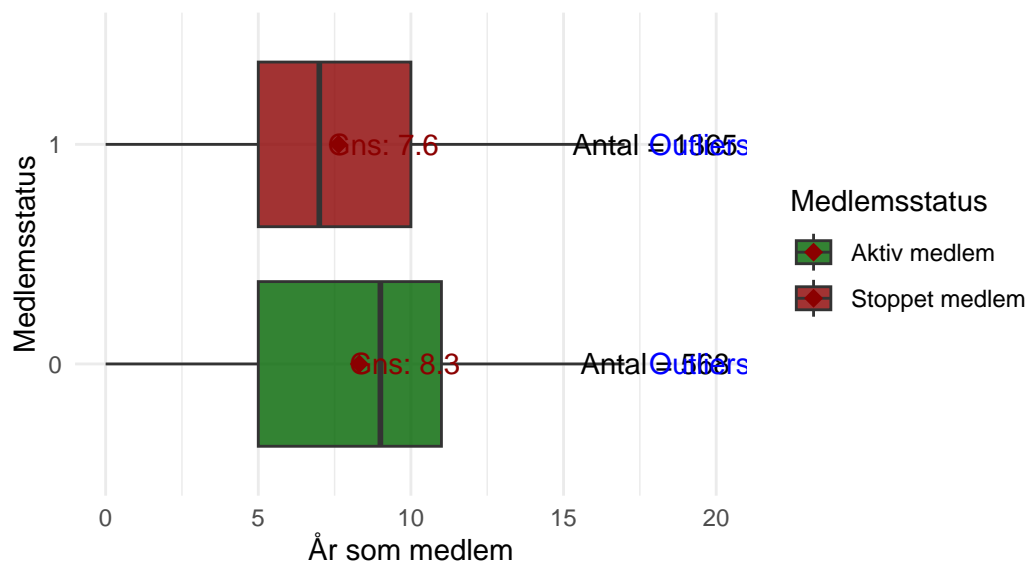
    outliers = sum(outlier_medlemsår, na.rm = TRUE),
    .groups = "drop"
)

featured |>
  ggplot(aes(x = factor(churn), y = medlem_antal_år, fill = factor(churn))) +
  geom_boxplot(alpha = 0.8, outlier.color = "blue") +
  stat_summary(fun = mean, geom = "point", shape = 18, size = 3, color = "darkred") +
  geom_text(data = nøgletal, aes(x = factor(churn), y = max(featured$medlem_antal_år, na.rm = TRUE),
                                label = paste0("Antal = ", antal)), inherit.aes = FALSE, size = 12, color = "black") +
  geom_text(data = nøgletal, aes(x = factor(churn), y = gennemsnit,
                                label = paste0("Gns: ", gennemsnit)), inherit.aes = FALSE, size = 12, color = "black") +
  geom_text(data = nøgletal, aes(x = factor(churn), y = max(featured$medlem_antal_år, na.rm = TRUE),
                                label = paste0("Outliers: ", outliers)), inherit.aes = FALSE, size = 12, color = "black") +
  scale_fill_manual(
    values = c("0" = "darkgreen", "1" = "darkred"),
    labels = c("0" = "Aktiv medlem", "1" = "Stoppet medlem"),
    name = "Medlemsstatus"
  ) +
  labs(
    title = "Antal år som medlem fordelt på medlemsstatus",
    subtitle = "Rød prik = gennemsnit • Blå prikker = outliers",
    x = "Medlemsstatus",
    y = "År som medlem"
  ) +
  coord_flip() +
  theme_minimal()

```

Antal år som medlem fordelt på medlemsstatus

Rød prik = gennemsnit . Blå prikker = outliers



```
ggsave("images/EDA_2_antal_år_medlem.png", width = 7, height = 4, dpi = 300)
```

```
# -----
# 5.3 Histogram: Fordeling af medlemsår
# -----

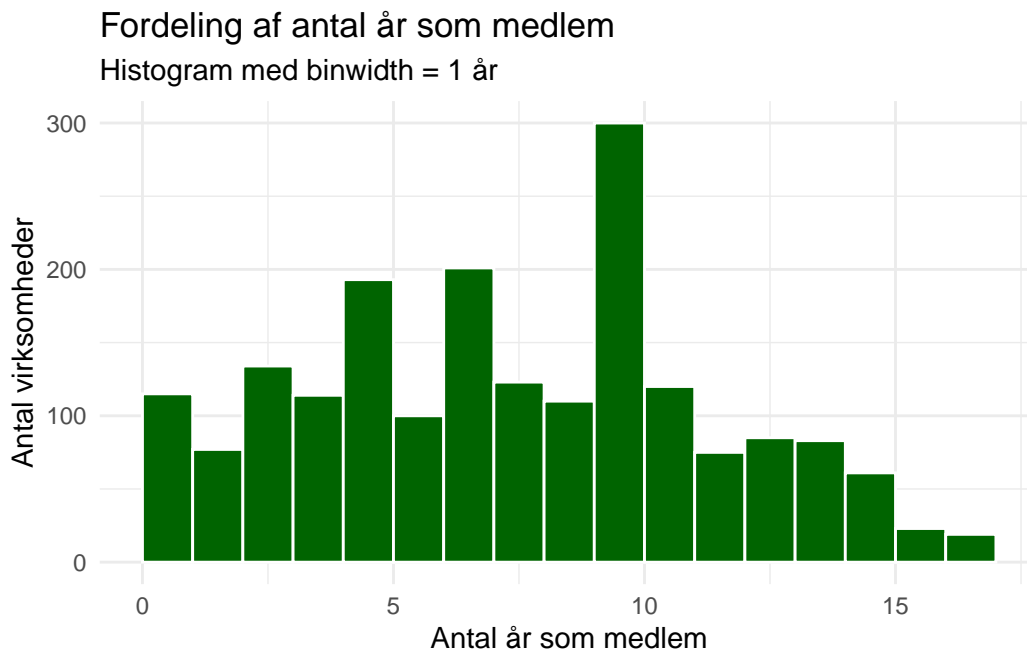
# Vi visualiserer fordelingen af hvor længe virksomheder har været medlem.
# Dette giver et billede af om der er mange nye vs. gamle medlemmer.

featured |>
  filter(!is.na(medlem_antal_år)) |>
  ggplot(aes(x = medlem_antal_år)) +
  geom_histogram(binwidth = 1, fill = "darkgreen", color = "white", boundary = 0) +
  labs(
    title = "Fordeling af antal år som medlem",
    subtitle = "Histogram med binwidth = 1 år",
  )
```

```

x = "Antal år som medlem",
y = "Antal virksomheder"
) +
theme_minimal()

```



```

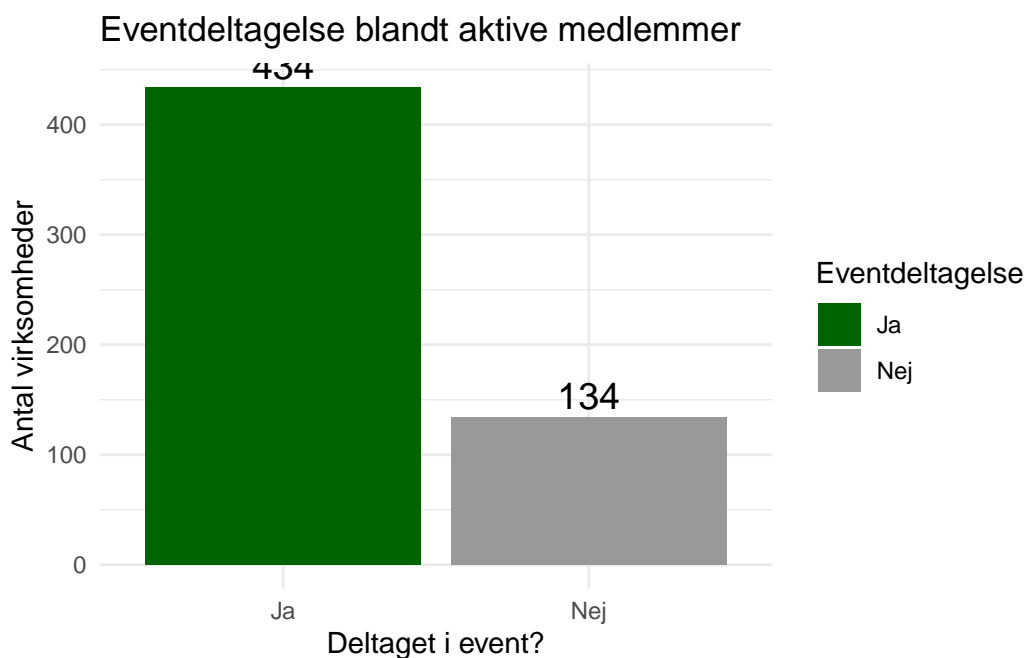
ggsave("images/EDA_3_fordeling_antal_år.png", width = 7, height = 4, dpi = 300)

# -----
# 5.4 Søjlediagram: Eventdeltagelse blandt aktive medlemmer
# -----
# Visualisering af eventdeltagelse blandt aktive medlemmer
# Her undersøger vi om virksomheder deltager i events, fordelt på 'Ja' og 'Nej'

featured |>
  filter(churn == 0) |>
  ggplot(aes(x = deltaget_i_event, fill = deltaget_i_event)) +

```

```
geom_bar() +
geom_text(stat = "count", aes(label = ..count..), vjust = -0.3, size = 5) +
scale_fill_manual(values = c("Ja" = "darkgreen", "Nej" = "grey60")) +
labs(
  title = "Eventdeltagelse blandt aktive medlemmer",
  x = "Deltaget i event?",
  y = "Antal virksomheder",
  fill = "Eventdeltagelse"
) +
theme_minimal()
```



```
ggsave("images/EDA_4_eventdeltagelse.png", width = 7, height = 4, dpi = 300)
```

```
# -----
# 5.5 Søjlediagram: Kombination af kontakt og eventdeltagelse vs. medlemsstatus
# -----
```



```

# Vi undersøger, hvordan kombinationen af kontakt og eventdeltagelse
# relaterer sig til medlemsstatus (aktiv eller stoppet).
# Derfor opretter vi en ny variabel "kontakt_event", der grupperer virksomheder
# efter disse kombinationer:
# - Både haft kontakt og deltaget i event
# - Kun haft kontakt
# - Kun deltaget i event
# - Ingen af delene

featured |>
  mutate(
    kontakt_event = case_when(
      har_haft_kontakt == "Ja" & deltaget_i_event == "Ja" ~ "Kontakt & Event",
      har_haft_kontakt == "Ja" & deltaget_i_event == "Nej" ~ "Kun kontakt",
      har_haft_kontakt == "Nej" & deltaget_i_event == "Ja" ~ "Kun event",
      TRUE ~ "Ingen af delene"
    )
  ) |>
  # Visualiser fordelingen med søjlediagram, hvor vi stabler medlemsstatus (churn)
  ggplot(aes(x = kontakt_event, fill = factor(churn))) +

  # geom_bar tæller antallet af observationer i hver kontakt_event-kategori,
  # og stabler dem efter medlemsstatus (aktiv = 0, stoppet = 1)
  geom_bar(position = "stack") +

  # Tilføj antals-labels direkte på søjlerne med hvid tekst
  geom_text(stat = "count", aes(label = ..count..),
            position = position_stack(vjust = 0.5), color = "white", size = 4) +

```

```

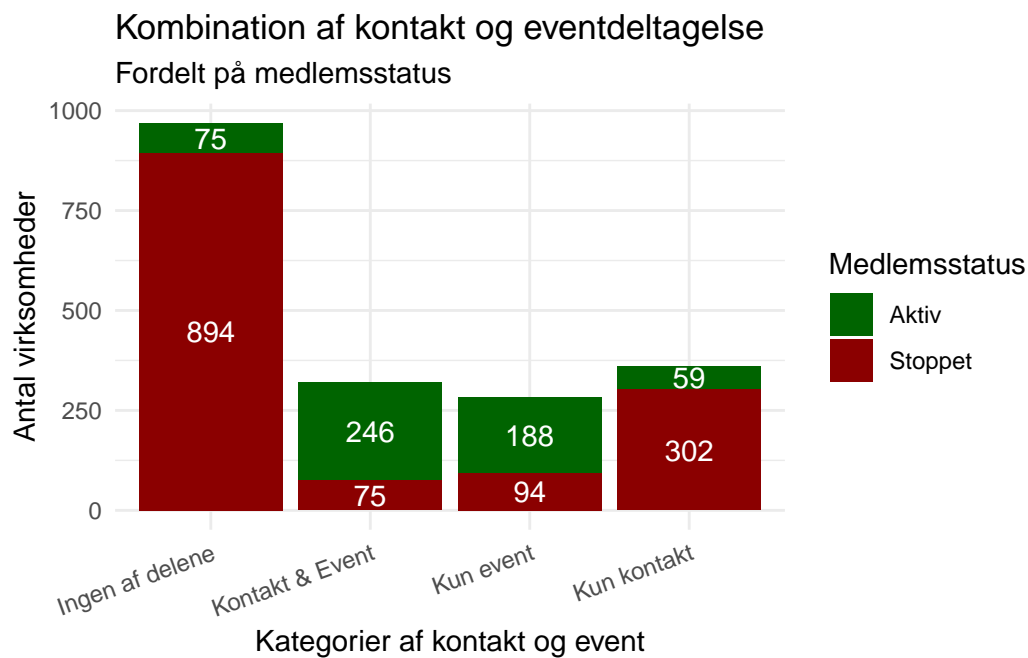
# Definér farver og labels for churn (medlemsstatus)
scale_fill_manual(
  values = c("0" = "darkgreen", "1" = "darkred"), # Grøn for aktiv, rød for stoppet
  labels = c("0" = "Aktiv", "1" = "Stoppet"),
  name = "Medlemsstatus"
) +

# Tilføj titler og akse-labels
labs(
  title = "Kombination af kontakt og eventdeltagelse",
  subtitle = "Fordelt på medlemsstatus",
  x = "Kategorier af kontakt og event",
  y = "Antal virksomheder"
) +

# Brug minimalistisk tema
theme_minimal() +

# Drej x-aksens tekst lidt for bedre læsbarhed
theme(axis.text.x = element_text(angle = 20, hjust = 1))

```



```
ggsave("images/EDA_5_kombination_kontakt_event.png", width = 7, height = 4, dpi = 300)
```

```
# -----
```

```
# 5.6 Søjlediagram: Hjælpekategori blandt aktive deltagere vs. ikke deltagere
```

```
# -----
```

```
# Vi undersøger hvilke typer hjælp virksomheder har fået.
```

```
# Det opdeles i to grupper:
```

```
# - Aktive medlemmer, der IKKE har deltaget i events
```

```
# - Aktive medlemmer, der HAR deltaget i events
```

```
# På den måde kan vi sammenligne hvilke behov de to grupper har.
```

```
# Ikke-deltagere i event - hvilke typer hjælp efterspørger de?
```

```
# -----
```

```
featured |>
```

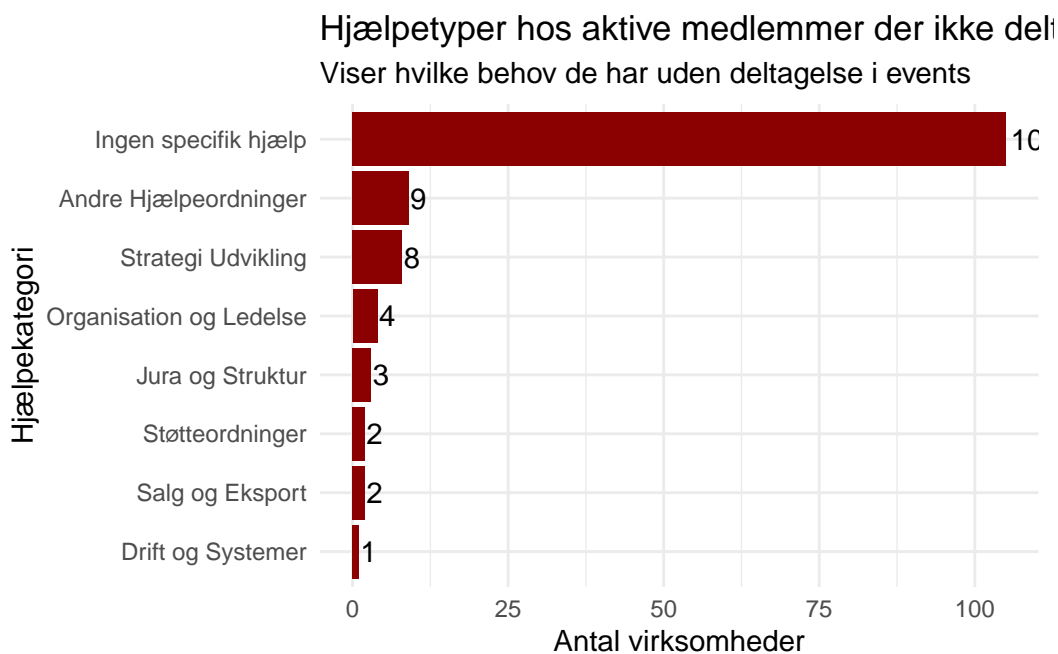
```
  filter(churn == 0, deltaget_i_event == "Nej") |>
```

```
  # Kun aktive, ikke-deltagere
```

```

count(hjælp_kategori, name = "antal") |> # Tæl antal pr. hjælpekategori
filter(!is.na(hjælp_kategori)) |> # Fjern NA
ggplot(aes(x = fct_reorder(hjælp_kategori, antal), y = antal)) +
geom_col(fill = "darkred") + # Farve: Rød for ikke-de
geom_text(aes(label = antal), hjust = -0.1, size = 4) +
coord_flip() + # Vandret søjlediagram
labs(
  title = "Hjælpetyper hos aktive medlemmer der ikke deltager i events",
  subtitle = "Viser hvilke behov de har uden deltagelse i events",
  x = "Hjælpekategori",
  y = "Antal virksomheder"
) +
theme_minimal()

```



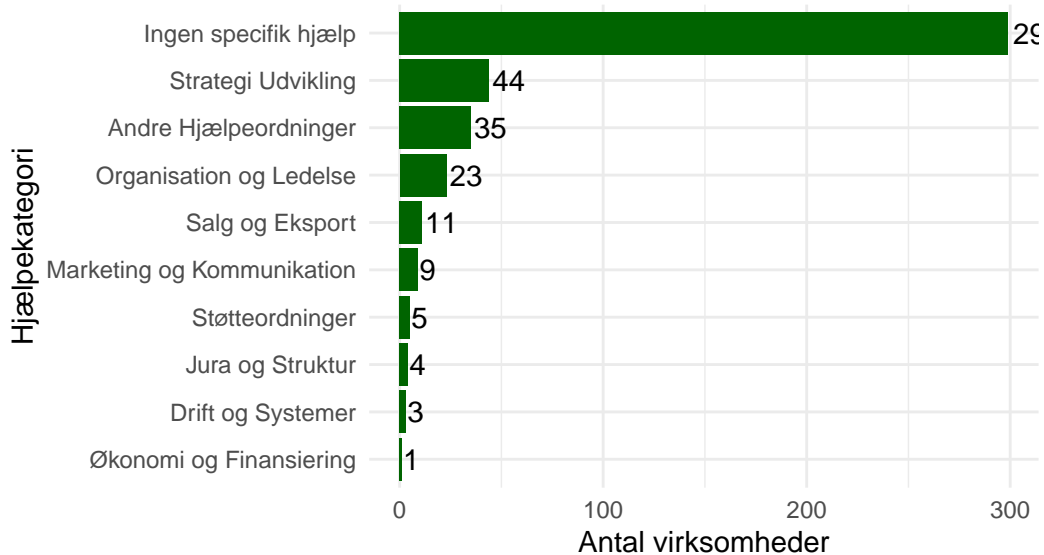
```

ggsave("images/EDA_6_deltager_ikke_events.png", width = 7, height = 4, dpi = 300)

# Deltagere i event - hvilke typer hjælp efterspørger de?
# -----
featured |>
  filter(churn == 0, deltaget_i_event == "Ja") |> # Kun aktive, deltagere
  count(hjælp_kategori, name = "antal") |>
  filter(!is.na(hjælp_kategori)) |>
  ggplot(aes(x = fct_reorder(hjælp_kategori, antal), y = antal)) +
  geom_col(fill = "darkgreen") + # Farve: grøn for deltagere
  geom_text(aes(label = antal), hjust = -0.1, size = 4) +
  coord_flip() +
  labs(
    title = "Hjælpetyper hos aktive medlemmer der deltager i events",
    subtitle = "Viser hvilke behov de har, trods deltagelse i events",
    x = "Hjælpekategori",
    y = "Antal virksomheder"
  ) +
  theme_minimal()

```

Hjælpetyper hos aktive medlemmer der delta
Viser hvilke behov de har, trods deltagelse i events



```
ggsave("images/EDA_6_1_deltager_i_events.png", width = 7, height = 4, dpi = 300)
```

```
# -----
# 5.7 Søjlediagram: Branchefordeling blandt aktive medlemmer
# -----
```

```
# Vi undersøger hvilke brancher de *aktive* virksomheder (medlemmer) tilhører.
# Formålet er at få et overblik over, hvilke brancher der er mest repræsenteret
# blandt dem, der stadig er med i fællesskabet (churn == 0).
# Kun brancher med mindst 10 virksomheder vises, for at sikre et læsbart plot.
```

```
featured |>
  # Filtrér: medtag kun virksomheder der stadig er medlemmer (aktive)
  filter(churn == 0) |>

  # Tæl antallet af virksomheder pr. branche
```

```

count(Branche_navn, name = "antal") |>

# Fjern brancher med færre end 10 aktive virksomheder
filter(antal >= 10) |>

# Sortér brancherne efter antal, så de vises i rigtig rækkefølge i plottet
mutate(Branche_navn = fct_reorder(Branche_navn, antal)) |>

# Visualiser fordelingen med søjlediagram
ggplot(aes(x = Branche_navn, y = antal)) +

# geom_col bruger vores forudberegnete 'antal' til at tegne søjler
geom_col(fill = "darkgreen") +

# Tilføj antals-labels til søjlerne (antal virksomheder)
geom_text(aes(label = antal), hjust = -0.1, size = 4) +

# Vend koordinaterne, så brancherne vises lodret og er nemmere at læse
coord_flip() +

# Tilføj titel, undertitel og aksetekster
labs(
  title = "Branchefordeling blandt aktive medlemmer",
  subtitle = "Viser kun brancher med mindst 10 aktive virksomheder",
  x = "Branche",
  y = "Antal virksomheder"
) +

```

```
# Brug minimalistisk tema for et rent visuelt udtryk
theme_minimal()
```

```

Engroshandel undtagen med motorkøretøjer og motorcykle
Bygge- og anlægsvirksomhed, som kræver specialiserin
Hovedsæders virksomhed; virksomhedsrådgivnin
Pengeinstitut- og finansieringsvirksomhed undtagen forsikring o
Arkitekt- og ingeniørvirksomhed; teknisk afprøvning og analys
Detailhandel undtagen med motorkøretøjer og motorcykle
Undervisnin
Fast ejendor
Juridisk bistand, bogføring og revisio
Reklame og markedsanalys
Computerprogrammering, konsulentbistand vedrørende informationsteknologi og lignende aktivitet
Andre liberale, videnskabelige og tekniske tjenesteydelse
Jern- og metalvareindustri, undtagen maskiner og udsty
Sundhedsvæse
Sport, forlystelser og fritidsaktivitete
Fremstilling af maskiner og udstyr i.a.r
Organisationer og foreninge
Administrationsservice, kontorservice og anden forretnings servic

```

Antal vi

```
ggsave("images/EDA_7_branchefordeling.png", width = 7, height = 4, dpi = 300)
```

```
# -----
# 5.8 Søjlediagram: Eventdeltagelse fordelt på branche (kun aktive medlemmer)
# -----
```

```

# Vi undersøger hvordan virksomhedernes deltagelse i events varierer
# på tværs af brancher.
# Kun aktive virksomheder (churn == 0) tages med i analysen.
# Vi vil gerne se, hvor mange fra hver branche der har deltaget /
# ikke deltaget i events.

```

```
brancher_antal <- featured |>
```



```

# Filtrér: medtag kun aktive virksomheder
filter(churn == 0) |>

# Tæl antallet af virksomheder for hver kombination af branche og eventdeltagelse
count(Branche_navn, deltaget_i_event, name = "antal") |>

# Beregn totalen pr. branche (summen af 'Ja' og 'Nej')
group_by(Branche_navn) |>
mutate(total = sum(antal)) |>

# Behold kun brancher med mindst 10 aktive virksomheder i alt
filter(total >= 10) |>
ungroup()

# Sortér brancher efter hvor mange virksomheder der har deltaget i events ("Ja")
sortering <- brancher_antal |>
  filter(deltaget_i_event == "Ja") |>
  arrange(desc(antal)) |>
  pull(Branche_navn)

brancher_antal |>
  # Sortér brancherne i plottet ud fra antallet af 'Ja'-svar
  mutate(Branche_navn = factor(Branche_navn, levels = unique(sortering))) |>

  # Opret stacked søjlediagram, hvor 'Ja' og 'Nej' ligger oven på hinanden
  ggplot(aes(x = Branche_navn, y = antal, fill = deltaget_i_event)) +

  # geom_col tegner søjlerne baseret på vores antal

```

```

geom_col(position = "stack") +

# Tilføj tekstetiketter med antal, centreret i hver del af søjlen
geom_text(aes(label = antal), position = position_stack(vjust = 0.5),
          color = "white", size = 4) +

# Vend akserne, så brancherne læses nemmere
coord_flip() +

# Brug manuelle farver: grøn = deltaget, grå = ikke deltaget
scale_fill_manual(values = c("Ja" = "darkgreen", "Nej" = "grey60")) +

# Tilføj titler og akse-labels
labs(
  title = "Eventdeltagelse fordelt på branche",
  subtitle = "Kun aktive medlemmer med min. 10 virksomheder per branche",
  x = "Branche",
  y = "Antal virksomheder",
  fill = "Eventdeltagelse"
) +

# Brug minimalistisk ggplot-tema
theme_minimal()

```

Kun akti

Even

```
# -----
# 5.9 Søjlediagram: Hjælpekategorier fordelt på branche (kun aktive medlemmer)
# -----
```

```
featured |>
```

```
filter(churn == 0, !is.na(hjælp kategori)) |>
```

Tæl antallet af hjælperegistreringer pr. branche og hjælpekategori

```

count(Branche_navn, hjælp_kategori, name = "antal") |>

# Beregn total antal hjælperegistreringer pr. branche
group_by(Branche_navn) |>
mutate(total = sum(antal)) |>
ungroup() |>

# Fjern brancher med under 5 hjælperegistreringer
filter(total >= 5) |>

# Sortér brancherne efter total, så de vises i rigtig rækkefølge i plottet
mutate(Branche_navn = fct_reorder(Branche_navn, total)) |>

# Opret søjlediagram med hjælpekategori som fill (farvelag)
ggplot(aes(x = Branche_navn, y = antal, fill = hjælp_kategori)) +

# geom_col tegner de stablede søjler (én farve pr. hjælpekategori)
geom_col() +

# Vend koordinatsystemet, så brancherne er på y-aksen (bedre læsbarhed)
coord_flip() +

# Tilføj titler og aksetekster
labs(
  title = "Hjælpekategorier fordelt på branche",
  subtitle = "Kun aktive medlemmer med mindst 5 registrerede hjælpetilfælde",
  x = "Branche",
  y = "Antal hjælperegistreringer",

```

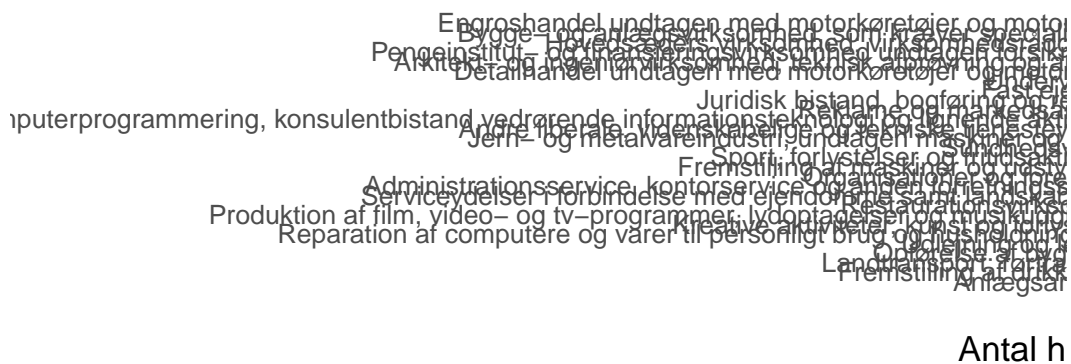
```

    fill = "Hjælpekategori"
) +

# Brug minimalistisk ggplot-tema og lidt større tekst
theme_minimal(base_size = 13) +

# Flyt legenden ned under plottet (bedre ved mange kategorier)
theme(legend.position = "bottom")

```



Hjælpekategori

■ Andre Hjælpeordninger	■ Ingen specifik h
■ Drift og Systemer	■ Jura og Strukturi

```

ggsave("images/EDA_9_hjælpekategorier_branche.png", width = 7, height = 4, dpi = 300)

# -----
# 5.10 Søjlediagram: Eventdeltagelse pr. postnummer (Aktive medlemmer)
# -----

# Vi undersøger, hvordan aktive virksomheders eventdeltagelse fordeler sig geografisk,

```

```

# baseret på postnummer. Vi viser både deltagelse ("Ja") og ikke-deltagelse ("Nej"),
# og sorterer postnumrene efter hvor stor andelen af 'Ja'-svar er.

# Beregn antal og andel for hver kombination af postnummer og eventdeltagelse
post_event <- featured |>

# Filtrér: medtag kun aktive virksomheder
filter(churn == 0) |>

# Tæl antallet af virksomheder pr. postnummer og eventstatus
count(PostalCode, deltaget_i_event, name = "antal") |>

# Beregn total antal og procentuel andel inden for hvert postnummer
group_by(PostalCode) |>
mutate(
  total = sum(antal),                # Total antal virksomheder
  andel = round(antal / total * 100, 1) # Andel i procent
) |>
ungroup()

# Sortér postnumre efter andel af virksomheder der har deltaget i events ("Ja")
post_order <- post_event |>
  filter(deltaget_i_event == "Ja") |>
  arrange(desc(andel)) |>
  pull(PostalCode)

# Visualiser data som stacked søjlediagram
post_event |>
  # Sortér postnumrene i plottet efter andel 'Ja'

```

```

mutate(PostalCode = factor(PostalCode, levels = post_order)) |>

# Opret plot med antal virksomheder pr. postnummer, farvet efter eventdeltagelse
ggplot(aes(x = PostalCode, y = antal, fill = deltaget_i_event)) +

# geom_col tegner stablede søjler
geom_col() +

# Tilføj antalslabels midt i søjlerne
geom_text(aes(label = antal), position = position_stack(vjust = 0.5),
          color = "white", size = 3.5) +

# Brug manuelle farver: grøn = deltager, grå = deltager ikke
scale_fill_manual(values = c("Ja" = "darkgreen", "Nej" = "grey60")) +

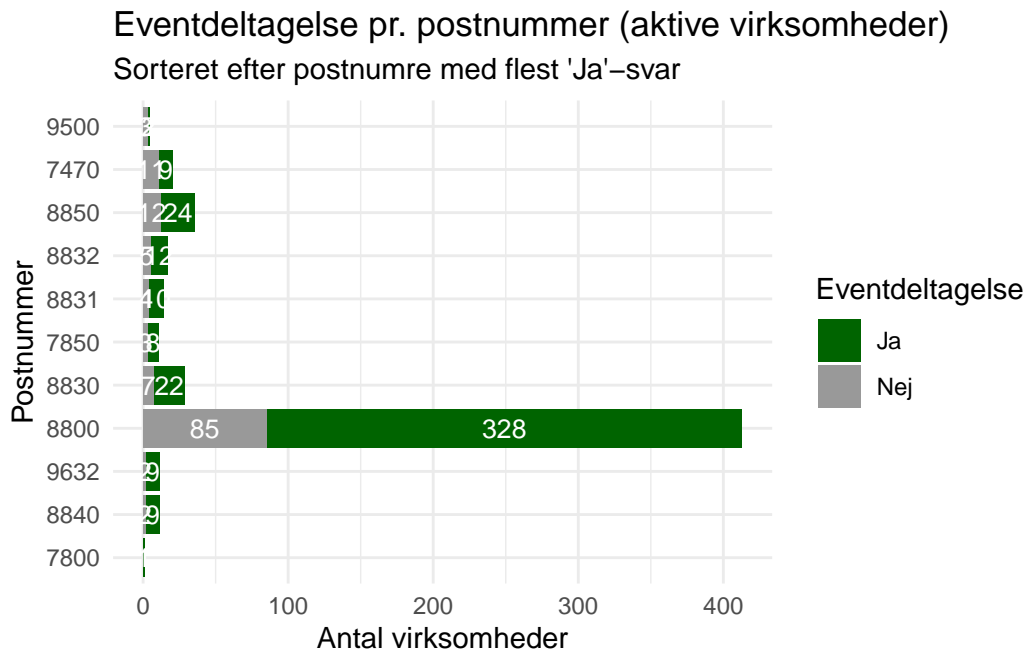
# Vend koordinatsystemet for bedre læsbarhed
coord_flip() +

# Tilføj titel, aksetitler og farveforklaring
labs(
  title = "Eventdeltagelse pr. postnummer (aktive virksomheder)",
  subtitle = "Sorteret efter postnumre med flest 'Ja'-svar",
  x = "Postnummer",
  y = "Antal virksomheder",
  fill = "Eventdeltagelse"
) +

# Brug minimalistisk ggplot-tema

```

```
theme_minimal()
```



```
ggsave("images/EDA_10_eventdeltagelse_postnummer.png", width = 7, height = 4, dpi = 300)
```

```
# -----  
# 5.11 Søjlediagram: Gennemsnitlig mødelængde pr. branche (kun aktive medlemmer)  
# -----
```

```
# Vi undersøger hvor lang tid møderne i gennemsnit varer for hver branche,  
# men ser kun på aktive medlemmer og brancher med tilstrækkeligt datagrundlag.  
# Det giver indsigt i, hvilke brancher der fx kræver mere sparring eller støtte.
```

```
featured |>
```

```
# Filtrér: medtag kun aktive medlemmer med gyldig mødelængde (> 0 og ikke NA)
```

```
filter(churn == 0, !is.na(MeetingLength), MeetingLength > 0) |>
```



```

# Beregn gennemsnitlig mødelængde og antal møder pr. branche
group_by(Branche_navn) |>
summarise(
  gennemsnit_min = round(mean(MeetingLength, na.rm = TRUE), 1), # afrundet gennemsnit
  antal = n(), # antal møder
  .groups = "drop"
) |>

# Behold kun brancher med mindst 5 registrerede møder
filter(antal >= 1) |>

# Sortér brancherne efter gennemsnitlig mødelængde (fra højest til lavest)
arrange(desc(gennemsnit_min)) |>

# Reordn faktorniveauerne så sorteringen overføres til plot
mutate(Branche_navn = fct_reorder(Branche_navn, gennemsnit_min)) |>

# Visualiser data med et søjlediagram
ggplot(aes(x = Branche_navn, y = gennemsnit_min)) +

# geom_col viser gennemsnitlig mødelængde pr. branche
geom_col(fill = "darkgreen", alpha = 0.8) +

# Tilføj tekstetiketter med mødelængden på hver søjle
geom_text(aes(label = paste0(gennemsnit_min, " min")), hjust = -0.1, size = 4) +

# Vend koordinaterne så brancherne vises lodret og er nemmere at læse
coord_flip() +

```

```
# Tilføj titler, undertitel og akse-labels
labs(
  title = "Gennemsnitlig mødelængde pr. branche (aktive medlemmer)",
  subtitle = "Viser kun brancher med mindst 5 registrerede møder",
  x = "Branche",
  y = "Gennemsnitlig mødelængde (minutter)"
) +

# Brug minimalistisk ggplot-tema og lidt større base-tekst
theme_minimal(base_size = 13)
```

Telekommunikation
Fremstilling af elektrisk udstyr
Reparation af computere og varer til personligt brug
Administrationsservice, kontorservice og anden forretnings-
Arkitekt- og ingeniørvirksomhed; tekniske konsulting og
Engros- og detailhandel undtagen med motorer, motorer, skibe
Trykning, sport, rekreative aktiviteter og sport
Fremstilling af gamle og gamle
Detailhandel undtagen med motorer, motorer, skibe
Serviceydelser i forbindelse med ejendomme samt landbr
Bygge- og anlægsvirksomhed undtagen bygge- og anlægsvirksomhed
Jern- og metalvirksomhed undtagen bygge- og anlægsvirksomhed
Hovedsædels virksomhed undtagen bygge- og anlægsvirksomhed
Kreative aktiviteter, kunst og håndværk
Hjælpevirksomhed i forbindelse med
Fremstilling af motorer, motorer, skibe og andre
Fremstilling af andre ikke-metalliske mineraler
Opsamlings- og behandling af affald
Landtransport
Lufttransport
Havtransport
Andre transportmidler
Andre aktiviteter
Andre serviceydelser
Andre detailhandel
Andre engros- og detailhandel
Andre bygge- og anlægsvirksomhed
Andre jern- og metalvirksomhed
Andre fremstilling af elektrisk udstyr
Andre fremstilling af mekaniske udstyr
Andre fremstilling af tekstil og tekstilvarer
Andre fremstilling af læder og lædervarer
Andre fremstilling af træ og trævarer
Andre fremstilling af papir og papirvarer
Andre fremstilling af glas og glasvarer
Andre fremstilling af keramik og keramikvarer
Andre fremstilling af stål og stålvarer
Andre fremstilling af jern og jernvarer
Andre fremstilling af kobber og kobbervarer
Andre fremstilling af aluminium og aluminiumvarer
Andre fremstilling af plast og plastvarer
Andre fremstilling af gummi og gummivarer

Gennemsnitlig

```
ggsave("images/EDA_11_mødelængde.png", width = 7, height = 4, dpi = 300)
```

```
# -----
# 5.12 Boxplot: Mødelængde vs. eventdeltagelse (kun aktive medlemmer)
```

```

# -----

# Vi undersøger, om der er forskel i mødelængde blandt aktive virksomheder
# afhængigt af om de har deltaget i events eller ej.
# Vi visualiserer forskellen med boxplots og markerer både gennemsnit og outliers.

# Gør datasættet klar: filtrér aktive, beregn outliers, og opret grupper
meeting_event <- featured |>
  # Filtrér kun aktive virksomheder med mødelængde registreret
  filter(churn == 0, !is.na(MeetingLength)) |>

  # Opret en ny variabel "Eventgruppe" baseret på om de har deltaget i events
  mutate(
    Eventgruppe = if_else(deltaget_i_event == "Ja", "Deltager i event", "Deltager ikke"),

    # Identificer outliers i mødelængde med IQR-metoden
    outlier = find_outliers(MeetingLength)
  )

# Udtræk statistik til annotationer i boxplottet
meeting_stats <- meeting_event |>
  group_by(Eventgruppe) |>
  summarise(
    antal = n(), # Antal observationer pr. gruppe
    gennemsnit = round(mean(MeetingLength), 1), # Gennemsnitlig mødelængde
    outliers = sum(outlier), # Antal outliers
    .groups = "drop"
  )

```

```

# Visualisér forskellen med boxplots
meeting_event |>
  ggplot(aes(x = Eventgruppe, y = MeetingLength, fill = Eventgruppe)) +

  # Boxplot viser fordeling, median og outliers
  geom_boxplot(alpha = 0.7, outlier.color = "blue") +

  # Tilføj en rød prik for gennemsnittet i hver gruppe
  stat_summary(fun = mean, geom = "point", shape = 18, size = 3, color = "darkred") +

  # Annotér med antal virksomheder pr. gruppe (øverst over boxplot)
  geom_text(data = meeting_stats, aes(x = Eventgruppe,
                                       y = max(meeting_event$MeetingLength, na.rm = TRUE) +
                                       label = paste0("n = ", antal)), size = 4) +

  # Annotér med gennemsnitlig mødelængde
  geom_text(data = meeting_stats, aes(x = Eventgruppe, y = gennemsnit,
                                       label = paste0("Gns: ", gennemsnit, " min")),
            size = 4, color = "darkred", nudge_y = 5) +

  # Vend koordinatsystemet, så grupperne vises lodret
  coord_flip() +

  # Brug manuelle farver til at skelne mellem deltager/ikke-deltager
  scale_fill_manual(values = c("Deltager i event" = "darkgreen", "Deltager ikke" = "red"))

  # Tilføj titler, aksetitel og undertekst
  labs(

```

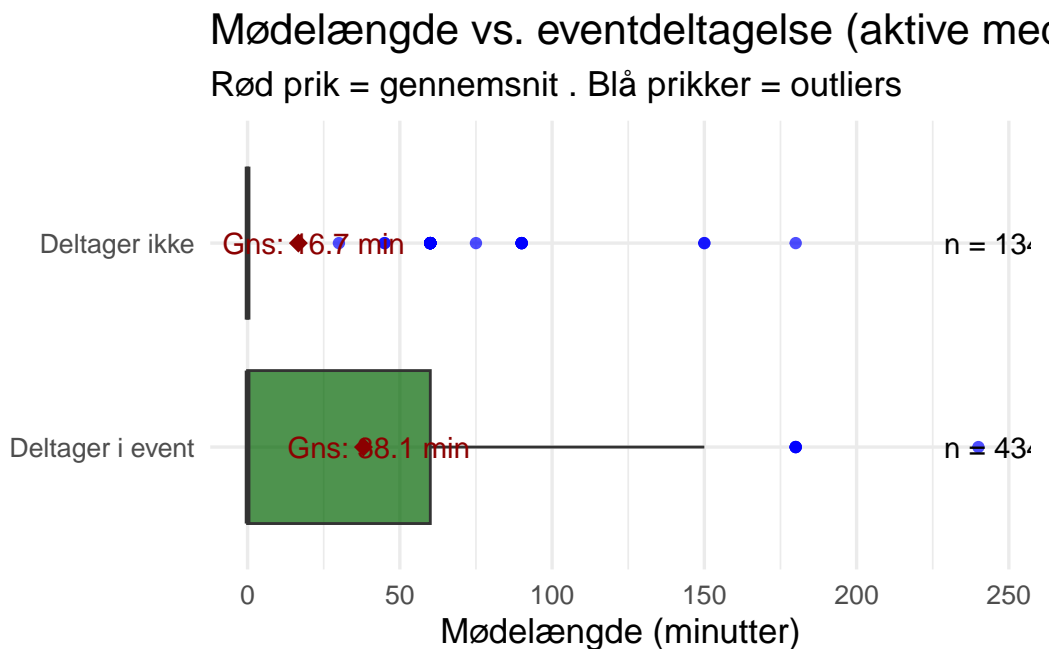
```

title = "Mødelængde vs. eventdeltagelse (aktive medlemmer)",
subtitle = "Rød prik = gennemsnit • Blå prikker = outliers",
x = NULL,                                # Ingen x-aksenavn (grupperne forklarer sig selv)
y = "Mødelængde (minutter)"
) +

# Brug minimalistisk tema med større font
theme_minimal(base_size = 13) +

# Fjern legend, da den ikke er nødvendig - informationen er allerede tydelig i x-aksen
theme(legend.position = "none")

```



```

ggsave("images/EDA_12_mødelængde_eventdeltagelse.png", width = 7, height = 4, dpi = 300)

# -----
# 5.13 Boxplot: Mødelængde vs. medlemsstatus

```

```

# -----

# Vi undersøger, om der er forskel i mødelængde mellem aktive og stoppede medlemmer.
# Dette kan give indsigt i, om engagement (målt som mødetid) har betydning for churn.

# Udregn nøgleoplysninger til visning i plottet
meeting_churn_stats <- featured |>
  filter(!is.na(MeetingLength)) |>
  group_by(churn) |>
  summarise(
    n = n(), # Antal observationer
    gennemsnit = round(mean(MeetingLength, na.rm = TRUE), 1), # Gennemsnitlig mødelængde
    outliers = sum(find_outliers(MeetingLength)), # Antal outliers
    .groups = "drop"
  ) |>
  mutate(churn = factor(churn, levels = c(0, 1), labels = c("Aktiv", "Stoppet")))

# Opret boxplot med medlemsstatus som grupperingsvariabel
featured |>
  filter(!is.na(MeetingLength)) |>
  mutate(
    churn = factor(churn, levels = c(0, 1), labels = c("Aktiv", "Stoppet")),
    outlier = find_outliers(MeetingLength)
  ) |>
  ggplot(aes(x = churn, y = MeetingLength, fill = churn)) +

# Vis fordeling og outliers som boxplot
geom_boxplot(alpha = 0.7, outlier.shape = 16, outlier.color = "blue", outlier.size = 2)

```

```

# Marker gennemsnittet med en rød prik
stat_summary(fun = mean, geom = "point", color = "darkred", size = 3, shape = 18) +

# Tilføj tekst med gennemsnitlig mødelængde for hver gruppe
geom_text(data = meeting_churn_stats, aes(x = churn, y = gennemsnit,
                                           label = paste0("Gns: ", gennemsnit, " min")),
          color = "darkred", size = 4, nudge_y = 5, inherit.aes = FALSE) +

# Vis antal observationer som tekst over hver boks
geom_text(data = meeting_churn_stats, aes(x = churn, y = max(featured$MeetingLength, na.rm = TRUE),
                                           label = paste0("n = ", n)),
          color = "black", size = 4, inherit.aes = FALSE) +

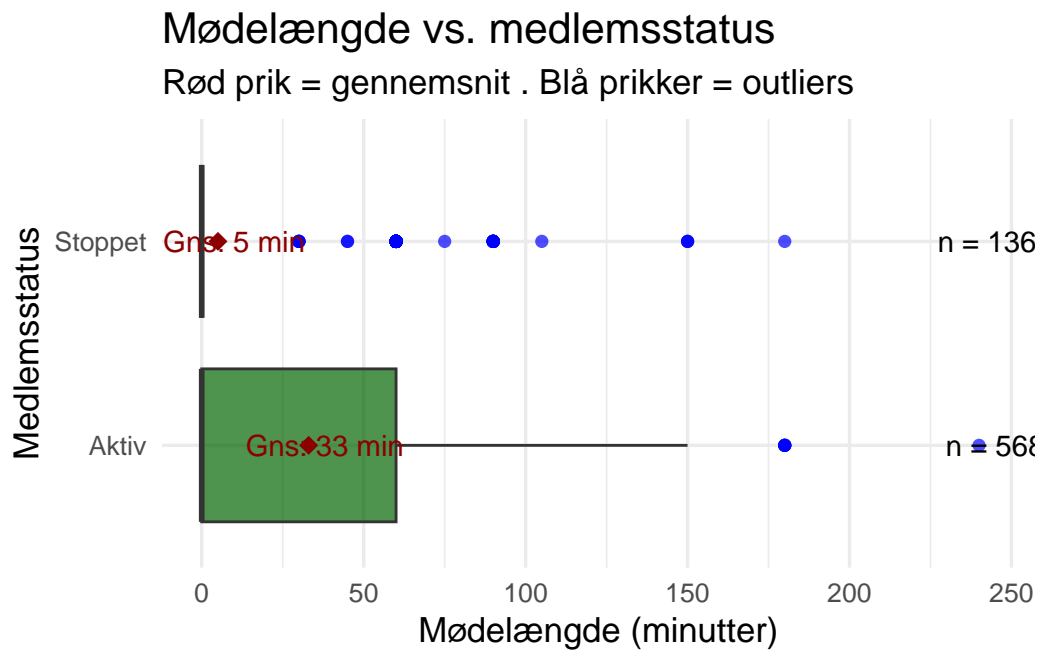
# Vend akserne, så medlemsstatus vises lodret
coord_flip() +

# Brug farver der skelner aktiv og stoppet medlemsstatus
scale_fill_manual(values = c("Aktiv" = "darkgreen", "Stoppet" = "darkred")) +

# Tilføj titler og aksetekster
labs(
  title = "Mødelængde vs. medlemsstatus",
  subtitle = "Rød prik = gennemsnit • Blå prikker = outliers",
  x = "Medlemsstatus",
  y = "Mødelængde (minutter)",
  fill = "Status"
) +

```

```
# Brug minimalistisk tema og fjern legend
theme_minimal(base_size = 13) +
theme(legend.position = "none")
```



```
ggsave("images/EDA_13_mødelængde_medlemsstatus.png", width = 7, height = 4, dpi = 300)

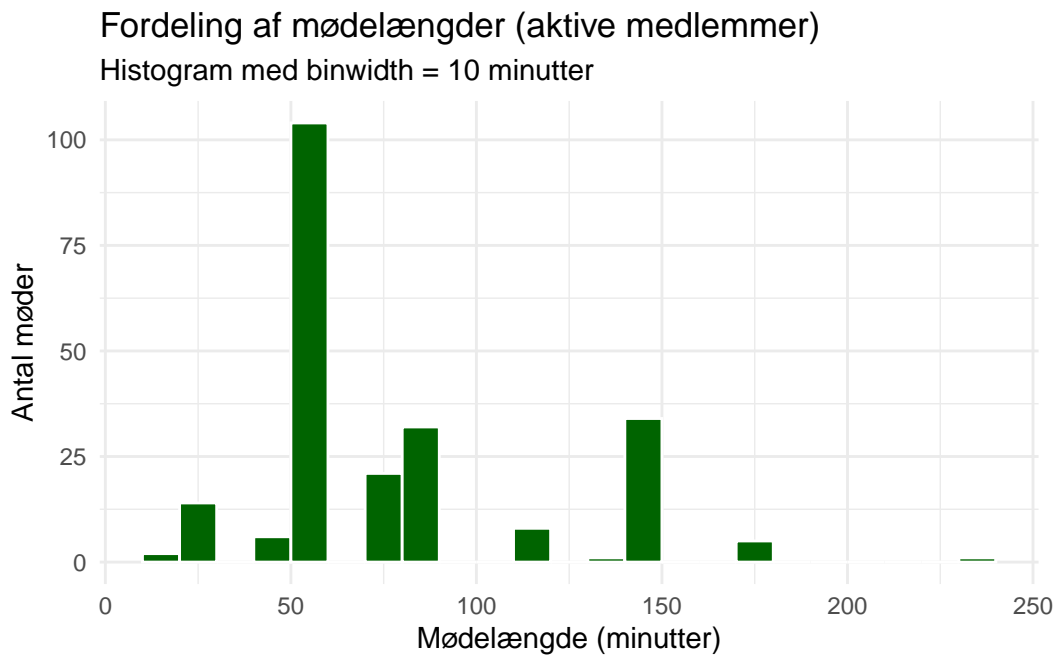
# -----
# 5.14 Histogram: Fordeling af mødelængder
# -----

# Vi undersøger hvordan mødelængder fordeler sig blandt aktive medlemmer,
# for at se om der er typiske længder, outliers eller skævhed.

featured |>
  filter(churn == 0, !is.na(MeetingLength), MeetingLength > 0) |>
  ggplot(aes(x = MeetingLength)) +
```



```
geom_histogram(binwidth = 10, fill = "darkgreen", color = "white", boundary = 0) +
labs(
  title = "Fordeling af mødelængder (aktive medlemmer)",
  subtitle = "Histogram med binwidth = 10 minutter",
  x = "Mødelængde (minutter)",
  y = "Antal møder"
) +
theme_minimal()
```



```
ggsave("images/EDA_14_fordeling_mødelængder.png", width = 7, height = 4, dpi = 300)
```

```
# -----
# 5.15 Korrelationsmatrix: Numeriske variable
# -----

# Vi ønsker at undersøge, hvordan de numeriske variable i datasættet hænger sammen.
```

```

# Det gør vi ved at udtrække alle numeriske kolonner og beregne en korrelationsmatrix.
# Denne visualiseres som et cirkelplot med Pearson-korrelationer.

# Udtræk kun de kolonner i datasættet, der er numeriske
featured_numerisk <- featured |>
  select(where(is.numeric)) |>

# Fjern rækker med NA-værdier, da korrelationsberegning kræver komplette værdier
drop_na()

# Beregn korrelationsmatrix ved hjælp af Pearson's metode
cor_matrix <- cor(featured_numerisk, use = "pairwise.complete.obs")

# Afrund korrelationerne til 2 decimaler for pænere visning (valgfrit trin)
cor_matrix_rounded <- round(cor_matrix, 2)

# Visualisér korrelationerne med et "cirkelplot" fra pakken 'corrplot'
corrplot::corrplot(
  cor_matrix,

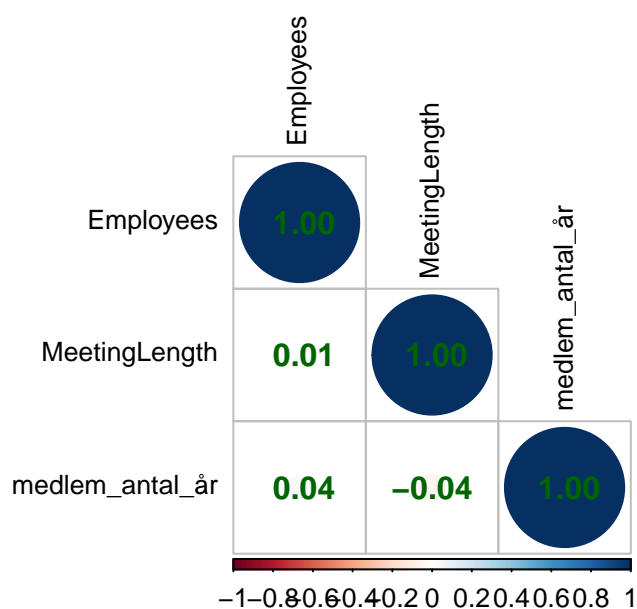
  method = "circle",      # Brug cirkler til at vise styrke og retning af korrelation
  type = "lower",         # Vis kun nederste trekant (mere overskueligt)

  # Tekst og talindstillinger
  tl.cex = 0.8,           # Størrelse på tekstetiketter (variabelnavne)
  tl.col = "black",       # Farve på variabelnavne

  addCoef.col = "darkgreen" # Vis selve korrelationstallet inde i cirklerne

```

)



```
ggsave("images/EDA_15_korrelationsmatrix.png", width = 7, height = 4, dpi = 300)
```

```
# -----  
# 6. Preprocessing  
# -----  
  
set.seed(2025)  
  
churn_split <- initial_split(feature_engineering, prop = 0.8, strata = churn)  
churn_train <- training(churn_split)  
churn_test  <- testing(churn_split)  
  
churn_folds <- vfold_cv(churn_train, v = 10, strata = churn)  
  
churn_recipe <-
```

```

recipe(churn ~ ., data = churn_train) |>
step_novel(all_nominal_predictors()) |>
step_dummy(all_nominal_predictors(), one_hot = TRUE) |>
step_zv(all_predictors()) |>
step_normalize(all_numeric_predictors()) |>
step_downsample(churn) # Brug evt. step_smote(churn) hvis ekstrem ubalance

```

```

# -----
# 7. Modelling
# -----

# Model specs

rf_spec <- rand_forest(mtry = tune(), min_n = tune()) |>
  set_engine("ranger", importance = "impurity") |>
  set_mode("classification")

xgb_spec <- boost_tree(trees = tune(), mtry = tune(), learn_rate = tune()) |>
  set_engine("xgboost") |>
  set_mode("classification")

log_reg_spec <- logistic_reg(penalty = tune(), mixture = tune()) |>
  set_engine("glmnet") |>
  set_mode("classification")

knn_spec <- nearest_neighbor(neighbors = tune(), weight_func = tune()) |>
  set_engine("kknn") |>
  set_mode("classification")

```

```

nb_spec <- naive_Bayes(smoothness = tune(), Laplace = tune()) |>
  set_engine("naivebayes") |>
  set_mode("classification")

svm_spec <- svm_rbf(cost = tune(), rbf_sigma = tune()) |>
  set_engine("kernlab") |>
  set_mode("classification")

# Samlet workflow set
churn_workflow_set <- workflow_set(
  preproc = list(churn_recipe = churn_recipe),
  models = list(
    rf = rf_spec,
    xgboost = xgb_spec,
    logistic = log_reg_spec,
    knn = knn_spec,
    naive_bayes = nb_spec,
    svm_rbf = svm_spec
  )
)

```

```

# -----
# 8. Evaluate metrics
# -----

churn_metrics <- metric_set(accuracy, roc_auc, f_meas, sens, spec)

grid_ctrl <- control_grid(
  verbose = TRUE,

```

```
  save_pred = TRUE,  
  parallel_over = "everything",  
  save_workflow = TRUE  
)
```

```
plan(multisession)  
strt.time <- Sys.time()
```

```
# Vi kører modellerne - den står og arbejder  
churn_results <- churn_workflow_set |>  
  workflow_map(  
    resamples = churn_folds,  
    grid = 5,  
    metrics = churn_metrics,  
    control = grid_ctrl,  
    seed = 2025  
  )
```

```
Sys.time() - strt.time
```

Time difference of 1.030561 mins

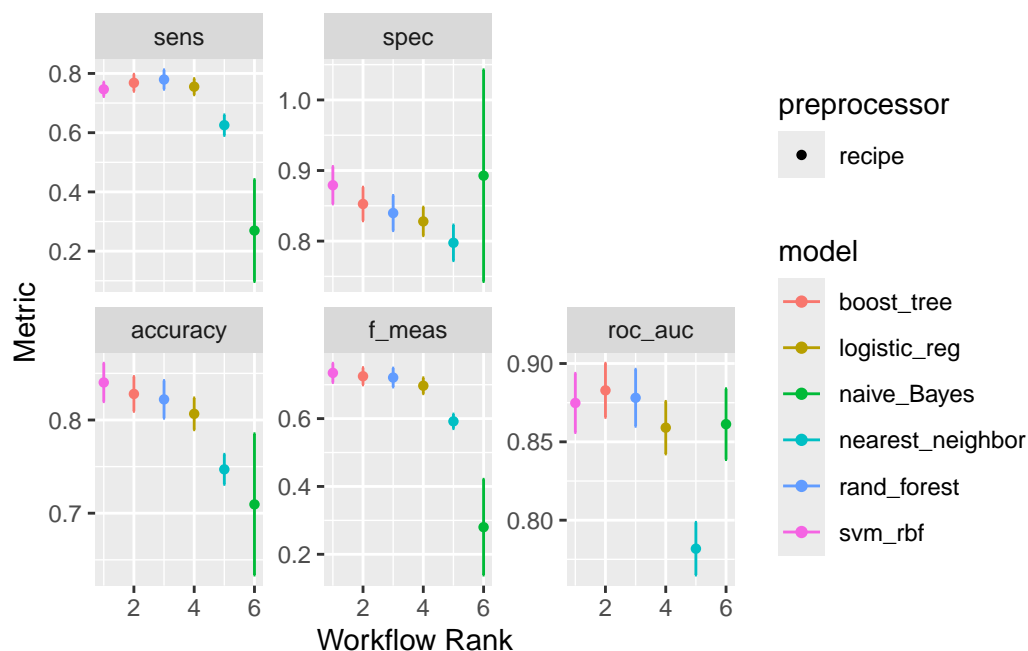
```
plan(sequential)  
  
# Sammenlign resultater  
churn_results |>  
  rank_results(select_best = TRUE) |>  
  select(wflow_id, .metric, mean) |>
```

```
pivot_wider(names_from = .metric, values_from = mean) |>
  arrange(-f_meas)
```

A tibble: 6 x 6

	wflow_id	accuracy	f_meas	roc_auc	sens	spec
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	churn_recipe_svm_rbf	0.840	0.735	0.875	0.746	0.879
2	churn_recipe_xgboost	0.828	0.725	0.883	0.768	0.853
3	churn_recipe_rf	0.822	0.721	0.878	0.779	0.840
4	churn_recipe_logistic	0.807	0.697	0.859	0.755	0.828
5	churn_recipe_knn	0.747	0.592	0.782	0.625	0.798
6	churn_recipe_naive_bayes	0.709	0.280	0.861	0.270	0.893

```
autoplot(churn_results, select_best = TRUE)
```



```

# -----
# 9. Visualiseringer
# -----

# Plot modeller efter deres performance
# -----

# Din tibble, hvis ikke du allerede har den i en variabel:
metrics_df <- tibble::tibble(
  wflow_id = c("churn_recipe_rf", "churn_recipe_xgboost", "churn_recipe_svm_rbf",
               "churn_recipe_logistic", "churn_recipe_knn", "churn_recipe_naive_bayes"),
  accuracy = c(0.825, 0.827, 0.845, 0.807, 0.743, 0.710),
  f_meas   = c(0.724, 0.723, 0.708, 0.697, 0.592, 0.287),
  roc_auc  = c(0.877, 0.881, 0.439, 0.858, 0.778, 0.855),
  sens     = c(0.777, 0.766, 0.643, 0.755, 0.634, 0.272),
  spec     = c(0.845, 0.852, 0.929, 0.828, 0.788, 0.893)
)

# Pivot til langt format
metrics_long <- metrics_df %>%
  pivot_longer(cols = -wflow_id, names_to = "metric", values_to = "score")

# Gør labels lidt pænere
metrics_focus <- metrics_long %>%
  filter(metric %in% c("accuracy", "f_meas", "roc_auc")) %>%
  mutate(metric = case_when(
    metric == "accuracy" ~ "Accuracy",
    metric == "f_meas"  ~ "F1-score",
  ))

```



```

    metric == "roc_auc" ~ "ROC AUC",
    TRUE ~ metric
  ))

# Nr. 1: BarPlot med værdier for denne 3 metrikker

ggplot(metrics_focus, aes(x = metric, y = score, fill = metric)) +
  geom_col(show.legend = FALSE) +
  geom_text(aes(label = round(score, 2)), vjust = -0.3, size = 3.5) +
  facet_wrap(~ wflow_id) +
  ylim(0, 1.05) +
  labs(
    title = "Model performance (Accuracy, F1 og ROC AUC)",
    x = NULL,
    y = "Score"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0),
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  ) +
  scale_fill_brewer(palette = "Set3")

```

Model performance (Accuracy, F1 og ROC AUC)



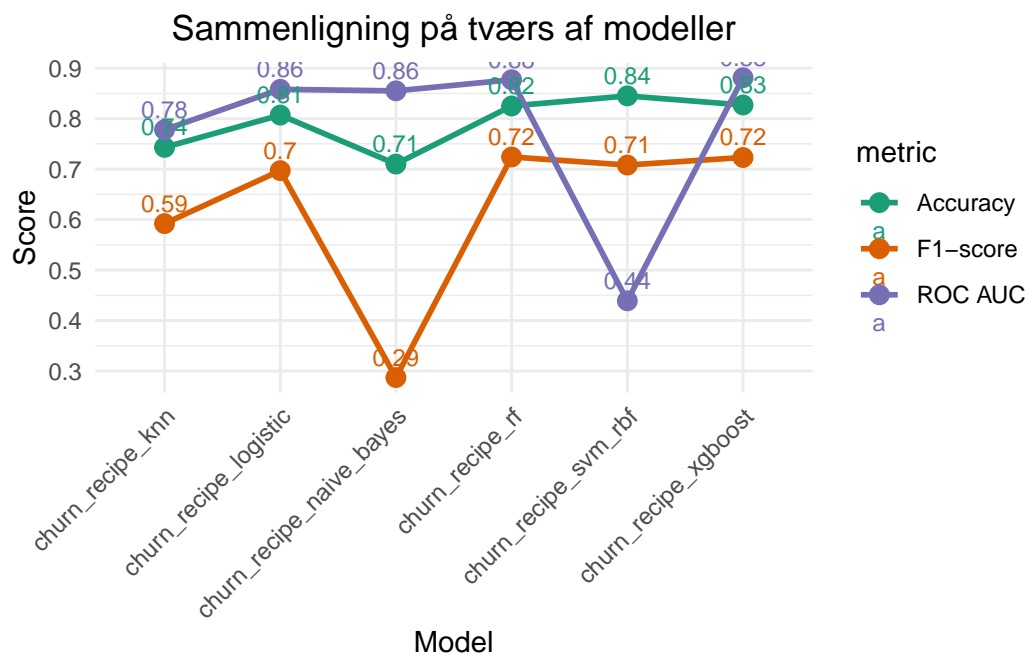
```
ggsave("images/1_model_performance.png", width = 7, height = 4, dpi = 300)
```

```
# Nr. 2: Linje plot
```

```
ggplot(metrics_focus, aes(x = wflow_id, y = score, color = metric, group = metric)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_text(aes(label = round(score, 2)), vjust = -0.7, size = 3) +
  scale_color_brewer(palette = "Dark2") +
  labs(
    title = "Sammenligning på tværs af modeller",
    x = "Model",
    y = "Score"
  ) +
  theme_minimal() +
  theme(
```

```
axis.text.x = element_text(angle = 45, hjust = 1),
plot.title = element_text(hjust = 0.5)
)
```

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



```
ggsave("images/2_linje_plot.png", width = 7, height = 4, dpi = 300)

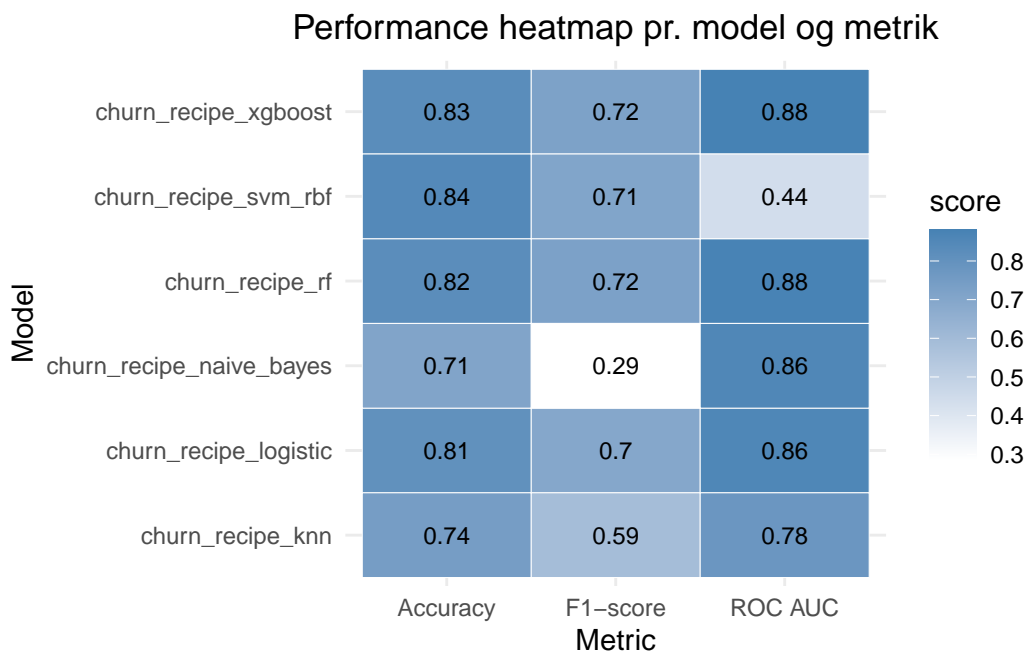
# Nr. 3: Heatmap pr. model og metrik

ggplot(metrics_focus, aes(x = metric, y = wflow_id, fill = score)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(score, 2)), size = 3) +
  scale_fill_gradient(low = "white", high = "steelblue") +
  labs(
```

```

title = "Performance heatmap pr. model og metrik",
x = "Metric",
y = "Model"
) +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5))

```



```

ggsave("images/3_heatmap_pr_model.png", width = 7, height = 4, dpi = 300)

# -----
# Plot for xgboost og Random forest med de vigtigste variabler
# -----

# Hent tuning-resultater for rf og xgboost
rf_result <- churn_results %>% extract_workflow_set_result("churn_recipe_rf")
xgb_result <- churn_results %>% extract_workflow_set_result("churn_recipe_xgboost")

```

```

# Hent workflow (før det er fit)
rf_workflow <- churn_results %>% extract_workflow("churn_recipe_rf")
xgb_workflow <- churn_results %>% extract_workflow("churn_recipe_xgboost")

# Vælg bedste parametre og fit modellen
best_rf <- rf_workflow %>%
  finalize_workflow(select_best(rf_result, metric = "f_meas")) %>%
  fit(data = churn_train)

best_xgb <- xgb_workflow %>%
  finalize_workflow(select_best(xgb_result, metric = "f_meas")) %>%
  fit(data = churn_train)

# Feature importance
vip_rf <- vi(extract_fit_parsnip(best_rf)) %>% mutate(model = "Random Forest")
vip_xgb <- vi(extract_fit_parsnip(best_xgb)) %>% mutate(model = "XGBoost")

# Kombinér og vis kun top 10 vigtigste variabler pr. model
vip_combined <- bind_rows(vip_rf, vip_xgb) %>%
  group_by(model) %>%
  slice_max(order_by = Importance, n = 10) %>%
  ungroup() %>%
  mutate(Variable = str_wrap(Variable, width = 25))

# Plot med labels og tekstrotation optimeret
ggplot(vip_combined, aes(x = reorder(Variable, Importance), y = Importance, fill = model))
  geom_col(show.legend = FALSE) +
  geom_text(aes(label = round(Importance, 2)), hjust = -0.1, size = 3) +

```

```

facet_wrap(~ model, scales = "free") +
coord_flip() +
labs(
  title = "Top 10 vigtigste variabler pr. model",
  x = "Variabel",
  y = "Vigtighed"
) +
theme_minimal() +
theme(
  plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
  strip.text = element_text(size = 12, face = "bold"),
  axis.text.y = element_text(size = 9)
) +
scale_y_continuous(expand = expansion(mult = c(0, 0.10))) #ekstra space til labels

```

Top 10 vigtigste variabler pr. mo



```

ggsave("images/4_top_10_variabler_pr_model.png", width = 7, height = 4, dpi = 300)
# -----

# -----
# 10. Endelig model - Finetuning af Random Forest
# -----

# undersøg kun på hele datasæt

# 1. Lav et workflow set med kun én model: Random Forest
churn_workflow_set_rf <- workflow_set(
  preproc = list(churn_recipe = churn_recipe),
  models  = list(rf = rf_spec) # kun én model
)

# 2. Tænd for parallelisering
plan(multisession)

# 3. Start tidstagning
strt.time <- Sys.time()

# 4. Tuning af kun Random Forest med 25 kombinationer
churn_results_rf <- churn_workflow_set_rf |> # <-- her var der fejl i dit input
  workflow_map(
    resamples = churn_folds,
    grid = 25,
    metrics = churn_metrics,
    control = grid_ctrl,
    seed = 2025
  )

```

```
)
```

```
# 5. Tid brukt
```

```
Sys.time() - strt.time
```

Time difference of 53.96061 secs

```
# 6. Sluk for parallelisering
```

```
plan(sequential)
```

```
# 7. Vis bedste resultater pr. metrik
```

```
churn_results_rf |>
```

```
  rank_results(select_best = TRUE) |>
```

```
  select(wflow_id, .metric, mean) |>
```

```
  pivot_wider(names_from = .metric, values_from = mean) |>
```

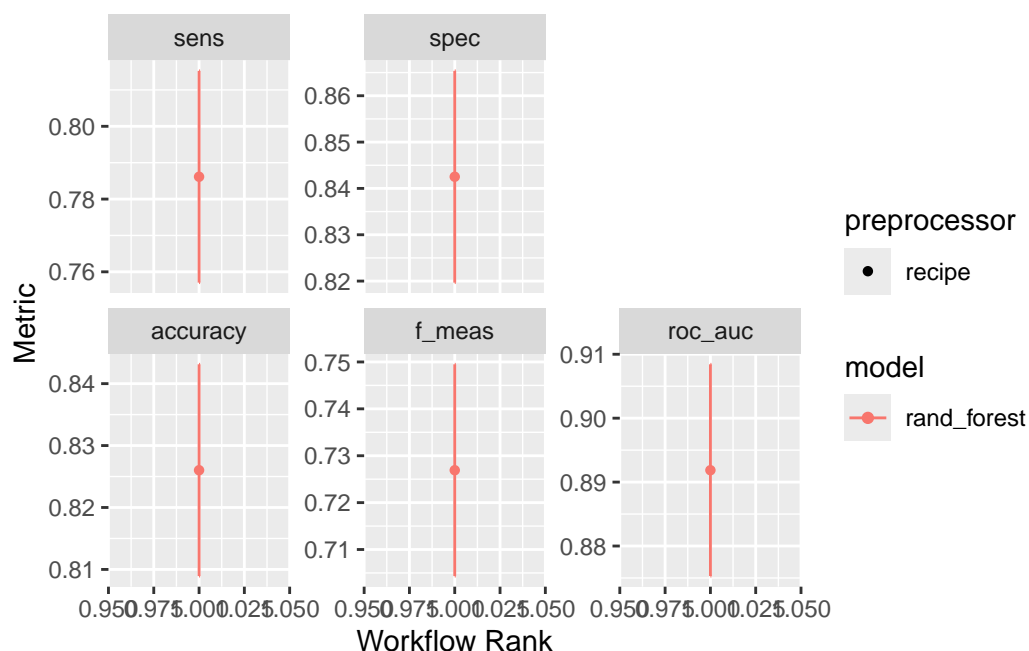
```
  arrange(-f_meas)
```

```
# A tibble: 1 x 6
```

	wflow_id	accuracy	f_meas	roc_auc	sens	spec
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	churn_recipe_rf	0.826	0.727	0.892	0.786	0.843

```
# 8. Visualisér den bedste model
```

```
autoplot(churn_results_rf, select_best = TRUE)
```

```
# -----
# 11. Evaluering af bedste model på testdatasættet (Random Forest)
# -----

# Bemærk: Modellen i dette afsnit er baseret på finetuning med 25 kombinationer

# 1. Find bedste parametre for den bedste model
best_results <- churn_results_rf |>
  extract_workflow_set_result("churn_recipe_rf") |>
  select_best(metric = "f_meas")

# 2. Finaliser workflow med de fundne parametre
final_wf <- churn_results_rf |>
  extract_workflow("churn_recipe_rf") |>
  finalize_workflow(best_results)
```

```
# 3. Træn modellen på træningsdata og evaluer på testdata
```

```
churn_last_fit <- final_wf |>
```

```
  last_fit(split = churn_split, metrics = churn_metrics)
```

```
# 4. Udskriv evalueringsmetrikker
```

```
collect_metrics(churn_last_fit)
```

```
# A tibble: 5 x 4
```

```
  .metric .estimator .estimate .config
```

```
  <chr>    <chr>          <dbl> <chr>
```

```
1 accuracy binary          0.873 Preprocessor1_Model1
```

```
2 f_meas   binary          0.803 Preprocessor1_Model1
```

```
3 sens     binary          0.877 Preprocessor1_Model1
```

```
4 spec     binary          0.872 Preprocessor1_Model1
```

```
5 roc_auc  binary          0.935 Preprocessor1_Model1
```

```
# 5. Gem confusion matrix som objekt (brugbar til præsentation)
```

```
conf_matrix <- churn_last_fit |>
```

```
  collect_predictions() |>
```

```
  conf_mat(estimate = .pred_class, truth = churn)
```

```
# 6. Gem test-prædiktioner hvis ønsket
```

```
test_preds <- collect_predictions(churn_last_fit)
```

```
# 7. Træn endelig model på hele datasættet
```

```
final_model <- fit(final_wf, data = feature_engineering)
```

```
# 8. Gem modellen
```

```
saveRDS(final_model, "final_churn_model.rds")
```

```

# -----
# 11.1 Eksempel: Forudsig churn for én ny virksomhed
# -----

new_company <- tibble(
  Employees = 15,
  PostalCode = factor("8800"),
  CompanyTypeName = factor("Aktieselskab"),
  har_haft_kontakt = factor("Ja"),
  deltaget_i_event = factor("Nej"),
  hjælp_kategori = factor("Strategi Udvikling"),
  medlem_antal_år = 2,
  Branche_navn = factor("Fremstilling af maskiner og udstyr i.a.n."),
  MeetingLength = 180,
  PNumber = 12345678
)

# Forudsiger klassifikation og sandsynlighed
predict(final_model, new_company)           # 0 = bliver, 1 = churn

# A tibble: 1 x 1
#   .pred_class
#   <fct>
1 0

```

```
predict(final_model, new_company, type = "prob")      # churn-sandsynlighed
```

```
# A tibble: 1 x 2
```

```
  .pred_0 .pred_1
```

```
    <dbl>   <dbl>
```

```
1    0.649   0.351
```

```
# -----  
# 11.2 Forudsig churn for ALLE virksomheder og tilføj resultater  
# -----
```

```
# Modellen anvendes nu på hele medlemsdatabasen for at identificere churn-risiko
```

```
# Forudsiger sandsynlighed og klasse
```

```
churn_probs <- predict(final_model, feature_engineering, type = "prob")
```

```
churn_classes <- predict(final_model, feature_engineering)
```

```
# Kombiner og omdøb kolonner
```

```
all_predictions <- bind_cols(churn_probs, churn_classes) |>
```

```
  rename(
```

```
    churn_prob = .pred_1,      # Sandsynlighed for churn
```

```
    churn_class = .pred_class # Klassifikation (0/1)
```

```
)
```

```
# Tilføj til datasættet og konvertér sandsynlighed til procent
```

```
full_results <- feature_engineering |>
```

```
  bind_cols(all_predictions) |>
```

```
  mutate(
```

```
    churn_prob = round(churn_prob * 100, 1)
```

```

)

# Tilføj churn-risikokategorier tidligt (bruges i visualiseringer og rapporter)
full_results <- full_results |>
  mutate(
    churn_risiko = case_when(
      churn_prob >= 80 ~ "Høj risiko",
      churn_prob >= 60 ~ "Moderat risiko",
      churn_prob >= 40 ~ "Lav risiko",
      TRUE           ~ "Minimal risiko"
    )
  )

# -----
# 11.3 Churn-risiko: Filtrér medlemmer (churn == 0) med høj risiko (churn_class == 1)
# -----

top_risiko_medlemmer <- full_results |>
  filter(churn == 0, churn_class == 1) |>
  arrange(desc(churn_prob)) |>
  slice_head(n = 20) # Call to action: top 20

# -----
# 11.4 Visualiseringer: Brancher og postnumre med høj churn
# -----

# Brancher med højest gennemsnitlig churn

```

```

full_results |>
  group_by(Branche_navn) |>
  summarise(gennemsnitlig_churn = mean(churn_prob), n = n()) |>
  arrange(desc(gennemsnitlig_churn)) |>
  slice_head(n = 5) |>
  ggplot(aes(x = reorder(Branche_navn, gennemsnitlig_churn), y = gennemsnitlig_churn)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 5 churn-risiko", x = "Branche", y = "Gns. churn sandsynlighed (%)") +
  theme_minimal()

```



```

ggsave("images/5_brancher_højeste_churn.png", width = 7, height = 4, dpi = 300)

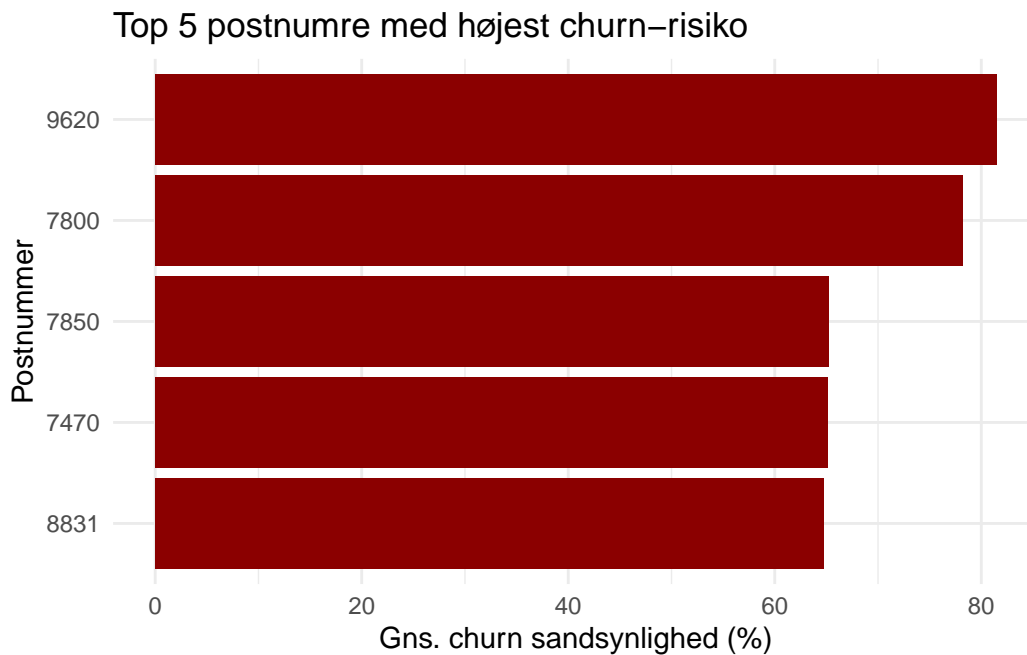
# Postnumre med højest gennemsnitlig churn
full_results |>
  group_by(PostalCode) |>

```

```

summarise(gennemsnitlig_churn = mean(churn_prob), n = n()) |>
arrange(desc(gennemsnitlig_churn)) |>
slice_head(n = 5) |>
ggplot(aes(x = reorder(as.character(PostalCode), gennemsnitlig_churn), y = gennemsnitlig_churn)) +
  geom_col(fill = "darkred") +
  coord_flip() +
  labs(title = "Top 5 postnumre med højest churn-risiko", x = "Postnummer", y = "Gns. churn") +
  theme_minimal()

```



```

ggsave("images/6_postnummer_højeste_churn.png", width = 7, height = 4, dpi = 300)

# -----
# 11.5 Hvad kendetegner virksomheder der IKKE cherner?
# -----

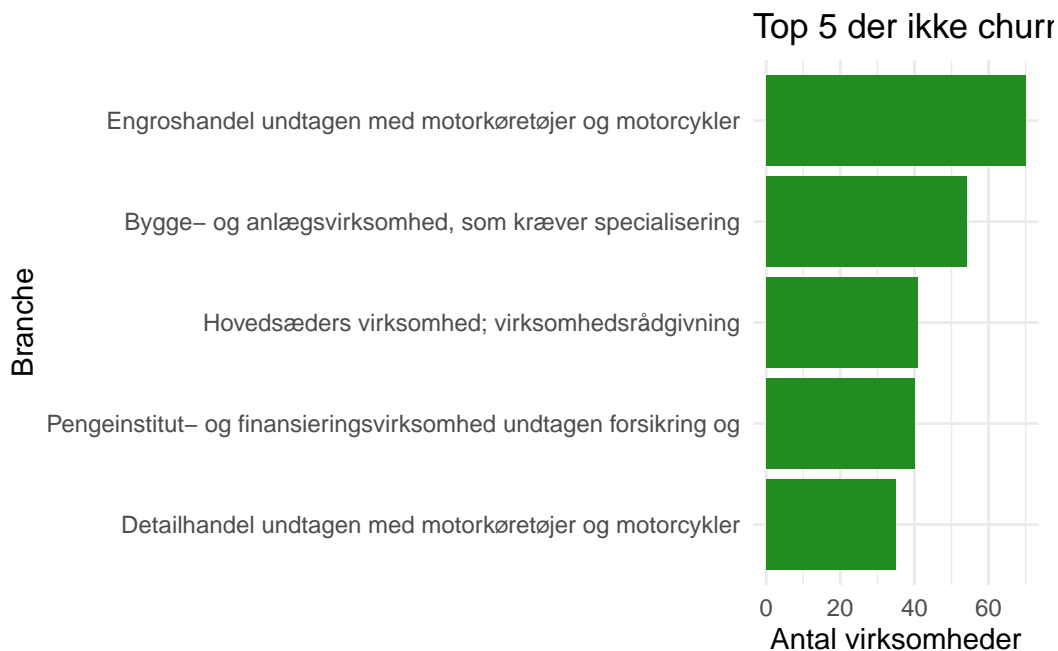
full_results |>

```

```

filter(churn_class == 0) |> # Virksomheder som modellen forudser bliver
count(Branche_navn, sort = TRUE) |>
slice_head(n = 5) |>
ggplot(aes(x = reorder(Branche_navn, n), y = n)) +
geom_col(fill = "forestgreen") +
coord_flip() +
labs(
  title = "Top 5 der ikke churner",
  x = "Branche",
  y = "Antal virksomheder"
) +
theme_minimal()

```



```

ggsave("images/7_top5_der_ikke_churner.png", width = 7, height = 4, dpi = 300)

# Sammenlignende statistik på udvalgte variabler

```



```
# churn_class:
# 0 = modellen tror de bliver
# 1 = modellen tror de cherner
full_results |>
  group_by(churn_class) |>
  summarise(
    mødelængde = mean(MeetingLength),
    medlem_år = mean(medlem_antal_år),
    kontakt_rate = mean(har_haft_kontakt == "Ja"),
    event_rate = mean(deltaget_i_event == "Ja")
  )
```

```
# A tibble: 2 x 5
  churn_class mødelængde medlem_år kontakt_rate event_rate
  <fct>         <dbl>      <dbl>         <dbl>      <dbl>
1 0             31.0        8.21          0.577      0.856
2 1              3.77        7.65          0.233      0.0222
```

```
# -----
# 11.6 Hvad er de vigtigste parametre
# -----

# 1. Udtræk tuning-resultater og workflow
rf_result <- churn_results_rf |> extract_workflow_set_result("churn_recipe_rf")
rf_workflow <- churn_results_rf |> extract_workflow("churn_recipe_rf")

# 2. Find bedste parametre og træ modelen på træningsdata
best_rf <- rf_workflow |>
  finalize_workflow(select_best(rf_result, metric = "f_meas")) |>
```

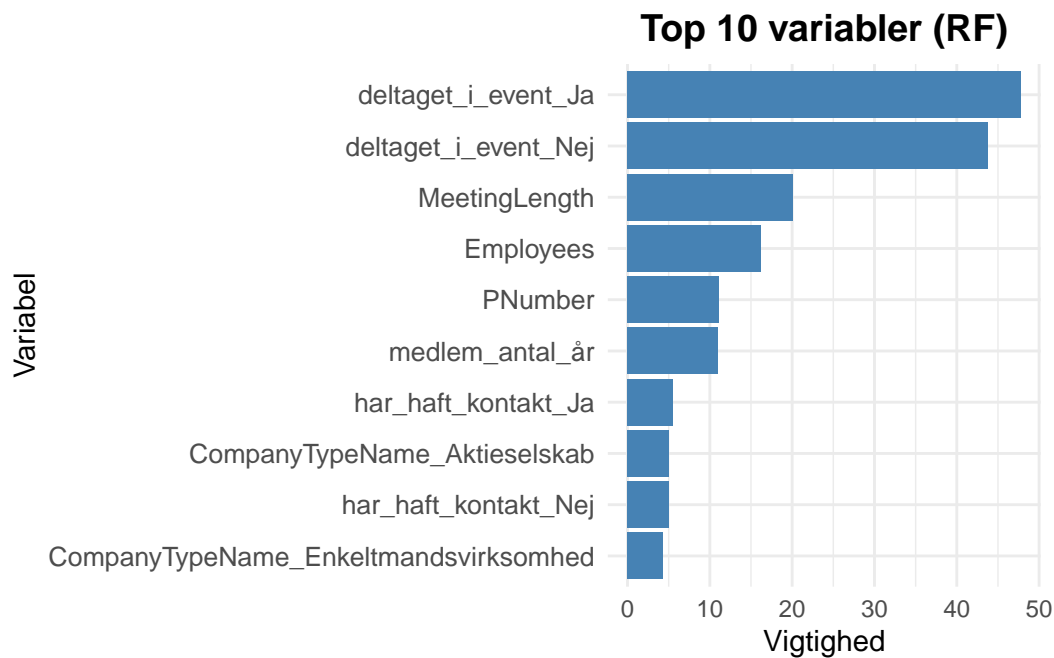
```

fit(data = churn_train)

# 3. Brug vip til at finde top 10 vigtigste variabler
vip_rf <- vi(extract_fit_parsnip(best_rf)) |>
  slice_max(order_by = Importance, n = 10) |>
  mutate(Variable = str_wrap(Variable, width = 30))

# 4. Plot
ggplot(vip_rf, aes(x = reorder(Variable, Importance), y = Importance)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Top 10 variabler (RF)",
    x = "Variabel",
    y = "Vigtighed"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.text.y = element_text(size = 10)
  )

```



```
# 5. Gem billedet
```

```
ggsave("images/8_top_10_variabler_rf.png", width = 7, height = 4, dpi = 300)
```

```
# -----
```

```
# Forklaring af churn-relaterede variabler
```

```
# -----
```

```
# 1. churn:
```

```
# Den faktiske status for virksomheden ifølge databasen.
```

```
# 0 = Virksomheden er stadig medlem.
```

```
# 1 = Virksomheden har meldt sig ud (churnet).
```

```
# Dette er det "rigtige facit", vi forsøger at forudsige.
```

```
# 2. .pred_0:
```

```
# Modellens vurdering af sandsynligheden for, at virksomheden IKKE cherner.
```

```
# Fx 0.93 betyder: modellen mener der er 93 % chance for, at virksomheden bliver medlem
```

```

# OBS: Denne bruges mest til teknisk forståelse - i praksis bruger vi oftest churn_prob

# 3. churn_prob:
# Modellens vurdering af sandsynligheden for churn - konverteret til procent.
# Fx 6.1 betyder: modellen vurderer, at der er 6,1 % risiko for, at virksomheden cherner.
# Denne kolonne er lettest at forstå og bruge i praksis.

# 4. churn_class:
# Modellens endelige beslutning: churn eller ej?
# 1 = modellen tror virksomheden cherner
# 0 = modellen tror virksomheden bliver
# Beslutningen bygger på en tærskel, typisk 50 %

# 5. churn_risiko:
# Kategori baseret på churn_prob - lavet for at gøre det endnu mere overskueligt.
# Fx:
# • "Minimal risiko" → under 40 %
# • "Lav risiko" → 40-59 %
# • "Moderat risiko" → 60-79 %
# • "Høj risiko" → 80 % eller højere

```