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

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 Support Vector Machine opnåede den bedste performance (AUC = 0,88, F1 = 0,74). Alligevel blev Random Forest valgt som slutmodel på baggrund af gennemsigtighed 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 interessevaretager. 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.

Med dette projekt søges udviklet 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 churner. 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.

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.

Pseudonymisering: En teknik hvor direkte identifikatorer fjernes eller maskeres, så data ikke uden videre kan knyttes til en bestemt virksomhed eller person.

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. Af disse har cirka 30 % valgt at opsige deres medlemskab i den analyserede periode. 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 relationel kontakt og engagement kan være centrale indikatorer for medlemsophør.

7.2 Egenskaber ved virksomheder med høj churn

Virksomheder med lav kontaktfrekvens, manglende eventdeltagelse og ingen dokumenteret konsulentinteraktion udviser signifikant højere sandsynlighed for churn. Eksempelvis ses en gennemsnitlig churn-rate på over 40 % blandt virksomheder, der ikke har modtaget nogen form for erhvervsfaglig støtte. I kontrast falder churn-raten til under 20 % blandt virksomheder, der har modtaget strategisk eller organisatorisk rådgivning.

Der identificeres desuden geografiske og branchemæssige mønstre. Brancher med lav netværksværdi – som detailhandel og enkeltmandsvirksomheder – har generelt højere churn, mens produktions- og vidensbaserede erhverv udviser en mere stabil medlemsbase.

7.3 Feature engineering

På baggrund af ovenstående mønstre blev der konstrueret nye forklarende variable for at styrke modellernes prædiktive kapacitet. 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 • hjælp_kategori: tematisk klassifikation af den modtagne konsulentbistand

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

7.4 Modelperformance

Seks maskinlæringsmodeller blev afprøvet: Support Vector Machine (SVM), XGBoost, Random Forest, logistisk regression, K-nearest Neighbors (KNN) og Naive Bayes. Blandt disse opnåede SVM den højeste F1-score (0,74) og AUC (0,88), hvilket indikerer en stærk balance mellem præcision og recall.

På trods af SVM's 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.

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

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 Etiske 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.

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 konsulenternes arbejdsrutiner og prioritering af medlemmer.

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

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

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.

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. Data load

```
# ------------------------------------------------------------------------
# 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) |>
```

```r
  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
```

```r
# 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(  # Erstatter NA i event-kolonner med "Ingen event"
    c(MeetingLength, EventExternalId, EventPublicId, Description,
      LocationId, MaxParticipants, EventLength, EventId),
    ~ if_else(is.na(.), "Ingen event", as.character(.))
  ))
```

```r
# 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
```

```r
  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. Clean data

```
# ------------------------------------------------------------------------------
# 3.1: Første kig på datastrukturen
# Giver et hurtigt overblik over variabelnavne, typer og eksempelværdier
# ------------------------------------------------------------------------------
# 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")
```

```r
# ------------------------------------------------------------------------
# 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("[ /\\-]+", "_") |> # Erstatter 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
```

```r
# ------------------------------------------------------------------------
# 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: Konverter udvalgte kolonner til faktorer,
```

```r
# 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")
```

5. EDA

```r
# EDA
```

6. Preprocessing

```r
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

```r
# 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

```r
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()
```

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

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

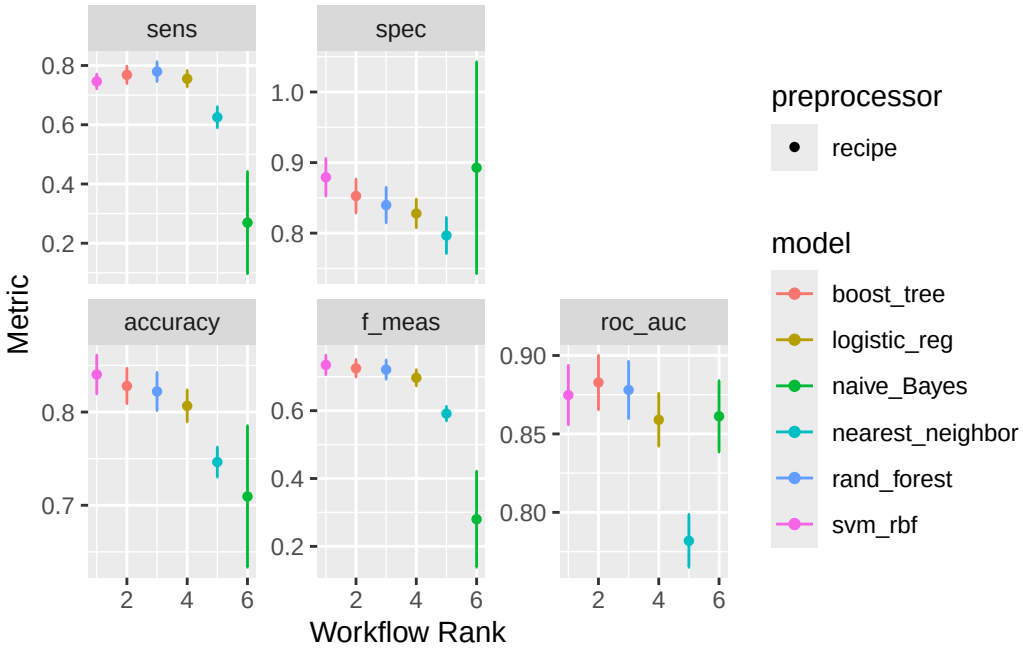
```
Time difference of 3.10225 mins
```

```r
plan(sequential)

# Sammenlign resultater
```

```r
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           0.746  0.591   0.782 0.625 0.797
6 churn_recipe_naive_bayes   0.709  0.280   0.861 0.270 0.893
```

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

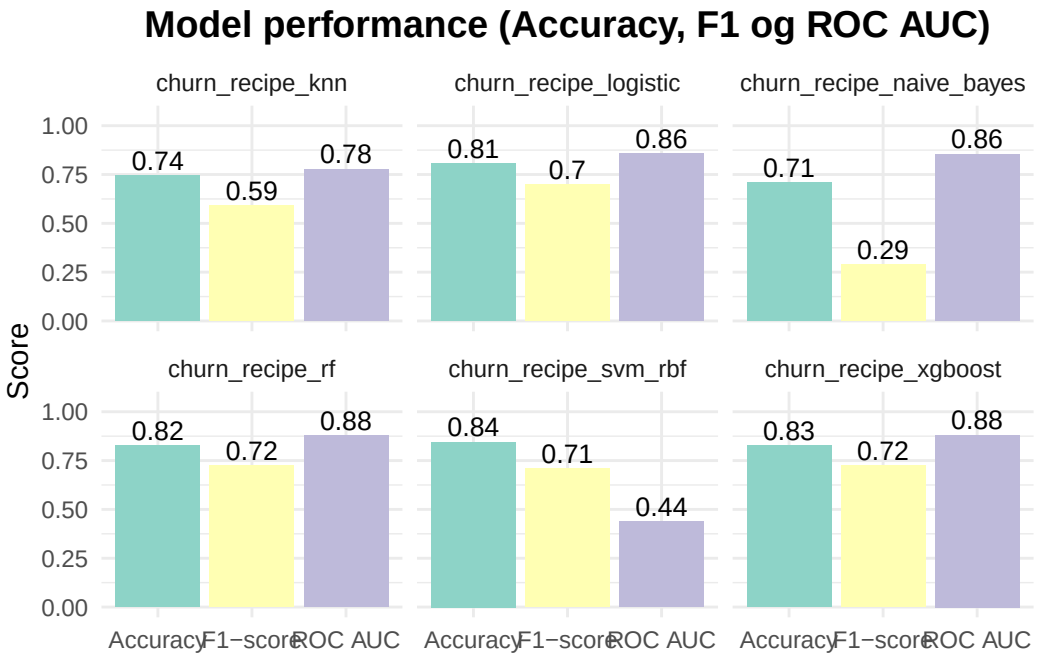


9. Visualiseringer

```
# ------------------------------------------------------------------------
# 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,
```

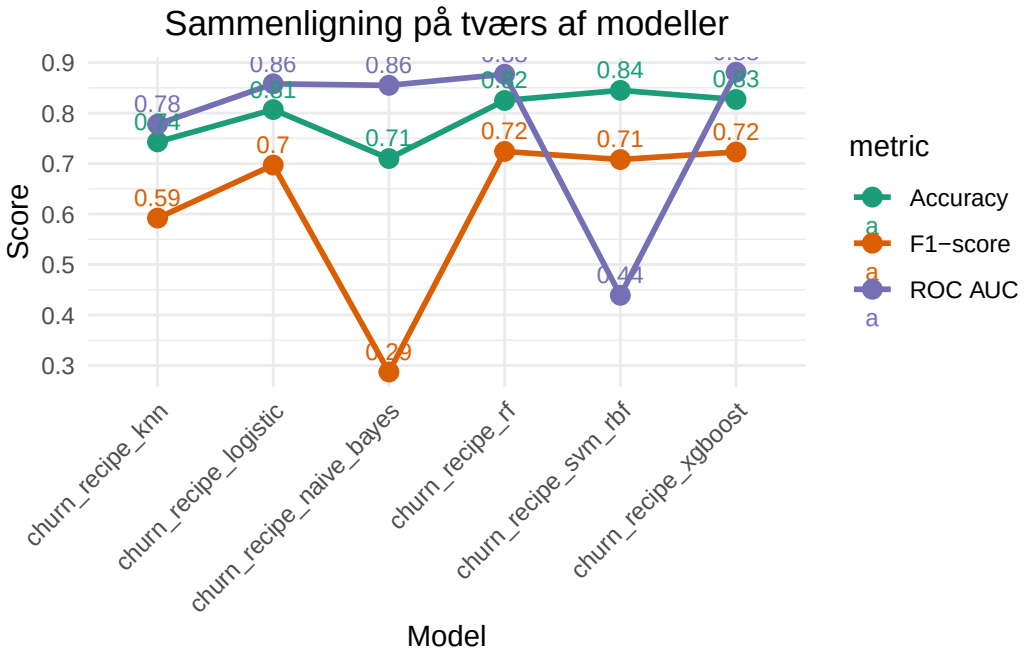
```
    y = "Score"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0),
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  ) +
  scale_fill_brewer(palette = "Set3")
```



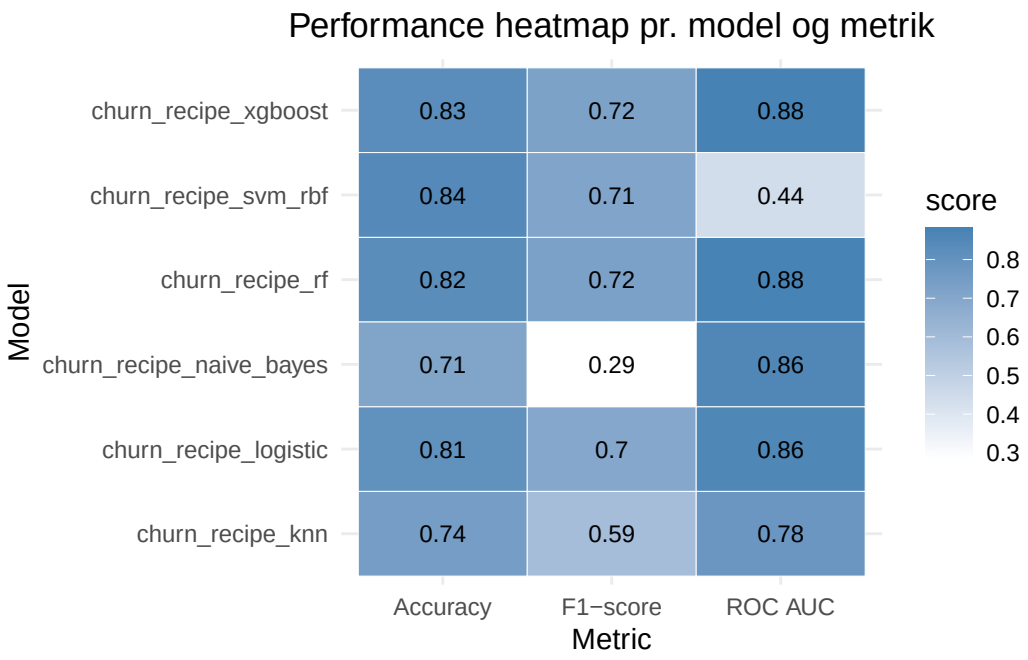

```
# Nr. 2: Linje plot


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.
```

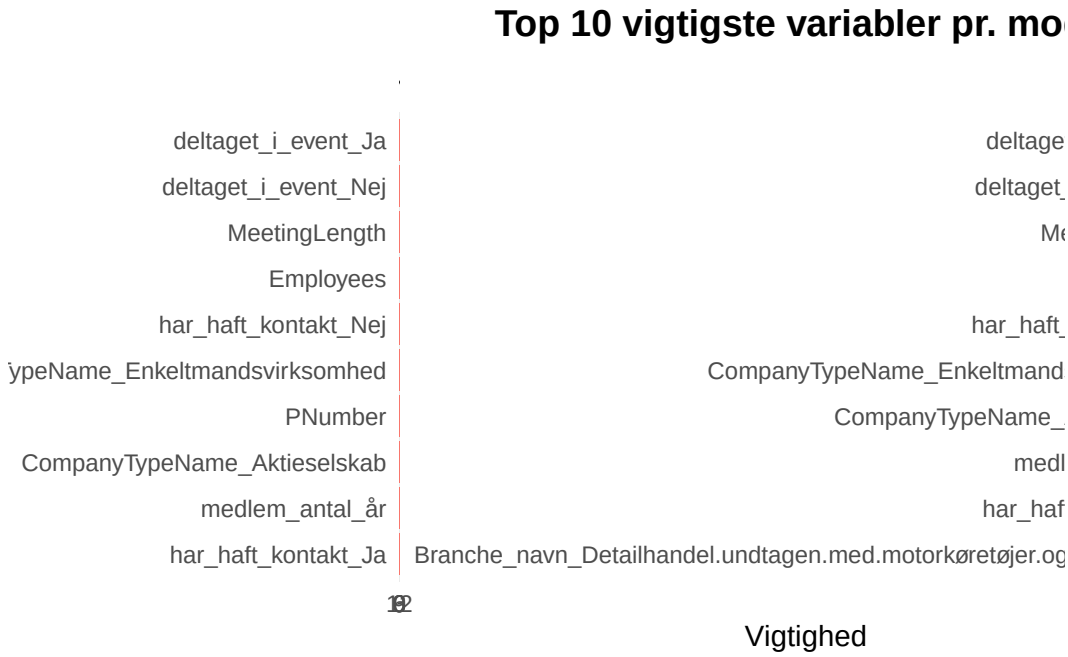


```r
# Nr. 3: Heatmap pr. model og metrik


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))
```



```r
# ------------------------------------------------------------------------
# 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) %>%
```

```r
  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))) #ektra space til labels
```




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

10. Endelig model (sidste tuning - skal vi gøre det?)

```
# # RANDOM FOREST: Finetuning og slutmodel
#
# # Tænd for parallelisering for hurtigere tuning
# plan(multisession)
#
# # Definér Random Forest med tunbare parametre
# rf_spec <- rand_forest(
#   mtry  = tune(),
#   min_n = tune()
# ) |>
#   set_engine("ranger", importance = "impurity") |>
#   set_mode("classification")
#
```

```
# # Afgræns tuningparametre baseret på træningsdata
# rf_params <- extract_parameter_set_dials(rf_spec) |>
#   finalize(churn_train)
#
# # Opret regulært grid (5 x 5 = 25 kombinationer)
# rf_grid <- grid_regular(rf_params, levels = 5)
#
# # Opsæt workflow med model og recipe
# rf_workflow <- workflow() |>
#   add_model(rf_spec) |>
#   add_recipe(churn_recipe)
#
# # Start tidstagning og kør tuning med 10-fold cross-validation
# start_time <- Sys.time()
#
# rf_results <- tune_grid(
#   rf_workflow,
#   resamples = churn_folds,
#   grid = rf_grid,
#   metrics = metric_set(f_meas, roc_auc, accuracy),
#   control = control_grid(save_pred = TRUE, verbose = TRUE)
# )
#
# Sys.time() - start_time  # Vis samlet træningstid
#
# # Sluk for parallelisering
# plan(sequential)
#
```

```
# # Find bedste parametre baseret på F1-score
# best_rf <- tune::select_best(rf_results, metric = "f_meas")
#
# # Finaliser workflow med optimal parameterkombination
# final_rf_workflow <- finalize_workflow(rf_workflow, best_rf)
#
# # Fit endelig model på hele træningsdatasættet
# final_rf_model <- fit(final_rf_workflow, data = churn_train)
#
# # Gem slutmodel til senere brug
# saveRDS(final_rf_model, "data/final_rf_model.rds")
#
# # Visualiser F1-score pr. parameterkombination
# rf_results |>
#   collect_metrics() |>
#   filter(.metric == "f_meas") |>
#   ggplot(aes(x = mtry, y = mean, color = factor(min_n))) +
#   geom_line(linewidth = 0.8) +
#   geom_point(size = 2.5) +
#   labs(
#     title = "F1-score pr. kombination (Random Forest)",
#     subtitle = "Grid search med 25 kombinationer (5 x 5)",
#     x = "mtry",
#     y = "F1-score",
#     color = "min_n"
#   ) +
#   theme_minimal()
#
```

```
# # Udskriv den bedste parameterkombination
# best_rf
#
# # Vis samlet metrikoversigt
# rf_results |>
#   collect_metrics()
#
# # Vis bedste kombination pr. metrik
# rf_results |>
#   collect_metrics() |>
#   group_by(.metric) |>
#   filter(mean == max(mean)) |>
#   arrange(.metric)
```

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

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

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

```r
# 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         0.863 Preprocessor1_Model1
2 f_meas   binary         0.791 Preprocessor1_Model1
3 sens     binary         0.877 Preprocessor1_Model1
4 spec     binary         0.857 Preprocessor1_Model1
5 roc_auc  binary         0.916 Preprocessor1_Model1
```

```r
# 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")
```

```r
# ------------------------------------------------------------------------------
# 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
```

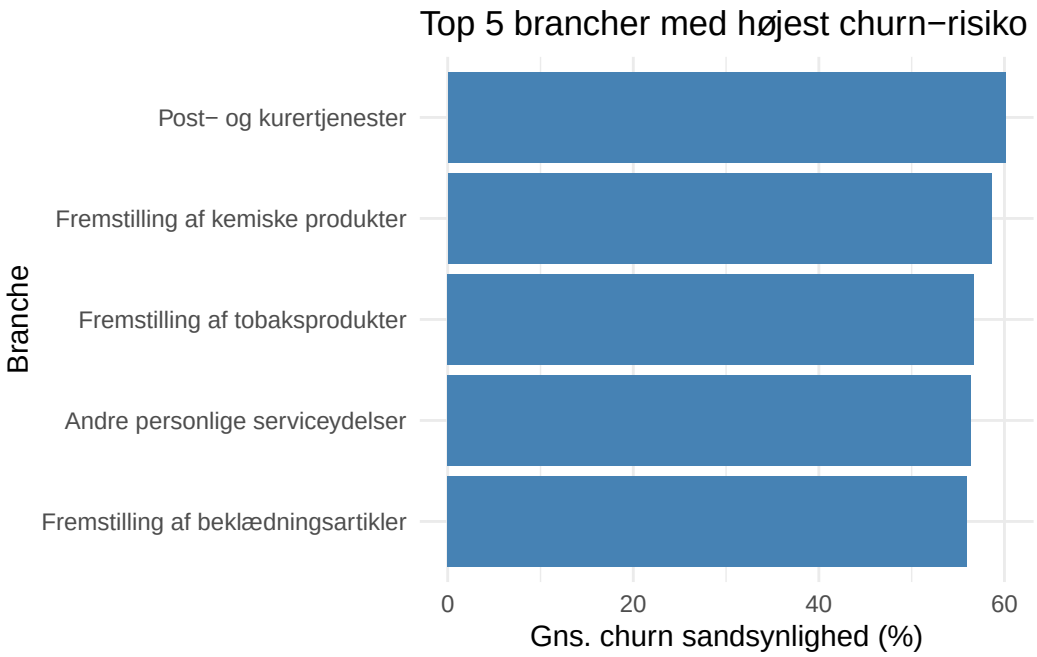
```r
predict(final_model, new_company, type = "prob")     # churn-sandsynlighed
```

```
# A tibble: 1 x 2
  .pred_0 .pred_1
    <dbl>   <dbl>
1   0.547   0.453
```

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

# 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)
  )
```

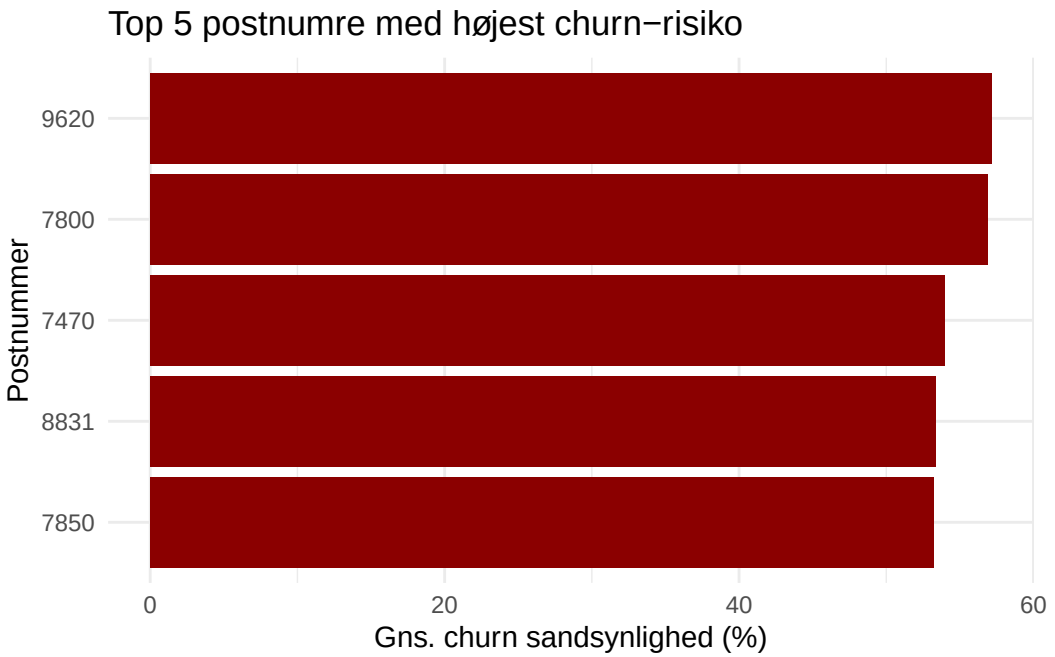
```r
# Tilføj churn-risikokategorier tidligt (bruges i visualiseringer og rapporter)
full_results <- full_results |>
  mutate(
    churn_risiko = case_when(
      churn_prob >= 80 ~ "Høj risiko",
      churn_prob >= 60 ~ "Moderat risiko",
      churn_prob >= 40 ~ "Lav risiko",
      TRUE            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

```r
  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 brancher med højest churn-risiko", x = "Branche", y = "Gns. churn sa
  theme_minimal()
```



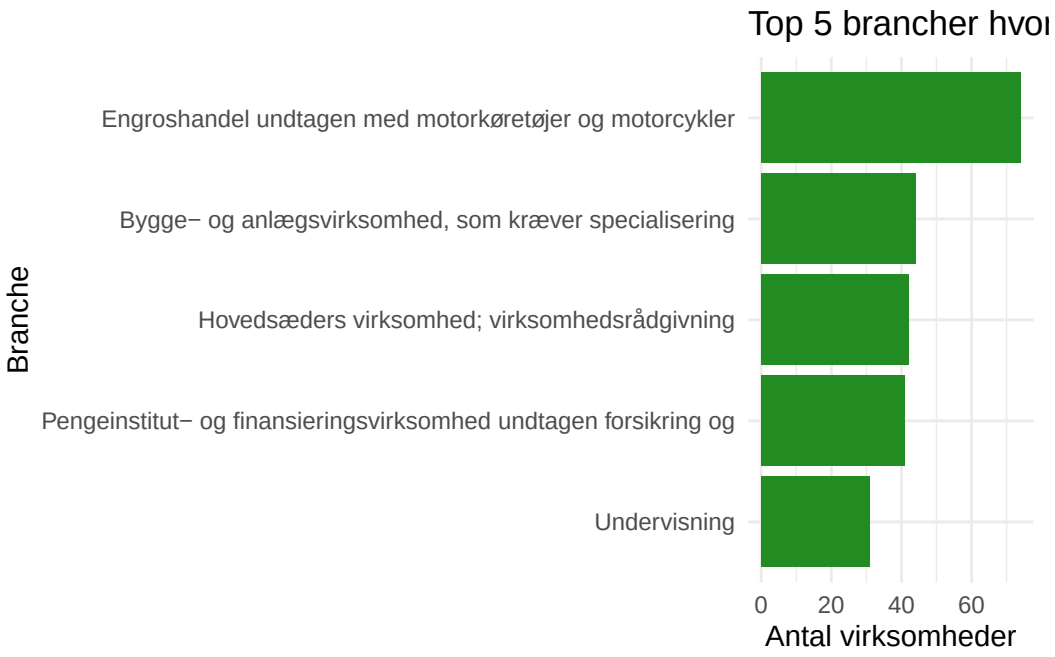

```r
# Postnumre med højest gennemsnitlig churn
full_results |>
  group_by(PostalCode) |>
  summarise(gennemsnitlig_churn = mean(churn_prob), n = n()) |>
  arrange(desc(gennemsnitlig_churn)) |>
  slice_head(n = 5) |>
  ggplot(aes(x = reorder(as.character(PostalCode), gennemsnitlig_churn), y = gennemsnitlig
```

```
  geom_col(fill = "darkred") +
  coord_flip() +
  labs(title = "Top 5 postnumre med højest churn-risiko", x = "Postnummer", y = "Gns. churn
  theme_minimal()
```




```
# ------------------------------------------------------------------------------
# 11.5 Hvad kendetegner virksomheder der IKKE churner?
# ------------------------------------------------------------------------------

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 brancher hvor virksomheder ikke churner",
    x = "Branche",
    y = "Antal virksomheder"
  ) +
  theme_minimal()
```




```
# Sammenlignende statistik på udvalgte variabler
# churn_class:
# 0 = modellen tror de bliver
# 1 = modellen tror de churner
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                31.4       8.26        0.595     0.891
2 1                 4.15      7.63        0.232     0.0225
```

```
# ------------------------------------------------------------------------
# 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 churner.
#    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 churne
#    Denne kolonne er lettest at forstå og bruge i praksis.
```

```
# 4. churn_class:
#    Modellens endelige beslutning: churn eller ej?
#    1 = modellen tror virksomheden churner
#    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
```