Norges miljø- og biovitenskapelige universitet

Masteroppgave 2020 30 stp

Handelshøyskolen
Dag Einar Sommervoll

Bruk av maskinlæring for kundebevaring i forsikringsbransjen: en case-studie

Machine learning as a tool for customer retention in the insurance industry: a case study

Fredrik Røed-Johansen og Håkon Strand
Master i Økonomi og Administrasjon

Forord

Innleveringen av denne masteroppgaven markerer avslutningen på et 5 års økonomistudium ved Norges Miljø- og Biovitenskapelige Universitet, NMBU. I oppgaven har vi sett på kunderelaterte utfordringer til forsikringsselskapet Frende. Vi vil takke selskapet for tilgangen til datasettet som gjorde dette mulig. I tillegg ønsker vi å rette en takk til Eivind Reikerås, Mats Bruun og Anders Dræge, våre kontaktpersoner fra Frende, for god hjelp og veiledning.

Videre vil vi rette en stor takk til vår veileder, Dag Einar Sommervoll, for god veiledning og konstruktive tilbakemeldinger gjennom hele prosessen. Takk også til familie og venner for støtte, oppmuntring og korrekturlesning i perioden.

Fredrik Røed-Johansen og Håkon Strand

Ås, 02. juni 2020

Sammendrag

Denne oppgaven er en case-studie i samarbeid med Frende Forsikring, og ser på bruken av maskinlæring som virkemiddel for å forbedre tilnærmingen til kundebevaring. Forskning viser til en betydelig høyere kostnad ved anskaffelse av nye kunder kontra bevaring av gamle, og en reduksjon i avgangsprosenten er derfor ønskelig. Problemet kundeavgang er i oppgaven delt inn i flere deler. Med dette er det skapt tre målvariabler, henholdsvis avgang, årsak til avgang og om kunde kan reddes, referert til som prediksjon 1,2 og 3. Utgangspunktet for oppdelingen er at det i realiteten kun er avgangskunder som avgår til nytt selskap det vil være mulig å redde.

Totalt er syv forskjellige maskinlæringsmodeller blitt testet, hvorav de tre med best resultater ble fokus for videre arbeid. Fra de tre, alle basert på gradient boosting, er *Light Gradient Boosted Machine (LightGBM)* valgt som endelig modell for samtlige prediksjoner, og følgelig modellen analysen er basert på. For prediksjon 1 og 2 ble F1-score det viktigste målet å vurdere modellen etter, mens det for prediksjon 3 hovedsakelig ble vektlagt en kostnadsmatrise for valg av modell.

Tolkning av modellens beslutningsprosess ble gjort ved å benytte SHAP-verdier, et mål som forklarer variablenes bidrag på utfallet, og ga grunnlag for gruppering av utsatte kunder.

Resultatene viser at maskinlæring kan forbedre prosessen rundt kundebevaring. Potensielle avgangskunder kan bli identifisert på et tidlig stadium og arbeid med å redde kunder kan bli gjort med større presisjon. Dette resulterer i høyere verdi per redningsforsøk og dermed økt forretningsverdi.

Abstract

The thesis is a case study in collaboration with the Norwegian insurance company Frende and will be exploring the use of machine learning as a tool to improve the approach regarding customer retention. Research indicates a significantly greater cost of acquiring new customers compared to retaining old ones, and a reduction of the churn rate is therefore desirable. The problem of customer churn is divided into three categories. With this, three new target variables have been created, respectively churn, reason for churning and whether the customer can be saved. For future reference these are referred to as prediction 1, 2 and 3. The reason for this division, is that only customers who are leaving for a new insurance company are the ones it will be possible to save.

There are seven different machine learning models tested in this thesis, of which the three with the most promising results were chosen for additional scrutiny. Of the three, all based on gradient boosting algorithm, *Light Gradient Boosted Machine* (*LightGBM*) was chosen as the final model for all three predictions, and hence also the model further analysis was based upon. For prediction 1 and 2 F1-score became the most important measure for model selection. For prediction 3, a cost matrix became the most important tool.

Interpretation of the model's decision-making process was done using SHAP-values, a measure that explains the variables' contribution to the outcome. This measure provided the basis for grouping exposed customers.

From the results it became clearer how machine learning can improve the process of customer retention. Potentially churned customers can be identified at an early stage and the effort related to saving customers can be done with greater precision. This results in higher value per rescue attempt, and thus increased business value.

Begrepsliste

True Positive (TP):	Prediksjoner klassifisert riktig i klasse 1.
False Positive (FP):	Prediksjoner klassifisert feilaktig i klasse 1.
True Negative (TN):	Prediksjoner klassifisert riktig i klasse 0.
False Negative (FN):	Prediksjoner klassifisert feilaktig i klasse 0.
Overfitting:	En modell plukker opp støy sammen med mønstrene i dataen. Oppstår hvis en modell legger for mye fokus på dataen under treningsperioden. Klarer ikke å generalisere for fremtidige tilfeller.
Underfitting:	En modell klarer ikke å plukke opp mønstrene i dataen. Oppstår hvis en modell legger for lite fokus på dataen under treningsperioden. Modellen generaliserer for mye.
Recall:	Et mål for antall korrekte gjenkjenninger i en bestemt klasse. For formel se vedlegg 12.1.3 (s. 57).
Precision:	Hvor stor andel av prediksjonene for en bestemt klasse som er riktig i forhold til feil. For formel se vedlegg 12.1.3 (s. 57).
F1-score:	Et sammensatt resultat av *Precison* og *Recall.* For formel se vedlegg 12.1.3 (s. 57).
Accuracy:	Et mål på en modells totale treffsikkerhet. For formel se vedlegg 12.1.3 (s. 57).
Cutoff:	I binær klassifisering er cutoff satt for å skille prediksjoner fra klasse 1 og 0. Observasjoner over en gitt sannsynlighet blir plassert i klasse 1. Øvrige observasjoner blir plassert i klasse 0. Cutoff bestemmer grensen for denne sannsynligheten.

Innhold

1 Innledning

1.1 Forskningsspørsmål

Oppgaven tar for seg følgende problemstilling:

- Hvordan kan maskinlæringsmodeller bidra til kundebevaring?

Den aktuelle problemstillingen vil bli sett på i form av en case-studie. Videre forklaring om bakgrunn og tilnærming vil bli spesifisert i påfølgende delkapitler.

1.2 Bakgrunn

Frende Forsikring er et relativt nytt forsikringsselskap med oppstart i 2007. Opprinnelig var det et samarbeid mellom 4 sparebanker, men har vokst raskt og eies i dag av 15 sparebanker. Denne oppgaven er resultat av et samarbeid med nevnte forsikringsselskap og vil se på kunderelaterte utfordringer for deres om lag 150 000 private forsikringstakere.

Gode kunderelasjoner er sentralt i forsikringsbransjen. Med en forretningsmodell som baserer seg på jevnlige innbetalinger er det naturlig at det fokuseres på å beholde kunder så lenge som mulig. Unntak eksisterer selvsagt, grunnet forsikringsbransjens natur, ved at enkeltkunder med mange, eller store, skader kan være utgiftsposter fremfor inntektskilder. På generelt grunnlag er det dog rimelig å videreføre antagelsen om at det å beholde kunder lenge er gunstig for bedriften. I tillegg viser forskning betydelig høyere utgifter, opp til 12 ganger mer, relatert til anskaffelse av nye kunder fremfor bevaring av gamle (Torkzadeh, Chang, & Hansen, 2006). Forsikringsbransjen fikk i 2006 også en ekstra utfordring ved at lovene rundt kundeavgang ble endret (Forsikringsavtaleloven, 2006). Tidligere var kunden nødt til å vente til utløpt kontrakt før vedkommende eventuelt kunne velge å endre forsikringsselskap. Den nye forsikringsavtaleloven, § 12-3 (2. ledd)**,** ga forsikringstaker mulighet til endring når de selv måtte ønske, selv om de i praksis allerede var knyttet til et selskap.
Samlet underbygger dette hvorfor kundebevaring er sentralt i bransjen, noe som vil bli sett videre på i løpet av denne case-studien.

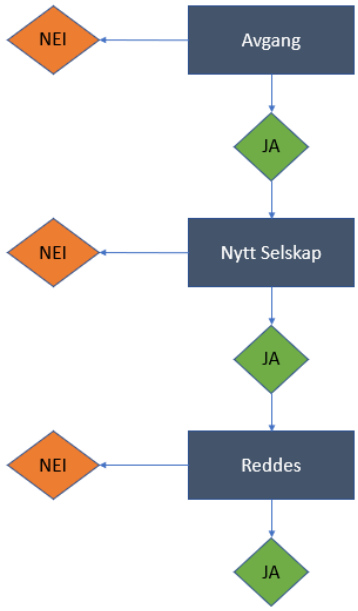
For å kunne kommentere dette nærmere kreves det mye informasjon. Som nevnt har Frende ved utgangen av 2019 om lag 150 000 privatkunder. En kundeportefølje av den størrelsen gir potensielt tilgang til svært mye informasjon og som et resultat av dagens digitale tidsalder blir

mye av informasjonen om disse kundene allerede lagret i strukturerte datasett. Dette danner hva vi kaller *big data*, et begrep som i praksis betyr lagring av store mengder data, og gir mulighet for bruk av maskinlæringsmodeller. Slike modeller er kompliserte algoritmer som ved hjelp av inngangsverdier kan være med å predikere diverse utfall før de inntreffer. Teknologien benyttes allerede i forsikringsbransjen og er i så måte testet (Vafeiadis, Diamantaras, Sarigiannidis, & Chatzisavvas, 2015). Man behøver med andre ord ikke oppfinne hjulet på nytt, men kan benytte tidligere forskning og metoder, og bygge videre på dette.

1.3 Tilnærming til problemstilling

Etter samtaler med samarbeidspartner, Frende, er det kommet frem til et overordnet mål om å predikere kundeavgang for den private delen av kundesegmentet. En kunde blir definert som avgått dersom vedkommende kansellerer alle forsikringer, enten ved endt behov eller bytte av selskap. Maskinlæring er metoden tatt i bruk for å komme med prediksjonene. I oppgaven benyttes et datasett brakt til veie av Frende. Datasettet inneholder diverse variabler relatert til kundenes karakteristikker, men kommer ikke ferdigstilt. Før modeller er programmert og testet er det derfor foretatt en hel del databehandling. I tillegg er problemet kundeavgang delt opp i mindre delproblemer.

Selv om kundeavgang i seg selv er interessant, vil det som regel være en sammensatt prosess som resulterer i at en kunde forlater selskapet, og enkelte ganger vil en avskjed være uunngåelig, som ved tilfeller der forsikring utløper av naturlige årsaker. Slike kunder vil ikke nødvendigvis være av direkte interesse for Frende, men kan like fullt bidra med verdifull informasjon. Med dette som grunnlag er problemet stykket opp i tre målvariabler. Disse er henholdsvis *avgang*, *nytt selskap* og *reddet*. De følger hverandre naturlig, slik illustrert ved figur 1, der avgang måler hvorvidt en kunde har avgått eller ikke, nytt selskap måler hvorvidt grunnen til avgang er at kunden forlater til fordel for et annet selskap og avsluttes med prediksjon rettet mot hvorvidt kunder som endrer selskap kan reddes. En reddet kunde vil være tilfeller der Frende forhindrer



Figur 1: Oversikt over hvordan målvariablene følger hverandre.

avgangsprosessen. Videre i oppgaven refereres disse stegene ofte til som prediksjon 1, 2 og 3. Målvariabelen knyttet til nytt selskap er variabelen Frende anser som mest interessant. Begrunnelsen ligger i at det i hovedsak er disse kundene det konkurreres om og utelukker de uunngåelige avgangene tidligere nevnt.

Oppgaven vil videre dreie seg om å benytte maskinlæring som metode for å skape økonomisk vinning. Modellene blir med andre ord et verktøy som bidrar til å løse problemene rundt kundeavgang. Oppgaven er strukturert slik at diverse teoretiske aspekter blir gjennomgått i kapittel 2, i hovedsak knyttet til kundefrafall og teknologien rundt maskinlæring. I kapittel 3 vil det gis en kort presentasjon av dataen, før metodene benyttet i oppgaven blir forklart i kapittel 4. Detaljert beskrivelse av databehandling er å finne i

12.2 (s. 58-62). Senere, i kapittel 5 og 6, blir henholdsvis maskinlæringsmodellene gjennomgått før resultatene fra endelig modell blir presentert. Avslutningsvis legger kapittel 7 opp til en diskusjon av nevnte resultater og forskningsspørsmål reist innledningsvis, samt kommentarer til videre arbeid, før det i kapittel 8 konkluderes.

1.4 Teknologi

1.4.1 Python

For oppgaven er Python, versjon 3.5, blitt benyttet som programmeringsspråk. Da koden er relativt enkel å lese, og det i tillegg eksisterer et bredt utvalg av relevante pakker og bibliotek, egner språket seg godt til maskinlæring. Det meste datarelaterte arbeidet har foregått i Python. I hovedsak er dette en kombinasjon av visualisering, databehandling og konstruksjon av maskinlæringsmodeller. Til arbeidet benyttes pakker og bibliotek som pandas, numpy, seaborn, matplotlib, sklearn, xgboost, lightgbm og catboost. Videre i oppgaven vil det legges frem de viktigste hyperparameterne til modellene benyttet. Det nevnes nå at de som ikke blir kommentert er satt til standard for sin gjeldende pakke, da det er svært mange å ta stilling til dersom alle skulle blitt kommentert.

1.4.2 Dataiku

Den andre teknologien benyttet for forhåndsbehandling, visualisering og maskinlæring er Dataiku. I Dataiku er mange av funksjonene som kan gjøres ved programmering i forhold til maskinlæring tilgjengelig, men med fokus på brukervennlighet og hurtighet. Funksjonene er

implementert i en «point and click»-form, hvor for eksempel dataforhåndsbehandling og tuning av hyperparametere kan gjøres med enkle klikk. Bruken av programmet har gitt muligheten til å teste flere modeller og hyperparametere mer effektivt, og gitt basis for hva hovedfokuset har vært for videre programmering i Python.

2 Teori

2.1 Maskinlæring

I dette delkapittelet vil det bli presentert en forklaring på hva maskinlæring er, og hva det kan brukes til. Allerede i 1959 ble begrepet tatt i bruk av Arthur Samuel, som definerte det som «datamaskinens evne til å lære uten å bli eksplisitt programmert» (Samuel, 1959). Maskinlæring skiller seg dermed fra å såkalt «hardkode» mulige utfall, det vil si å gi maskinen et utfall Y dersom X skjer, og legger i stedet vekt på å lære fra tidligere erfaringer og eksempler. Selv om definisjonen i seg selv er gammel, fanger den fremdeles opp kjernen i hva maskinlæring baseres på.

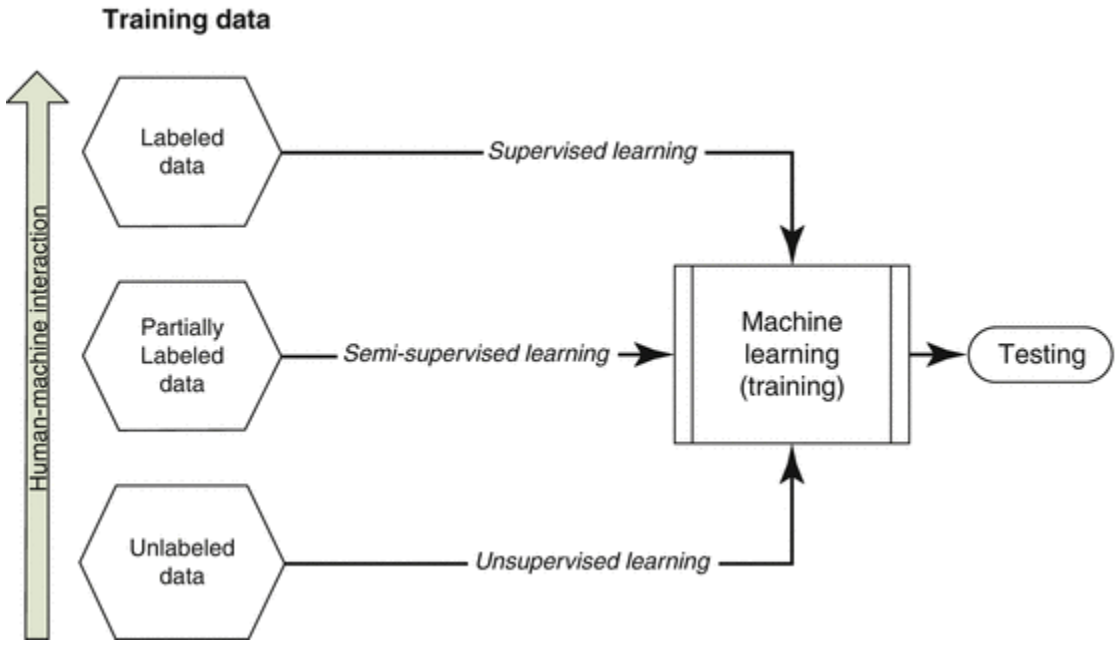
Nytten av å lære fra tidligere eksempler kommer svært tydelig frem ved å se på algoritmer relatert til bildegjenkjenning. Gitt et problem der målet er å gjenkjenne ansikter, er det ikke godt å si hvor man ville begynt dersom det skulle programmeres eksplisitte algoritmer. Noe av problemet ligger i at mennesker selv ikke alltid har evnen til å forklare fremgangsmåter og bakgrunn for egne beslutninger, et fenomen referert til som Polanyis paradoks, oppkalt etter filosof Michael Polanyi (Autor, 2014). Når en selv ikke kan forklare fremgangsmåte, blir det utfordrende å forklare det til en maskin, som i all hovedsak baserer seg på regler den blir fortalt. Ved å ta i bruk maskinlæringsmodeller, og gi tilgang til tidligere observerte eksempler av utfallet, både korrekt og galt, vil modellen selv finne frem til kjennetegn den benytter i beslutningsprosessen.

2.2 Veiledet læring

Det finnes forskjellige typer maskinlæring, nærmere bestemt veiledet læring, ikke-veiledet læring, semi-veiledet læring og forsterket læring. I denne oppgaven vil hovedfokuset ligge på veiledet læring. Navnet stammer fra at man vet utgangsverdiene til de ulike tilfellene, og feil eller riktig predikering blir henholdsvis «straffet» eller «belønnet» av en kunstig lærer som overvåker prosessen.

Under treningsprosessen vil algoritmen forsøke å finne mønstre i inngangsverdiene som fører til korrekt utgangsverdi, eller målvariabel. I korte trekk vil neste steg i prosessen være å simulere et miljø som ligner virkeligheten, og deretter teste modellen. Dette gjøres ved å dele opp datasettet i tre mindre datasett, henholdsvis trenings- og valideringssett benyttet til

trening og testsett benyttet til å teste modellen. Senere i oppgaven, ved introduksjon av datasett, vil en mer detaljert forklaring bli fremlagt.



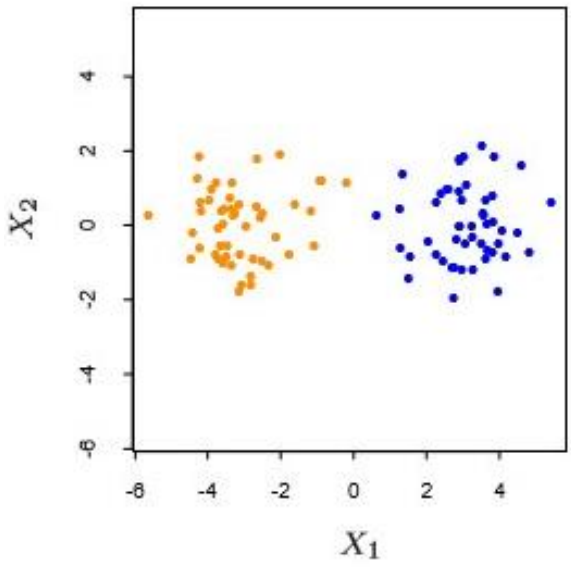
Figur 2: Grad av menneskelig innblanding ved diverse maskinlæringsteknikker (El Naqa & Murphy, 2015). Datasettet i oppgaven faller innunder «Labeled data».

Videre kan veiledet læring deles opp i to undergrupper: *Klassifisering* og *regresjon*. Klassifisering dreier seg om å la modellen predikere et utfall og plassere det i to eller flere kategorier. Ved bruk av to kategorier kalles det binær klassifisering, og et typisk eksempel kan være hvorvidt en kunde får avslag eller ikke på lånet, eller som en del av oppgavens tilfelle, hvorvidt en kunde avgår eller ikke. I treningsprosessen vil modellen få tilgang til eksempler i begge gruppene, samt tilhørende karakteristikker den benytter for å foreta prediksjonene. Regresjon dreier seg om å predikere verdier utenfor forhåndsbestemte kategorier. Dette kan eksempelvis være dersom en ønsker å predikere inntekt eller pris på bolig. Algoritmene vil i slike tilfeller forsøke å finne sammenhenger mellom uavhengige variabler (x_i) og avhengig variabel (y).

2.3 Ikke-veiledet læring

Hovedfokuset for oppgaven vil som nevnt være veiledet læring, men noe ikke-veiledet læring vil også bli benyttet. Ikke-veiledet læring er en form for maskinlæring som forsøker å gruppere observasjoner med lignende karakteristikker, og skiller seg hovedsakelig fra veiledet læring ved

at målvariabel i nevnte tilfelle ikke er til stede. En av de mest utbredte metodene for ikke-veiledet læring er clusteranalyse, som innenfor markedsføring i lang tid har vært et viktig verktøy for kundesegmentering (Punj & Stewart, 1983). I figur 3 vises hvordan metoden har gruppert et sett av observasjoner inn i to grupper med lignende karakteristikker.



Figur 3: Gruppering fra Cluster-analyse (Hastie, Tibshirani, & Friedman, 2009). Gruppe 1: Oransje. Gruppe 2: Blå.

2.4 Bias-varians tradeoff

Total error er sentralt blant begrepene diskutert i dette delkapittelet, og er forklart ved følgende formel:

$$Err(x) = (E\left[\hat{f}(x)\right] - f(x))^2 + E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right] + \sigma_e^2 \quad (1)$$

$$Err(x) = Bias^2 + Varians + Ureduserbar\ Error$$

Bias, varians, og tradeoff mellom disse, er viktige aspekter å forstå for hvordan man kan konstruere en best mulig maskinlæringsmodell som hverken *over*- eller *underfitter* (Hastie et al., 2009). Dersom en modell har høy bias vil den generalisere dataen for mye og dermed underfitte modellen. Høy varians derimot vil ikke generalisere dataen og vil videre føre til at modellen overfitter. For å spesifisere:

Underfitting skjer når en modell ikke klarer å plukke opp mønstrene som ligger i dataen.

Overfitting skjer når modellen plukker opp støy sammen med mønstrene i dataen.

Hva er bias?

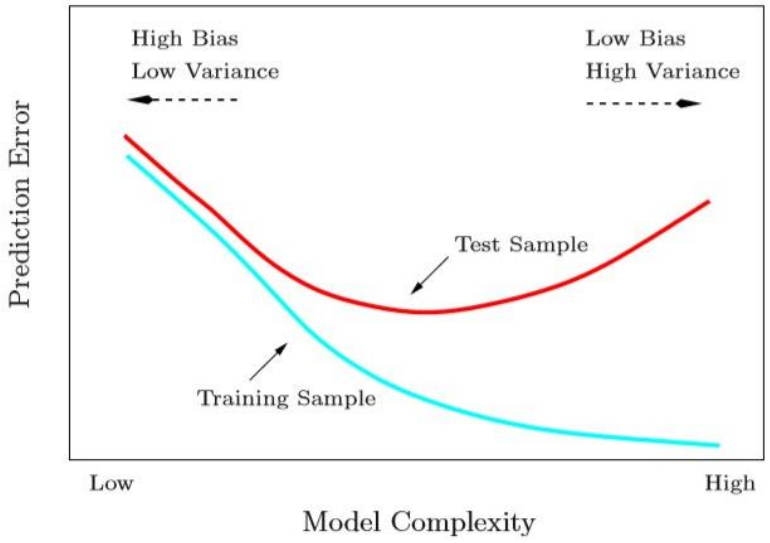
Bias er forskjellen mellom den gjennomsnittlige prediksjonen til modellen og den riktige verdien for målvariabel. Enkle modeller har typisk høy bias og gir lite oppmerksomhet til treningsdataen (Hastie et al., 2009). Følgelig gir dette høy error på trenings-, validerings-, og testsettet.

Hva er varians?

Varians er variasjonen i modellens prediksjoner for et gitt datapunkt som forteller om spredningen til dataen. Dersom en modell har høy varians, vil den se veldig nøye på treningsdataen og dermed ikke generalisere med hensyn på ny data. Dette resulterer i at ytelsen til modellen er svært god på treningssettet, men gir høy error på testsettet.

Hvorfor er det en tradeoff mellom bias og varians?

En tradeoff mellom de to finner sted fordi bias reduseres med økt modellkompleksitet, mens varians til motsetning øker. En modell kan ikke bli mer kompleks og mindre kompleks på samme tid, og derfor eksisterer det en tradeoff mellom dem. Lav bias fører til høy varians, og motsatt. Når modellen utvikles er det viktig å finne en god balanse dem imellom, slik at total error blir minimert. Ved en optimal balanse vil det føre til at modellen hverken over- eller underfitter. Figur 4 viser hvordan dette typisk kan se ut. Optimalt punkt vil være der *test sample* ikke lenger reduserer error.



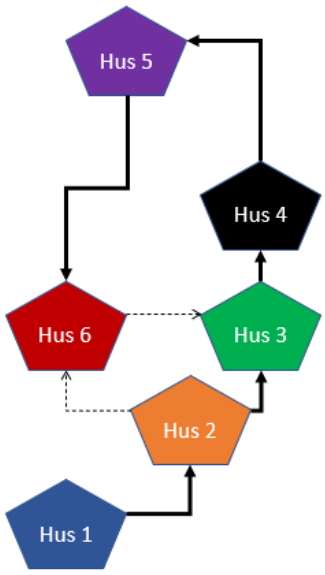
Figur 4: Hvordan modellkompleksitet typisk påvirker bias og varians for et trenings- og testsett (Hastie et al., 2009).

2.5 Gradient boosting

Maskinlæringsmodellene benyttet senere i oppgaven bygger alle på gradient boosting, og en forklaring av hvordan en slik algoritme opererer vil derfor bli fremlagt i dette delkapittelet. Algoritmen for et klassifiseringsproblem er presentert i vedlegg 12.1.1 (s. 55-56). Gradient boosting er en maskinlæringsteknikk som benytter seg av boosting for å løse regresjons- eller klassifiseringsproblemer (Friedman, 2001). Under boosting blir modellene laget sekvensielt, hvor gamle modeller lærer fra feil gjort i tidligere iterasjoner. Samtidig blir svake lærere, definert lik en variabel som gir error marginalt bedre enn tilfeldig gjetting, omgjort til sterke lærere. Fordi de lærer fra tidligere feil tar det mindre tid, og færre gjennomganger, å komme til prediksjoner som er gode enn ved bruk av andre teknikker. Av samme grunn er de også svært utsatt for overfitting da de i teorien kan fortsette å lære alt det er å kunne fra et treningssett, inkludert unødvendige sammenhenger som ikke kan generaliseres. Stoppkriterier blir derfor sentralt å benytte for å hindre at det hentes støy fra treningssettet.

Videre regnes gradient boosting som en grådig algoritme. Hva som menes med dette er at modellen følger en problemløsningsmetode som ved hvert enkelt steg finner det lokale optimale valget, med mål om å finne en global optimal løsning. Typisk fører det til at en global optimal løsning ikke blir funnet, men løsningen som blir utregnet er ofte relativt nærme uten å bruke uforsvarlig mye tid.

Tenk et pizzabud som skal levere pizza til boliger ved å kjøre kortest mulig rute totalt. Måten en grådig algoritme ville løst dette problemet på er at den til enhver tid ville valgt det nærmeste huset for neste levering. En slik løsning begrunnes med at neste hus alltid vil være det lokale optimale valget. Pizzabudet vil ikke nødvendigvis finne den korteste ruten ved denne strategien, men ofte vil løsningen bli tilnærmet optimal. Videre vil det være mye enklere og raskere å utregne enn om alle muligheter skulle blitt analysert før leveringen starter. Når grådige algoritmer løser alt fra enkle til mer kompliserte problemer, blir metoden ovenfor benyttet, noe som gjør modellene vesentlig raskere enn om det globale optimum



Figur 5: En grådig algoritmes tilnærming til pizzabudeksempelet. Grådig algoritme følger heltrukken linje. Optimal løsning følger stiplet linje fra hus 2 til hus 3, for deretter å følge heltrukken linje til hus 5.

skulle blitt utregnet til enhver tid. I figur 5 vises en visualisering av pizzabudeksempelet. Heltrukken linje viser ruten en grådig algoritme ville valgt, mens den stiplede linjen viser det optimale valget. Ved optimal løsning blir heltrukken linje mellom hus 5 og 6 naturligvis sløyfet.

Gradient boosting benytter seg av én forhåndsbestemt *loss*-funksjon som den forsøker å minimere, blant annet ved å oppdatere prediksjonene etter hver gjennomgang. Måten det gjøres på er at modellen finner residualene til prediksjonene og forsøker deretter å finne et mønster for å tilpasse modellen. Prosessen blir gjort mange ganger for å finne mønstrene og styrke modellen. For hver gjennomgang blir de svake prediksjonene marginalt styrket, og når residualene ikke lenger har noe mønster som kan bli modellert, stopper modelleringen av residualer.

Når trærne bygges, gjøres dette typisk *level-wise*. Det vil si at alle bladene utvides i dybden først. For en gitt dybde vil alle bladene splittes for den gitte dybden, før det deretter går til neste steg. En annen tilnærming, kalt *leaf-wise,* er også til tider benyttet, og vil bli diskutert senere ved algoritmen det gjelder.

Figur 6: Hvordan trærne blir bygget for en algoritme som benytter level-wise vekst vs. leaf-wise vekst (Keitakurita, 2018).

2.6 Tolkning

En viktig del av maskinlæring handler om å kunne forstå modellene, og i dette delkapittelet vil derfor tolkning være tema. Ved enkelte tilfeller vil man likevel følge prinsippet om å forholde seg til gode prediksjoner uten å gå i dybden på hvorfor modellen har kommet frem til resultatet. Dette gjelder spesielt tilfeller der utfallet ikke har stor betydning, som ved prediksjon av filmpreferanser, og det kan dermed tidvis være en grei tilnærming. Man kan spare store mengder tid på å la være å forstå grunnlaget til modellen, da utfallet i seg selv ikke skader nevneverdig. Dersom utfallet derimot vil føre til at enkelte parter vil kreve en forklaring, som ved diagnostisering av sykdom eller forstå hvorfor kundeavgang forekommer, kan man bli nødt

til å forstå hvorfor modellene foretar prediksjoner på måtene de gjør. Algoritmene benyttet i maskinlæring blir stadig mer kompliserte, og dette fører ofte til hva man refererer til som «black box»-problemet (Bathaee, 2017). Hva som inngår her, er i hovedsak at man har et forhold til inngangsverdiene og utgangsverdiene, men svært lite kjennskap til de interne prosessene som fører til nevnte utgangsverdier.

Diverse metoder for å forstå maskinlæringsmodeller er derfor blitt utviklet, men hva betyr egentlig tolkning i denne sammenhengen? Tim Miller beskrev i 2019 tolkning som «graden et menneske kan forstå årsaken til en beslutning» (Miller, 2019). For å forklare hvor viktig det kan være å forstå den interne prosessen mellom inngangsverdi og utgangsverdi, vil det legges frem to praktiske eksempler. Maskinlæring er mye brukt innenfor utviklingen av selvkjørende biler, blant annet ved bildegjenkjenning av objekter bilene møter langs veien, som syklister. Slike modeller bør ha god kjennskap til karakteristikker som tilhører en sykkel for å kunne gjenkjenne objektet. Enkelt forklart kan dette være at modellen gjenkjenner to hjul og et styre, og dermed anslår høy sannsynlighet for at det er en sykkel, og helt riktig unngår objektet. Et scenario kan dog oppstå der hjulene er dekket til, for eksempel ved store sykkelvesker. For oss mennesker vil det fremstå som åpenbart at det stadig er en sykkel det er snakk om, men modellen kan legge så mye vekt på at to hjul er helt avgjørende for at objektet er en sykkel, og dermed kategorisere feil. Slike feilkategoriseringer kan i verste fall føre til fatale ulykker, og en forståelse av hvorfor feilene oppstår blir dermed viktig for å kunne forhindre lignende episoder i fremtiden. For å sikre at modellen vektlegger de riktige tingene, kan man se på et annet velkjent eksempel som dreier seg om å skille mellom husky og ulv, to raser som i stor grad ligner på hverandre (Applied Innovation, 2017). Modellen fikk tilgang til utallige bilder av de to mulige utfallene, og ut ifra resultatene eksempelet viser til, virket modellen godt til sitt formål. Problemet oppstod dersom man viste maskinen bilde av en husky ute i vilt landskap, da ble den umiddelbart kategorisert som ulv. Det viste seg nemlig at de viktigste karakteristikkene som ble vektlagt til å skille de fra hverandre, ikke handlet om å identifisere ansikt og kropp, men heller omgivelsene rundt i bildet. Til tross for at modellen kunne vise til sterke resultater, var den trent opp på feil grunnlag og var i praksis helt ubrukelig.

Disse eksemplene gir et bilde på hvorfor tolkning er viktig når man snakker om maskinlæring, både for å kunne forhindre at modellen foretar samme feil gjentatte ganger, men også for å bekrefte grunnlaget den er trent på. For å kunne forklare modellene benyttet i denne oppgaven, som i stor grad er kompliserte og definitivt faller innenfor «black box»-kategorien, blir det benyttet en relativt ny metode for tolkning kalt SHAP.

2.6.1 SHAP

SHAP, eller «SHapley Additive exPlaination», er en metode utledet av Lundberg og Lee i 2016 (Lundberg & Lee, 2016). Metoden baserer seg på Shapley-verdier publisert av Lloyd Shapley tilbake i 1953. SHAP benyttes for å forklare hvordan modellen har kommet frem til individuelle prediksjoner basert på variablenes bidrag i hvert enkelt tilfelle. Før SHAP kan forklares, er man nødt til å ha kjennskap til hva Shapley-verdier er.

Shapley-verdier er et konsept innenfor spillteori for å tildele utbytte til hver enkelt spiller basert på bidrag til totalt utbytte. Spillerne danner koalisjoner og mottar utbytte basert på samarbeidet de har dannet. Innenfor feltet maskinlæring er nevnte spillere, spill og utbytte omgjort til å gjelde forskjellige deler av prediksjonsprosessen. Spillet blir definert som den spesifikke oppgaven algoritmen har fått i oppdrag å løse, spillerne er verdi relatert til hver enkelt variabel, og utbytte vil være den faktiske prediksjonen for en gitt observasjon minus gjennomsnittlig verdi for hele datasettet, kalt *base value*. Med koalisjon menes en samling av flere variabler. Shapley-verdien vil være det gjennomsnittlige marginale bidraget for en bestemt variabel på tvers av alle koalisjoner. I praksis betyr det at man foretar en prediksjon inkludert variabelen man ønsker å teste, deretter uten samme variabel, og ser på det marginale bidraget ved å ha den med. Dette repeteres for samtlige koalisjoner. SHAP implementerer denne logikken til modellene og henter ut bidrag for hver variabel. I vedlegg 12.1.2 (s. 57) kan den matematiske utregning av henholdsvis Shapley-verdier og SHAP leses.

De positive sidene ved SHAP er at metoden tilbyr både lokal og global tolkning. Som nevnt kalkuleres verdiene for hvert tilfelle, men de kan også summeres for å danne en oversikt over globalt bidrag til modellen. Kombinert med et solid matematisk grunnlag gir dette en fullstendig metode for å forstå modellene på en intuitiv måte. Noe negativt er dog hvor treg prosessen kan

være ettersom antallet variabler øker. Antall sammensetninger øker eksponentielt med antall variabler og fører i de fleste tilfeller til en estimert utregning av SHAP. Estimert verdi skyldes at man i mange tilfeller ved bruk av store datasett ikke benytter hele datasettet, men skalerer det ned slik at prosessen tar kortere tid.

2.7 Kundeavgang som forretningsproblem

Ved prediksjon av kundeavgang er det økonomiske perspektivet essensielt for å kunne koble modellen sin ytelse opp mot forretningsverdien den gir bedriften. Det er bred enighet om at det er langt mer kostbart å tilegne seg nye kunder enn å beholde gamle. Dette vist gjennom tidligere studier hvor funnene er at kostnadene kan være opp til 12 ganger høyere (Günther, Tvete, Aas, Sandnes, & Borgan, 2014).

En studie gjort av Forbes viser til at bedrifter som har styrket budsjettet sitt for kundebevaring over de siste 3 årene, har hatt en 200 % større sannsynlighet for å øke markedsandeler sammenlignet med de som prioriterer anskaffelse av nye kunder (Chylinski, 2016). Dette viser til at god kundelojalitet knyttes tett opp mot bedrifters utvikling, og et målrettet fokus er dermed viktig for økt inntjening og videre vekst. Selv om fordelene ved kundebevaring er godt kjent, er det fortsatt mange bedrifter som ikke klarer å se sammenhengen mellom bevaring av kunder og påvirkningen dette har på inntjening, og derfor nedprioriterer kundebevaring fremfor tilegning av nye kunder (Chylinski, 2016).

For å forstå det virkelige tapet av inntekt ved kundeavgang er kundenes livsløpsverdi, og ikke bare verdien til kundene det inneværende året, viktig. Dersom en kunde mistes tidlig, kunne den potensielle inntekten fra kundeforholdet vært betydelig større dersom kunden hadde blitt bevart. Alle kunder vil til slutt avgå bedriften av naturlige årsaker, men desto lengre kundeforholdene varer desto større er potensialet for god inntjening fra kunden.

Kundens livsløpsverdi kan defineres som det totale finansielle bidrag til bedriften, nærmere bestemt inntekt fratrukket kostnad gjennom hele perioden vedkommende kjøper tjenester eller varer fra bedrift (Farzanfar & Delafrooz, 2016).

Kundens livsløpsverdi blir utregnet ved hjelp av følgende formel:

$$CLV = \sum_{t=1}^{T} \frac{Inntekt}{(1+d)^t} - \sum_{t=1}^{T} \frac{Kostnad}{(1+d)^t} \quad (2)$$

Her blir forventede inntekter og kostnader regnet ut over kundens livsløp, hvor summen blir diskontert med en passende diskonteringsrente for videre perioder. Senere i oppgaven vil utregninger relatert til formelen bli presentert.

3 Data

3.1 Opprinnelig datasett

Det opprinnelige datasettet er konstruert av samarbeidspartner Frende Forsikring og inneholder totalt 183 645 observasjoner og 41 kolonner (inkludert indeks). I tabell 1 vises en grov oppsummering av hvilke steg som er gjort fra opprinnelig datasett til de benyttet for trening og testing av modeller. En mer detaljert forklaring av databehandling er å finne i vedlegg 12.2 (s. 58-62).

Steg	***Observasjoner***	***Variabler***	***Type: Int/float***	***Type: Kategori***
Opprinnelig datasett	183645	41	32	9
Fyller inn tomme celler	183645	41	32	9
Konstruerer nye variabler	183645	45	36	9
Kunder 2019 fjernes	180709	45	36	9
Feil i alder	180705	45	36	9
Dropper variabler	180705	29	24	5
Legger indeks til akse	180705	28	23	5
Gjør om datatyper	180705	28	19	9
Omgjør målvariabel 2	55503	28	19	9
Omgjør målvariabel 3	28238	28	19	9
Datasett 1	*180705*	*28*	*19*	*9*
Datasett 2	*55503*	*28*	*19*	*9*
Datasett 3	*28238*	*28*	*19*	*9*

Tabell 1: Grov beskrivelse av hvilke steg som er gjennomført fra opprinnelig datasett til de endelige datasettene benyttet i oppgaven.

Noe bakgrunnsinformasjon relatert til uttrekket kan likevel være nyttig for leser å være klar over. Først og fremst mangler dataen tidsstempel fra når de er trukket ut fra Frendes database. Det betyr at samtlige observasjoner fremstår som at de har samme tidspunkt, selv om de i realiteten er hentet ut på forskjellig tid. Variablene er en kombinasjon av uttrekk et halvt år og én måned før det aktuelle tidspunktet[1], der majoriteten er fra et halvt år i forveien. Sannsynligheten hentet fra prediksjonene antas å være rimelig stabile under tidsperioden dataen stammer fra.

[1] For en kunde med avgang i august 2016 vil majoriteten av variablene relatert til kunden være hentet ut i februar 2016. Resterende vil være hentet ut i juli 2016.
For en kunde som fortsatt er medlem vil aktuelt tidspunkt være da datasettet ble satt sammen, altså desember 2019. Majoriteten av variablene for disse kundene stammer derfor fra juni 2019, mens resterende er fra november 2019.

I tillegg omfatter datasettet kun den private delen av markedet. Dette gjelder da enkeltpersoner som er eller har vært kunde hos Frende Forsikring, bedriftskunder er ikke med. Hva gjelder kunder med livsforsikring er ikke disse tatt med om det er deres eneste forsikring. Dersom de i tillegg har andre typer forsikring, som forsikring av hus, vil de dog være inkludert og det vil bli informert om antall registrerte livsforsikringer.

I vedlegg 12.2.1 (s. 58-59) kan det leses grunnlaget for at nylig registrerte kunder ikke er tatt med i endelig datasett. Det følges en antagelse om at slike kunder har samme handlingsmønster som kunder med relativt lik medlemstid, og ved nye tilfeller vil disse bli gruppert sammen.

3.2 Oppdeling av datasett

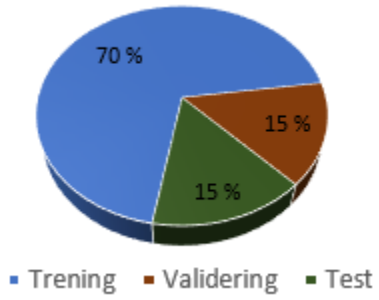
For å få en unbiased og best mulig evaluering av modellen må den testes på et ukjent datasett. I tillegg til datasettene benyttet i treningsprosessen, henholdsvis trenings- og valideringssett, er derfor et testsett blitt konstruert. De tre settene utfyller hver sin spesifikke oppgave. Treningssettet er det som benyttes til å faktisk trene modellen. Modellen har tilgang og lærer fra denne delen av dataen.

Valideringssettet benyttes i kombinasjon med tidlig stopping. Ved trening av modell vil det testes kontinuerlig opp mot både trenings- og valideringssett. Når resultater fra valideringssett ikke lenger forbedres, over et forhåndsdefinert antall iterasjoner, vil treningsprosessen avsluttes. Dette går tilbake til forklaringen under bias-varians tradeoff, figur 4, og gjør at modellen i større grad unngår overfitting. Uten nevnte metode risikerer man at treningsprosessen blir for grundig, og med det fanger opp støy, slik at modellens evne til å generalisere blir svekket.

Testsettet er data modellen, fram til den er ferdig trent og hyperparameterne justert, aldri har sett før. Et slikt datasett vil gi en unbiased evaluering av modellen, og er følgelig hva som vil benyttes når resultatene for oppgaven legges frem.

Det er ingen fastsatt standard på antall observasjoner som burde være i de ulike settene, men det er viktig at det er en stor nok andel i samtlige datasett til å kunne forklare variablenes spredning, kalt *feature space*. Normalt er likevel å benytte en 70-, 15-, 15-prosent spredning

mellom datasettene og er følgelig hva som er valgt for oppgaven. Det vil si at 70 % av observasjonene er å finne i treningssettet, 15 % i valideringssettet og 15 % i testsettet. En slik inndeling er gjort ved tilfeldig utvalg.



Figur 7: Oppdeling av datasett. 70 % i treningssett og 15 % i validerings- og testsett.

4 Metode

4.1 Forretningsverdi av kundebevaring

Verdien av en redning er bestående av flere ulike aspekter som forklart i delkapittel 2.7. For å beregne den gjennomsnittlige inntekten og kostnaden Frende vil ha for en vellykket redning, er det flere steg som må regnes ut. Kalkulasjonen begynner med å regne ut inntektene tilknyttet en redning. Utgangspunktet er at dersom en kunde avgår vil verdien være lik 0 kr, og ved en redning vil en positiv verdi fremkomme. Ved en slik tilnærming vil den ekstra verdien en redning tilfører bli belyst, i motsetning til hvilken verdi som går tapt ved kundefrafall, som kunne vært en alternativ fremgangsmåte. Likevel anses det som mer relevant å se hvilken verdi det har å forlenge kundeforholdene til avgangskunder, og derfor er dette i fokus. De ulike stegene som ligger til grunn for verdien vil nå bli lagt frem.

Utregningen for inntekt starter ved at den gjennomsnittlige premien per polise blir utregnet fra datasettet om kundene.

$$Gjennomsnittlig\ premie\ per\ polise = \frac{1\ 903\ 436\ 245}{601\ 206} = 3\ 166\ kr\ (3)$$

Fra formelen kommer det frem at gjennomsnittlig premie er lik 3 166 NOK. Neste steg vil være å finne hvor mange poliser kundene har i gjennomsnitt. Utregningen skjer ved å ta antall poliser totalt fordelt på antall kunder i datasettet.

$$Gjennomsnittlig\ antall\ poliser = \frac{601\ 206}{183\ 646} = 3{,}27\ poliser\ (4)$$

Når steg én og to er fullført kan forventet gjennomsnittlig årlig inntekt beregnes ved å multiplisere tallene sammen. Vi ender på 10 365 NOK.

$$Gjennomsnittlig\ årlig\ inntekt\ per\ kunde = 3\ 166\ kr * 3.27\ poliser = 10\ 365\ kr\ (5)$$

Videre må kostnadene beregnes for å vise til profitten hver kunde innbringer årlig. I forsikringsbransjen er *combined ratio* en vanlig betegnelse som brukes for å vise hvilken kostnadsprosent, erstatning ved skade og annet, selskapene har for sine kunder i forhold til premieinntektene (Hayes, 2019). Frende har foreslått en forventet combined ratio på 95 %, og

dette er derfor brukt. De gjennomsnittlige kostnadene per kunde blir dermed som følger:

$$Gjennomsnittlig\ årlig\ kostnad\ per\ kunde = 10\ 365\ kr * 0.95 = 9\ 847\ kr\ (6)$$

Det siste som må utregnes før det konkluderes med gjennomsnittlig livsløpsverdi, er forventet lengde av kundeforhold etter vedkommende er blitt reddet. Fra Frendes kundedata fremkommer en årlig avgangsprosent på om lag 15 %. Ved å ta den inverse avgangssannsynligheten blir gjennomsnittlig levetid på et kundeforhold tilnærmet lik 7 år.

$$Gjennomsnittlig\ levetid\ på\ kundeforhold = \frac{1}{0.15} = 6.67\ år\ (7)$$

Da kundene allerede har vært medlem en periode før redning kan finne sted, justeres tallet noe ned, til 4 år. Som en del av redningsprosessen vil det være logisk å anta at kundene får bedre avtaler da insentiver ofte må til for at noen skal endre mening. Det kan argumenteres med at dette restarter livsløpet til kunden, men forholdet skaleres likevel ned, slik at verdi ikke overestimeres.

Fra Frendes årsrapport kommer ingen diskonteringsrente frem. Tilnærming har dermed blitt å se på bransjens standard. Ved å se på årsrapporter til andre forsikringsselskap i Norge og Skandinavia er funnet at diskonteringsrentene ligger mellom 7.5 % og 8.4 % (Gjensidige Forsikring ASA, 2019) (Insr Insurance Group ASA, 2019) (Topdanmark Forsikring A/S, 2019). Grunnet de marginale forskjellene antas det at Frende har en diskonteringsrente i nærheten av de andre forsikringsselskapene, og et gjennomsnitt på 8 % er brukt ved videre utregning.

$$CLV = \sum_{t=1}^{T=4} \frac{10\ 365\ kr}{(1+0{,}08)^t} - \sum_{t=1}^{T=4} \frac{9\ 847\ kr}{(1+0{,}08)^t} = 1\ 716\ kr\ \ (8)$$

Kundenes videre livsløpverdi estimeres til 1 716 NOK, men for å komme frem til den virkelige verdien ved å redde en kunde må også kostnaden for det enkelte redningsforsøket tas høyde for. Frende har gitt et estimat på deres kostnader lik 500 NOK. Dermed blir den

gjennomsnittlige verdien for en reddet kunde justert for denne kostnaden, og ender opp på 1 216 NOK.

$$Verdi\ for\ reddet\ kunde = 1\,853\ kr - 500\ kr = 1\,216\ kr \quad (9)$$

Slik Frende har operert i forhold til redning av kunder fram til nå, har gjort at de reddet 30 % blant alle de forsøkte å redde. Dette tilsier at for 70 % av redningsforsøkene har de kun realisert kostnaden som viser til et redningsforsøk, uten å få noen videre profitt fra kundene. Gitt denne informasjonen kan den gjennomsnittlige verdien per redningsforsøk utregnes for å gi en baseline å vurdere prediksjon 3, med målvariabel «Reddet», opp mot.

$$Gjennomsnittlig\ verdi\ per\ redningsforsøk\ uten\ modell = 0.3 * 1216 - 0.7 * 500 = 15\ kr \quad (10)$$

Til tross for at *verdi for reddet kunde* er relativt høy blir bedriftens gjennomsnittlige realiserte gevinst liten grunnet lav presisjon blant redningsforsøkene. En modell som øker presisjonen vil derfor være ønskelig, slik at verdi tilknyttet hvert enkelt redningsforsøk blir høyere. Verdt å merke seg er at verdien for et redningsforsøk er veldig sensitiv til selv små endringer innenfor kostnadsprosent eller antatt videre livsløp. For ikke å overestimere gevinsten er derfor, som nevnt, videre livsløp blitt nedjustert.

4.1.1 Kostnadsmatrise

Resultatene fra utregningene i forrige delkapittel benyttes videre for en kostnadsmatrise som viser verdiene til de ulike utfallene til prediksjon 3, reddet eller ikke. Matrisen er konstruert fra en *confusion matrix* lik den presentert i tabell 2 på neste side. For å forstå den er det viktig å ha et forhold til TN, FN, FP og TP. Disse står for henholdsvis true negative, false negative, false positive og true positive. True negative viser til tilfeller av klasse 0 korrekt gjenkjent av modellen. False negative blir dermed tilfellene som faktisk er av klasse 1, men modellen klassifiserer til 0. For true positive og true negative gjelder samme logikk. True positive er korrekte predikeringer av klasse 1, mens false positive er observasjoner feilaktig predikert til klasse 1. Tabellen viser en oversikt over hvor disse plasseres.

REELL KLASSE	PREDIKERT KLASSE	
	0	**1**
0	TN	FP
1	FN	TP

Tabell 2: Oversikt over hvor de ulike prediksjonene blir plassert i en confusion matrix.
True negative (TN): Korrekt prediksjon av klasse 0
False negative (FN): Faktiske tilfeller av klasse 1 feilaktig lagt i klasse 0
True positive (TP): Korrekt prediksjon av klasse 1
False positive (FP): Faktiske tilfeller av klasse 0 feilaktig lagt i klasse 1

Kostnadsmatrisen gir grunnlag for *cutoff* til modellene for prediksjon 3, hvor den vil bli satt der hver enkelt modell gir størst økonomisk gevinst. Med cutoff menes alt over gitt verdi legges til klasse 1, mens alt under legges til klasse 0. Eksempelvis vil en modell med cutoff lik 0.5 plassere samtlige prediksjoner med en sannsynlighet over 50 % i klasse 1, resten 0. Endelig verdi tilknyttet prediksjon 3 vil bli kommentert i delkapittel 6.3.

REELL KLASSE	PREDIKERT KLASSE	
	0	**1**
0	0 kr	- 500 kr
1	0 kr	1 216 kr

Tabell 3: Kostnadsmatrise med tilhørende verdier for prediksjon 3, målvariabel «reddet». Korrekte prediksjoner lagt i klasse 1 vil bli belønnet med 1216 kr. Dersom modell predikerer feil i klasse 1 vil kun kostnad bli realisert, og modellen straffes -500 kr. Observasjoner lagt i klasse 0 vil ikke bli forsøkt reddet. Ingen gevinst eller kostnad er dermed assosiert med disse tilfellene.

Det har ikke blitt konstruert noen kostnadsmatrise for prediksjon 1 og 2 da disse er steg i en helhetlig prosess som ender med redning eller ikke. Mye av verdien tilknyttet prediksjonene ligger i innsikten modellene gir om kundene, i form av blant annet SHAP-verdier. Den faktiske økonomiske verdien er vanskelig å tallfeste, og derfor er heller ingen konkrete tall benyttet.

4.2 Optimering av hyperparametere

Hyperparametere referer til verdiene som kan settes på forhånd før en modell trenes, eksempelvis *max depth* og *learning rate* for modellene benyttet i oppgaven. Dette skiller seg fra parametere som er verdiene modellene lærer under treningsprosessen.

For å kunne fremstille de beste resultatene for et gitt datasett, er det viktig å velge de riktige hyperparameterne. Den enkleste metoden for å fremstille disse er å sette hyperparametere basert på intuisjon og erfaring, kjøre modellen, lese av resultatene og repetere prosessen til man er fornøyd. Dette blir fort en svært unøyaktig og tidkrevende prosess, og det finnes derfor

verktøy som kan være med å forbedre prosessen. I oppgaven er *GridSearch* benyttet som optimeringsmetode.

4.2.1 GridSearch

Gridsearch er en form for hyperparameteroptimering som baserer seg på å teste et utvalg forhåndsbestemte hyperparametere. I tillegg er man nødt til å bestemme et mål modellene skal bli målt etter, som *logloss*. Deretter trenes og testes modellen opp mot en kombinasjon av samtlige hyperparametere og returnerer et utvalg av de beste.

I praksis testes det svært mange modeller med forskjellige hyperparametere og kan anses som en samlet prosess av å foreta testingen manuelt. Med andre ord kunne man like gjerne definert én og én modell med ulike hyperparametere, notert resultatet og testet videre. Gridsearch samler denne prosessen og gjør både kodestrukturen mer oversiktlig og sparer tid. For å unngå at modellen overfitter er det i tillegg innebygd *cross validation* i pakken benyttet for optimering. Dette er en metode som hele tiden tester hvor mye bedre modellen blir på et trenings- og valideringssett. Man unngår derfor at modellen lærer for mye fra treningssettet og sikrer generalisering av kunnskapen, slik nevnt i delkapittel 2.4.

Et bredt spekter av hyperparametere blir testet uten manuell koding av samtlige sammensetninger. Negativt er at metoden ikke benytter informasjon tillært fra tidligere gjennomkjøringer, slik andre mer avanserte optimeringsalgoritmer gjør *(*for eksempel *Bayesian Optimization)*. Samtlige modeller presentert senere i oppgaven er forsøkt optimert med GridSearch. Ved enkelte tilfeller viste det seg svært nyttig, mens ved andre ga det liten til ingen forbedring.

4.3 Tilnærming til modellvalg

Etter at ulike modeller er trent, må deres respektive resultater gjennomgås for å vurdere hvilken modell som er best for det aktuelle målet. I hovedsak vil det være ønskelig å gjenkjenne så mange tilfeller av klasse 1 som mulig, dette gjelder for samtlige prediksjoner. Et mål på dette er *recall*, og dermed blir det viktig å se etter høye verdier for nettopp dette målet. Det kan dog assosieres kostnader ved å predikere for mange i klasse 1, og høy *precision* blir i så måte også

viktig. Kombinert resulterer dette i F1-score, se vedlegg 12.1.3 (s. 57) for formel, som blir målet sterkest vektlagt.

Andre kriterier som vektlegges er *accuracy*, hurtighet og resultater fra kostnadsmatrise. Accuracy er den totale treffprosenten til modellen og vil i enkelte tilfeller være nyttig. Målenheten vil likevel bli sett på med kritiske øyne da den raskt kan gi misvisende informasjon. Prediksjon 1 har en fordeling innad i målvariabel lik 70/30. I teorien kan modellen da oppnå 70 % accuracy ved å plassere samtlige tilfeller i klasse 0. Isolert sett virker det som et greit mål, men i praksis vil en slik modell være ubrukelig. For prediksjon 2, der fordelingen er jevn, blir målet sterkere vektlagt.

Hurtighet vektlegges der det vises til tilnærmet identiske resultater blant de øvrige målene. Det predikeres ikke i sanntid, hvilket gjør det mindre viktig enn ved andre tilfeller. Fordelen ligger i at raske modeller er betydelig enklere å optimere. I delkapittel 4.2 ble det nevnt hvor tidkrevende en slik prosess kan være, og desto lenger tid som benyttes per gjennomkjøring, desto mer tid vil dette ta.

Spesifikt for tredje prediksjon er at en kostnadsmatrise er tilgjengelig. Her vil det være mulig å tallfeste hvordan en modell kan skape økonomisk vinning og følgelig er matrisen sterkt vektlagt i valg av modell for gjeldende prediksjon.

5 Modeller

5.1 Kandidatmodeller

For hver prediksjon ble syv maskinlæringsmodeller testet. En komplett liste, inkludert resultater, kan sees i vedlegg 12.3.1 (s. 63-64). Modellene som skilte seg ut i positiv forstand, var XGBoost, LightGBM og CatBoost. Slike modeller regnes for å være *state-of-the-art* teknikker for maskinlæring på strukturert data (de Oliveira, 2019). Det ble derfor bestemt å forkaste resterende modeller og heller rette fokus mot å optimere de tre. Videre vil nevnte modeller bli presentert i hvert sitt delkapittel. Dette innebærer i tillegg en oversikt over endelige hyperparametere benyttet.

5.1.1 XGBoost

XGBoost, Extreme Gradient Boosting, er en implementering av gradient boosting, men designet for å dytte beregningsgrensene til maskinen til det «ekstreme» for å gi en skalerbar og nøyaktig modell (Chen & Guestrin, 2016). Modellen oppnår dette ved å ta hånd om noen av de mest ineffektive aspektene ved gradient boosting. På nettsamfunnet Kaggle blir det gjennomført en rekke maskinlæringskonkurranser, og i 2015 ble XGBoost benyttet for et flertall av de vinnende løsningene, en observasjon som viser den potensielle styrken til modellen (Chen & Guestrin, 2016).

Som forklart i delkapittel 2.5 splitter gradient boosting ved lokale optimale punkt. Selv om denne fremgangsmåten er vesentlig mer effektiv enn å lete etter et globalt optimum, kan det fortsatt være svært mange splittmuligheter for hver enkelt variabel. Måten XGBoost effektiviserer dette ytterligere på, er at den først foreslår kandidatsplitter ut ifra distribusjonen i variablene, hvor numeriske variabler blir gruppert. Ved en slik måte å splitte bladene på vil ikke alle mulige splittpunkter bli evaluert, noe som gir vesentlig økt hastighet, men resultatene vil fortsatt bli tilnærmet likt som ved å evaluere samtlige muligheter.

Noe negativt ved XGBoost er dens manglede evne til å håndtere kategoriske variabler. Problemet ble løst ved å lage dummyvariabler av slike. Tilnærmingen medfører riktignok et høyere antall variabler og følgelig blir både treningstid og fremstilling av SHAP vesentlig lenger.

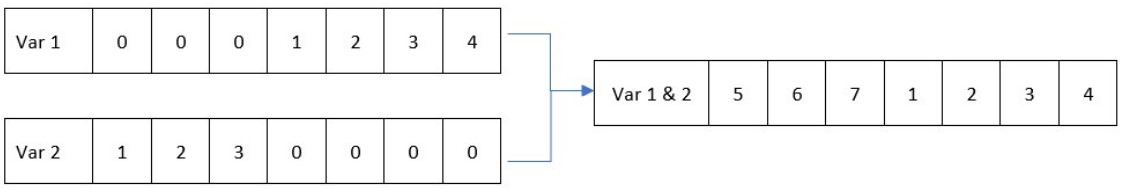
HYPERPARAMETER	PREDIKSJON 1	PREDIKSJON 2	PREDIKSJON 3
BOOSTER	gbtree	gbtree	gbtree
MAX DEPTH	10	6	7
GAMMA	1	0.5	1.2
MIN CHILD WEIGHT	1.8	2	0
LEARNING RATE	0.1	0.06	0.05
COLSAMPLE BYTREE	0.7	0.7	0.7
REG ALPHA	0.5	0.4	0.5
REG LAMBDA	2	1.8	2
NUM BOOST ROUND	5000	5000	5000
EARLY STOPPING ROUNDS	50	50	50

Tabell 4: Hyperparametere benyttet for XGBoost for prediksjon 1, 2 og 3. Hyperparametere som ikke er i listen er satt til standard for xgboost-pakken.

5.1.2 LightGBM

Som XGBoost baserer LightGBM (Light Gradient Boosted Machine) seg på gradient boosting rammeverket (Ke et al., 2017). I motsetning til gradient boosting level-wise tilnærming til å bygge trærne har modellen en leaf-wise tilnærming. Det vil si at der hvor andre boosting-algoritmer bygger bladene til samme dybde før den går videre til neste steg, bygger LightGBM bladene ved at den tar beste først, og deretter bygger i dybden for nevnte blad. Dette gjør at LightGBM kan være raskere enn andre gradient boosting modeller, men medfører økt kompleksitet og dermed økt fare for overfitting, spesielt for mindre datasett. Tuning av enkelte hyperparametere kan redusere denne risikoen, spesielt de relatert til dybde.

Videre benytter LightGBM en metode som reduserer antall variabler i datasettet, kalt *Exclusive Feature Bundling*. I et datasett vil det ofte være variabler som er gjensidig utelukkende. Med dette menes at dersom «Variabel A» er 0 vil «Variabel B» inneha en annen verdi, og motsatt. Dersom slike tilfeller oppstår, vil LightGBM gruppere to og to variabler og gi de ulike kombinasjonene unike verdier. Denne metoden kan gjøre treningen av modellen markant raskere ved at antall variabler reduseres uten å gå på bekostning av nøyaktigheten til modellen. I figur 8 nedenfor er et enkelt eksempel på en slik gruppering visualisert.



Figur 8: LightGBMs gruppering av gjensidig utelukkende variabler.

Studier viser at LightGBM ofte er den raskeste av boosting-algoritmene, noe som bringer med seg fordeler (Al Daoud, 2019). Flere hyperparametere kan testes på kortere tid, og optimerte modeller blir dermed enklere å oppnå.

Hva gjelder håndteringen av kategoriske variabler har LightGBM en innebygd funksjon i pakken, dog en relativt simpel en. Måten de kategoriske variablene transformeres på, er ved tildeling av numeriske verdier. Hvis en variabel inneholder for eksempel fem ulike kategorier, vil de bli tildelt verdier fra 0 til 4.

HYPERPARAMETER	PREDIKSJON 1	PREDIKSJON 2	PREDIKSJON 3
BOOSTING TYPE	gbdt	gbdt	gbdt
LEARNING RATE	0.15	0.05	0.07
MAX DEPTH	10	10	15
MIN CHILD WEIGHT	0.002	0.001	0.005
MIN SPLIT GAIN	0.3	0.4	0.25
N ESTIMATORS	99999	99999	99999
EARLY STOPPING ROUNDS	100	100	100
NUM LEAVES	60	40	50
REG ALPHA	0.6	0.5	0.5
REG LAMBDA	2	2	1.8
SUBSAMPLE	0.5	1	0.6
COLSAMPLE BYTREE	0.6	0.65	0.7

Tabell 5: Hyperparametere for LightGBM for prediksjon 1,2 og 3. Hyperparametere som ikke er i listen er satt til standard for lightgbm-pakken.

5.1.3 CatBoost

Av de tre kandidatmodellene er CatBoost den nyeste, og ble lansert så sent som i 2017. Modellen bygger på prinsippene fra gradient boosting, men med enkelte funksjoner som avviker fra de øvrige. I hovedsak skiller den seg ved håndteringen av kategoriske variabler og en alternativ form for boosting (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018).

CatBoost benytter en metode lignende *mean encoding* for kategoriske variabler i et datasett. Mean encoding kan forklares ved å se på den kategoriske variabelen «Distribusjonskanal» fra datasettet. I datasett 1, som ser på avgang, har 124 098 observasjoner kategorien «Eierbank», hvor 32 048 av disse er registrert med målvariabel avgang lik 1. Gjennom mean encoding vil forholdet mellom de to tallene bli utregnet for å tilegne kategorien en numerisk verdi, som for dette eksempelet ender på 0.26. Videre vil like utregninger bli gjort for de andre kategoriene.

Resultatet for «Franchise», «Frendes egne kanaler» og «Partner» blir henholdsvis 0.28, 0.43 og 0.40. Problemet rundt overfitting blir igjen sentralt da modellen får tilgang til informasjon om målvariabelen.

For å takle problemet har CatBoost en innebygd metode inspirert av måten nettbaserte algoritmer tilegner seg kunnskap ettersom nye observasjoner blir registrert (Prokhorenkova et al., 2018). Metoden går ut på å tildele et tilfeldig tidsstempel til alle observasjonene. Når de kategoriske variablene blir omgjort til numeriske, vil kun de med et tidsstempel før den aktuelle observasjonen bli benyttet. Gjennom denne metoden blir informasjonen om målvariabelen redusert, og dermed mindre utsatt for overfitting.

En utpreget styrke for CatBoost er dens svært gode standard hyperparametere. Utvikleren av algoritmen, Yandex, har forsket på dette og sammenlignet resultater fra bruken av nevnte standard hyperparametere, mot optimerte, og kan rapportere at de svært ofte blir tilnærmet identiske (CatBoost, 2020). Styrken ligger i at en slipper å bruke mye tid på hyperparameteroptimering, og følgelig kan modellen implementeres og ferdigstilles raskt. Av den grunn er de fleste hyperparameterne satt til standard for CatBoost-modellene benyttet i oppgaven. Følgende har likevel blitt spesifisert ved programmering:

HYPERPARAMETER	PREDIKSJON 1	PREDIKSJON 2	PREDIKJSON 3
BOOSTING TYPE	Plain	Plain	Plain
NUM ESTIMATORS	500	700	600
MAX DEPTH	10	12	9
LEARNING RATE	0.01	0.04	0.1

Tabell 6: Hyperparametere for CatBoost for prediksjon 1,2 og 3. Hyperparametere som ikke er i listen er satt til standard for catboost-pakken.

5.2 Endeling modell

Valg av endelig modell følger kravene satt fra delkapittel 4.3. Basert på resultatene fra kandidatmodellene, se vedlegg 12.3.1 (s. 63-64) for komplett oversikt, har maskinlæringsmodellen LightGBM blitt valgt for prediksjon 1, 2 og 3. Modellen kan skilte med de beste resultatene fra samtlige kriterier, selv om det kun er marginalt bedre. I tillegg er det den raskeste modellen. Med marginalt bedre resultater og raskere kjøretid blir det dermed vanskelig å velge noe annet som endelig modell. Lignende resultater er ikke spesielt

overraskende da maskinlæringsmodellene er basert på gradient boosting og kan anses som state-of-the-art algoritmer.

Deres nær identiske resultat kan fremme spørsmålet om andre, mer varierte, maskinlæringsteknikker burde blitt benyttet, men igjen understrekes det at andre modeller er blitt vurdert i startfasen av prosjektet. Algoritmene basert på gradient boosting utkonkurrerte de resterende og et videre fokus på andre modeller, kun for å fremme variasjon i oppgaven, ble derfor droppet.

6 Resultat fra endelig modell

I delkapittel 1.2 ble det forklart hvordan det opprinnelige datasettet ble oppdelt til å følge en prosess bestående av tre målvariabler. En komplett beskrivelse av hvordan oppdeling ble gjort kan leses i vedlegg 12.2.5 (s. 61-62). I første steg skal det avgjøres hvorvidt en kunde predikeres å avgå fra bedriften, deretter årsaken til avgang, før det til slutt predikeres hvorvidt en kunde kan reddes eller ikke. For hvert ledd er det blitt utledet en målvariabel, og i tillegg blitt konstruert et trenings-, validerings- og testsett. I dette kapittelet vil det legges frem resultater fra testsettet til hver målvariabel. Endelig modell vil på den måten få testet seg på usett data, og resultatene vil gi et mål på hvor god evnen til å generalisere kunnskap tilegnet fra treningsperioden har vært. En sammenligning av resultatene fra trenings-, validerings- og testsettet er vist i vedlegg 12.3.2 (s. 65).

Resultatene fra hvert datasett vil bli gjennomgått i detalj i delkapitlene nedenfor. I hovedsak vil følgende punkter bli presentert:

- Klassifiseringsrapport
- Confusion matrix
- ROC-AUC
- Viktige variabler og identifisering av utsatte kundegrupper

De statistiske målene er i stor grad sammensatte i den betydning at dersom én verdi øker er sannsynligheten stor for at dette går ut over et annet mål. Flere statistiske mål er dermed nødt til å være med for å skape et helhetlig bilde av modellens prestasjon. Eksempelvis ble det i delkapittel 4.3 lagt frem hvorfor accuracy ofte blir misvisende å vurdere uten andre mål ved skjevt fordelte datasett. Videre vil det være interessant å se fra en confusion matrix det faktiske antallet modellen treffer og bommer på.

Disse målene vil variere i stor grad avhengig av hvilken cutoff som er satt for modellen. Her er det tenkt noe annerledes for de ulike datasettene, der datasett 1 og 2 er tildelt en verdi satt for å optimere F1-score, mens datasett 3 er satt som følge av resultater fra en kostnadsmatrise. Etter hvert som leser kommer til punktene det gjelder vil dette bli ytterligere spesifisert.

I tillegg er det lagt ved en ROC-kurve *(Reciever Operating Characteristics)* som viser et mål for AUC *(Area Under Curve)*. En slik kurve gir en oversikt over forholdet mellom true positives og false positives på tvers av ulike cutoff-verdier.

Det vil også være gunstig å kunne si noe om hvordan modellene vektlegger ulike variabler i sine predikeringer. På denne måten kan man sikre at modellene er trent på korrekt grunnlag, men også finne typiske karakteristikker ved kundegrupper som har høy eller lav sannsynlighet for å havne i de ulike klassene. For dette benyttes SHAP-verdier hentet fra treningssettet og funnene implementeres på testsettet.

6.1 Datasett 1 – Målvariabel: Avgang

I første steg blir det predikert hvorvidt en kunde avgår eller ikke. Testsettet består av 26880 observasjoner og 28 variabler (inkludert målvariabel), med en avgangsprosent på om lag 30 %. Det er ingen konkret forretningsgevinst knyttet til riktig predikering, da mange faktorer spiller inn i etterkant med tanke på hvorvidt kunden er verdt å redde eller ikke, men det er likevel en viktig del av den helhetlige prosessen. Hvordan endelig modell har klassifisert observasjonene fra testsettet er å finne i tabell 7 og 8 nedenfor, henholdsvis fra en klassifiseringsrapport og confusion matrix.

Klassifisering	*Precision*	*Recall*	*F1-score*	*Accuracy*
0	0.82	0.91	0.86	0.79
1	0.72	0.53	0.61	

Tabell 7: Klassifiseringsrapport for målvariabel «Avgang». Cutoff benyttet er 0.5.

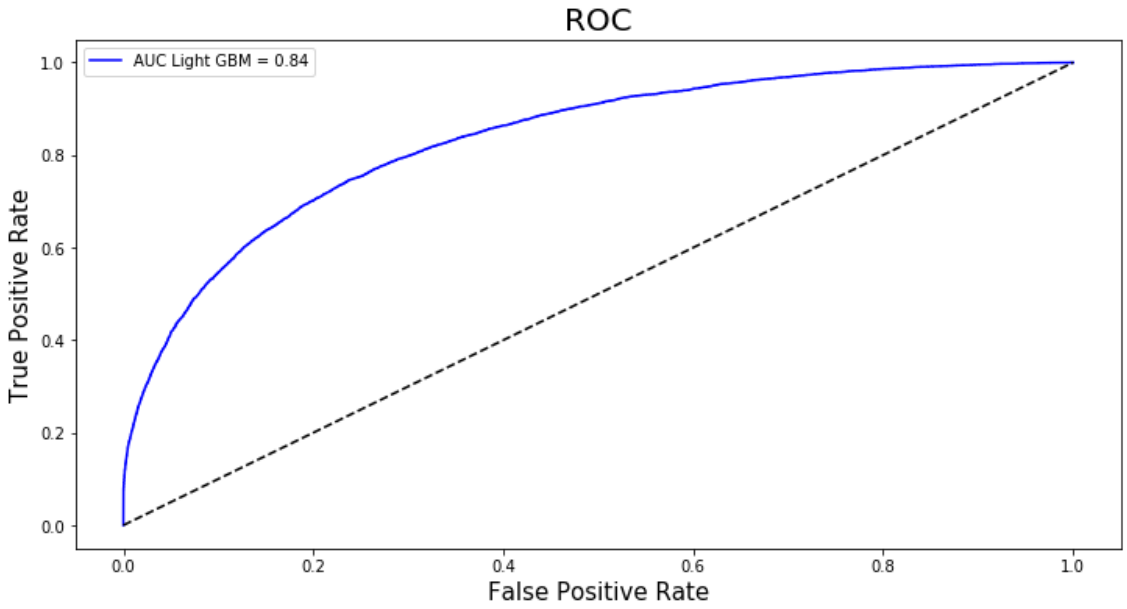
		PREDIKERT	
		0	1
REELL	0	17025	1708
	1	3829	4318

Tabell 8: Confusion matrix for målvariabel "Avgang". Cutoff benyttet er 0.5.

Modellen har en samlet treffsikkerhet på 79 %. Dette trekkes noe opp av hvor godt den klassifiserer kunder i klasse 0, et ikke uventet resultat da fordelingen mellom de to er relativt skjev. Den gjenkjenner også de aller fleste tilfellene av denne klassen, med recall på 91 %. For klasse 1 er resultatet noe dårligere, men igjen stammer dette trolig fra den skjeve fordelingen.

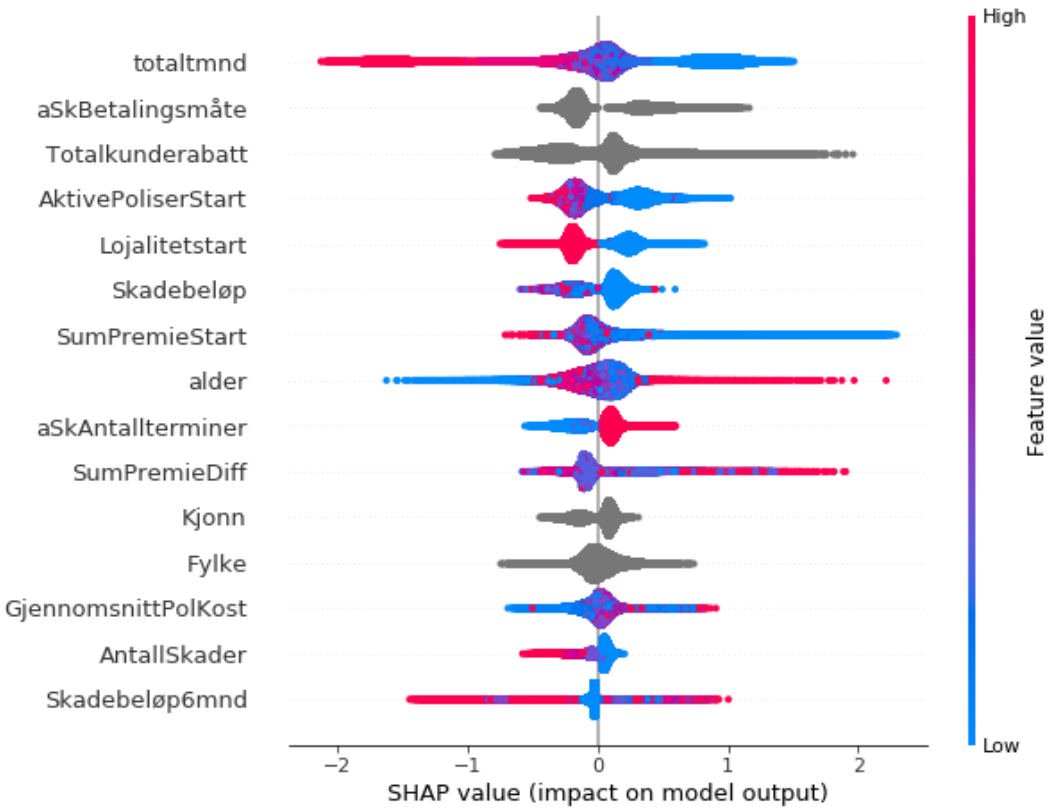
Modellen er nødt til å generalisere kunnskapen tilegnet fra treningssettet, og da det er lettere å gjenkjenne tilfeller det er flere av, blir slike resultater å forvente.

Figur 9 nedenfor viser til en AUC på 0.84 for prediksjonen. For referanse ses dette opp mot en modell som gjetter vilkårlig, og følgelig følger den stiplede linjen, eller gjenkjenner samtlige korrekt. Referansepunktene vil henholdsvis bli målt til AUC lik 0.5 og 1.



Figur 9: ROC-kurve for målvariabel «Avgang».

Ved å se på variablenes tilknyttede SHAP-verdier kan det forstås hvilke, og i hvor stor grad, modellen har vektlagt de ulike variablene. Figur 10 viser et sammendrag av de 15 viktigste variablene i synkende rekkefølge. Verdier som befinner seg på høyre side av skillelinjen i midten, påvirker mot å plassere kunder i klasse 1, mens verdier på venstre side påvirker mot klasse 0. Desto lenger ut fra midten, desto mer påvirkes utfallet. Fargene beskriver verdier innad i gitte variabler, der blå er lave verdier og en gradvis overgang til rød betyr høyere verdier. Fargekoden grå representerer kategoriske variabler. Observasjoner med lik påvirkning vil legge seg ved siden av hverandre og er grunnen til at «klumpene» i oversikten dannes.



Figur 10: SHAP-plot for målvariabel «Avgang». Variablenes viktighet er presentert i synkende rekkefølge. Blå farge representerer lav verdi innad i variabel. Rød farge representerer høy verdi innad i variabel. Skillelinjen i midten markerer hvorvidt observasjon påvirker utfallet mot klasse 0 eller 1. Observasjoner til venstre påvirker mot klasse 0 og observasjoner til høyre påvirker mot klasse 1.

Kombinasjonen av rangert viktighet og variabelverdi gjør at man kan hente mye informasjon ut av figuren. De fleste andre metoder for å tolke maskinlæringsmodeller er kun egnet til å se påvirkningskraften de ulike variablene har i modellens utfall, uten å vite om dette er positivt eller negativt. SHAP gir en oversikt over hvilken retning prediksjonene dras mot for bestemte verdier, og kan dermed benyttes videre som et verktøy for fremstilling av kundegrupper. Først og fremst er utsatte grupperinger interessante, nærmere bestemt grupper med høy avgangsprosent. Mindre utsatte grupper, med lav avgangsprosent, vil også være viktig å identifisere. Som nevnt har datasett 1 en målvariabel med omtrent 30 % i klasse 1, og det ønskes følgelig å sammenligne funn med denne verdien. Tabellen nedenfor er et resultat av informasjonen tilgjengelig fra SHAP-verdiene.

Gruppering	Kjennetegn	N	Klasse 0	Klasse 1	Andel Avgått
Gruppe 1	Lojalitetstart = 0 Totaltmnd < 2 Kunderabatt = A Antall terminer > 4	3051	1379	1672	55 %
Gruppe 2	Totalmnd > 3 Betalingsmåte = Avtalegiro Kunderabatt = C AktivePoliserStart > 3 Lojalitetstart = 1	1559	1467	92	6 %

Tabell 9: Kundegrupperinger med kjennetegn og oversikt over «Andel Avgått».

I de to gruppene introdusert fra tabell 9 kan man se verdier som kjennetegner grupper med henholdsvis høy og lav avgangsprosent. Verdien av slik informasjon knyttes hovedsakelig til bedre bruk av ressurser eller videre utvikling av modeller. Det kan eksempelvis være nyttig å flagge kunder med tilhørende verdier som trygge eller mer utsatt. Man kan dermed vite om de er verdt å følge med på videre eller ikke. Likevel er det et problem at restriksjonene fjerner store kundemasser, og grupperingene kan bli relativt små i forhold til datasettet.

6.2 Datasett 2 – Målvariabel: Nytt selskap

Målvariabel tilknyttet datasett 2 er hvorvidt en kunde avgår grunnet nytt selskap eller ikke, med en tilnærmet 50/50-fordeling. Testsettet består av 8256 observasjoner og 28 variabler (inkludert målvariabel). Statistiske mål kan sees fra tabell 10 og 11.

Klassifisering	*Precision*	*Recall*	*F1-score*	*Accuracy*
0	0.80	0.72	0.76	0.77
1	0.74	0.82	0.78	

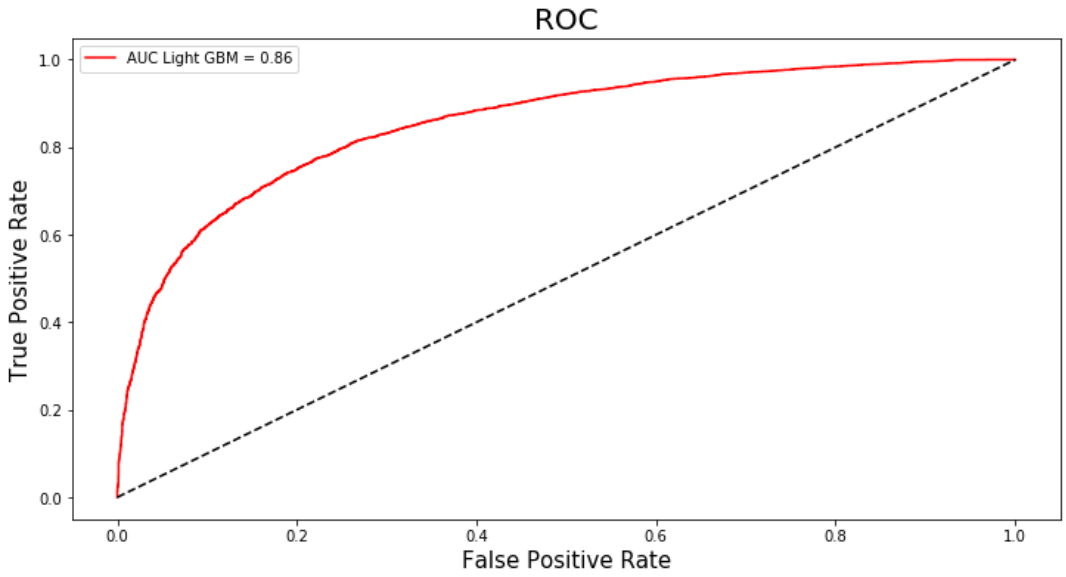
Tabell 10: Klassifiseringsrapport for målvariabel «Nytt Selskap». Cutoff benyttet er 0.4.

		PREDIKERT	
		0	1
REELL	0	2991	1159
	1	735	3371

Tabell 11: Confusion matrix for målvariabel «Nytt Selskap». Cutoff benyttet er 0.4.

De statistiske målene viser til at modellen klassifiserer betydelig flere tilfeller korrekt enn feil. Den samlede treffsikkerheten er dog lik som ved målvariabel avgang, men andel true positives er her høyere. Dette kan skyldes en mer jevn fordeling av målvariabel, da denne ligger på

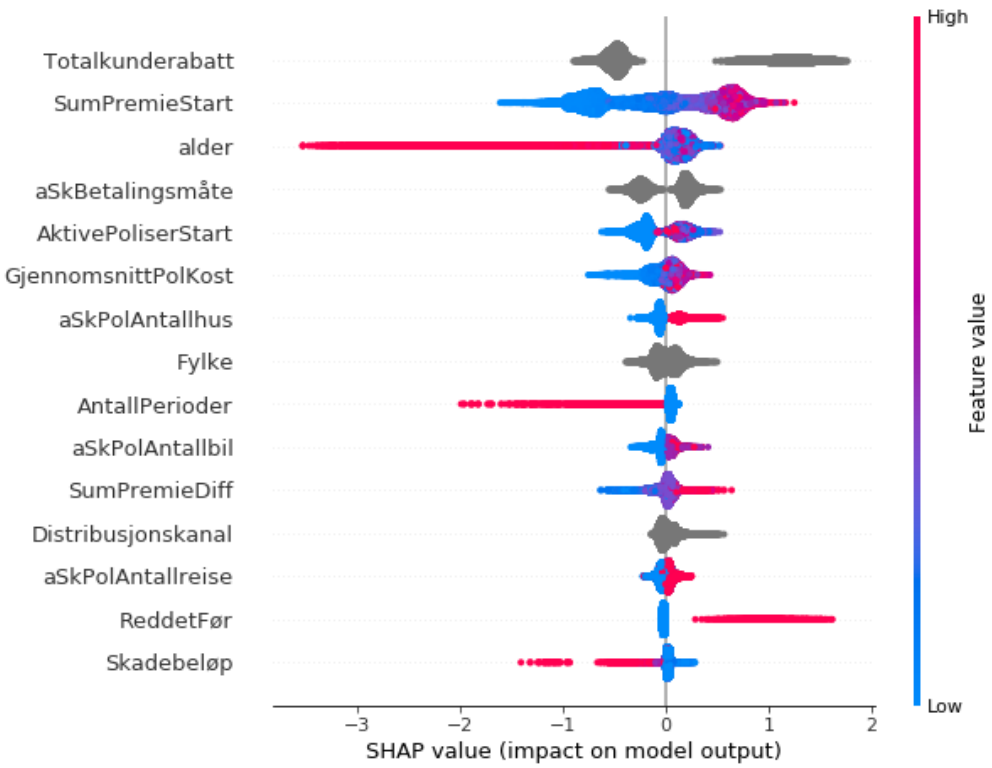
omtrent 50 % i hver klasse, og karakteristikker tilhørende de to klassene kan derfor være mer gjenkjennelig for modellen.



Figur 11: ROC-kurve for målvariabel «Nytt Selskap».

Målt AUC er i dette tilfellet 0.86, marginalt høyere enn ved prediksjon 1.

Ved å benytte SHAP-verdiene i figur 12 kommer det frem en interessant sammenheng mellom de som er reddet før og andel som forlater til fordel for et nytt selskap. Da dette er en binær variabel, vil blå verdier tilsvare 0, altså ikke til stede, og røde verdier tilsvare 1. Det kan sees en markant påvirkning mot klasse 1 dersom de er reddet før, og videre analyse tilsier at om lag 85 % av kundene som er reddet tidligere, finner seg et nytt selskap ved fremtidig avgang.



Figur 12: SHAP-plot for målvariabel «Nytt Selskap». Variablenes viktighet er presentert i synkende rekkefølge. Blå farge representerer lav verdi innad i variabel. Rød farge representerer høy verdi innad i variabel. Skillelinjen i midten markerer hvorvidt observasjon påvirker utfallet mot klasse 0 eller 1. Observasjoner til venstre påvirker mot klasse 0 og observasjoner til høyre påvirker mot klasse 1.

Variabelen tilknyttet kunderabatt viser også til et tydelig skille mellom verdiene. Kategorien innebærer at man har rabattkode A, B eller C, som gir henholdsvis 0 %, 10 % eller 17 % rabatt, avhengig av hvor mange produkter kunden har. En oversikt vises i tabell 12.

KUNDERABATT	KLASSE 0	KLASSE 1	ANDEL NYTT SELSKAP
A	3657	1838	33 %
B	349	1435	80 %
C	144	833	85 %

Tabell 12: «Andel Nytt Selskap» ved ulik kunderabatt.

Kunder med rabattkode B eller C har betydelig høyere andel som forlater til fordel for et nytt selskap. Det skal likevel sies at man er nødt til å se på dette noe kritisk. Som nevnt er rabattkodene avhengig av andel produkter hver kunde har, da de med A har færrest og de med C har flest. At de med rabattkode A har mange avganger som ikke skyldes nytt selskap, kan skyldes at behovet deres for å være kunde opphører fordi de ikke lenger trenger sin gjeldende forsikring. Eksempelvis dersom en kunde kun har registrert én forsikring hos Frende og behovet

for forsikringen opphører, ved salg av forsikret vare, vil også kundeforholdet avsluttes. Det er likevel verdt å notere den svært høye andelen med rabattkode B og C som bytter selskap ved avgang.

Videre vil det presenteres grupperinger med særegne trekk som er hentet ut fra datasettet. Da det fra et forretningsmessig perspektiv vil være spesielt viktig å hente ut kjennetegn ved gjeldende prediksjon, er det i tillegg blitt benyttet *k-means clustering* til å danne store grupperinger som enklere kan generaliseres. Selv om uthenting basert på SHAP kan finne utsatte kundegrupper, blir grupperingene raskt små, da hver restriksjon utelukker et stort antall observasjoner. Dette skjer fordi SHAP-plottet er ment til å hente ut de overordnede trendene innad i datasettet. Selv om høy alder generelt medfører lavere andel kunder som velger nytt selskap, fanges det ikke opp dersom det eksempelvis eksisterer grupperinger med høy alder som også avgår til nytt selskap.

Den fullstendige oversikten over gjennomsnittlig variabelverdi til variabler i *cluster 1* og *cluster 2* kan sees i vedlegg 12.4 (s. 66-67).

Gruppering	Kjennetegn	N	Klasse 0	Klasse 1	Andel Nytt Selskap
Gruppe 1	SumPremieStart > 8000 35 < Alder < 55 Betalingsmåte = Avtalegiro	825	93	732	89 %
Gruppe 2	SumPremieStart < 7000 Alder > 60 AktivePoliserStart < 3	801	645	156	19 %
Cluster 1	*	2217	1951	266	88 %
Cluster 2	*	1894	455	1439	24 %

Tabell 13: Kundegrupperinger og clusters med kjennetegn og oversikt over «Andel Nytt Selskap».
** Oversikt over kjennetegn finnes i vedlegg 12.4 (s.66-67).*

I tillegg er det blitt konstruert en oversikt for å vise hvordan modellen vektlegger prediksjoner ved individuelle tilfeller. Dersom en person blir flagget, vil det videre være mulig å gå inn og se hvorfor vedkommende er blitt utvalgt, og eventuelt gjøre manuell evaluering av tilfellet. Man skaper med det et samspill mellom menneske og maskin, hvilket *kan* bedre resultatene ytterligere. Figuren viser utfallet for vilkårlig valgt kunde med indeks 104. Base value, 0.52, viser til verdien modellen ville predikert uten å ta i bruk noen variabler, et mål som betyr 52 % sikkerhet for at vedkommende velger nytt selskap ved avgang, tilnærmet lik fordelingen av

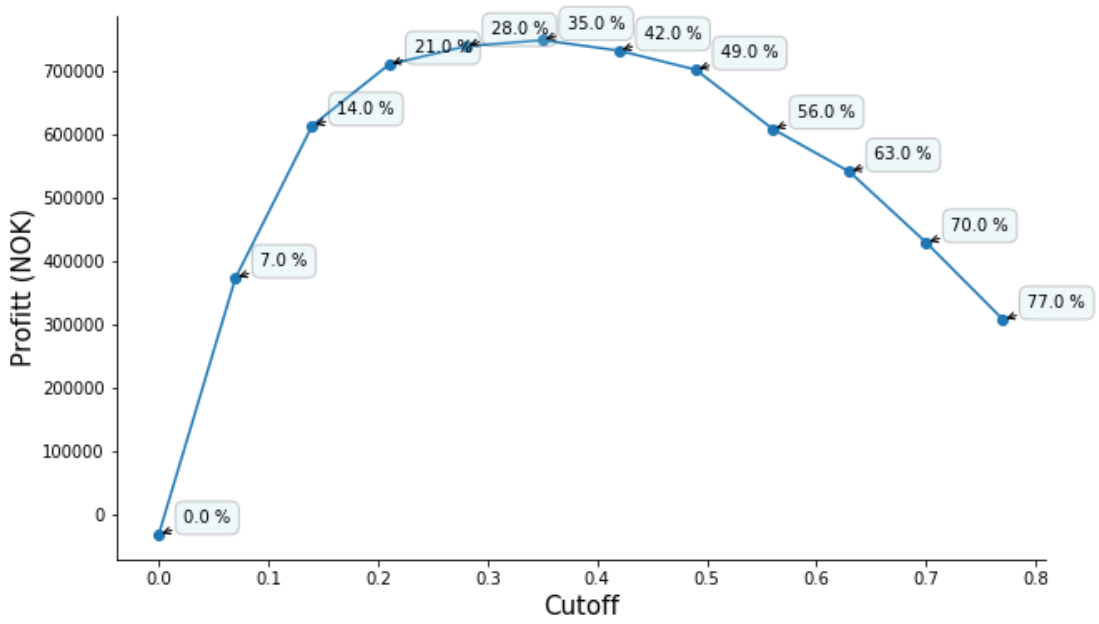
målvariabel. Variabler markert med blått skyver sannsynligheten nærmere 0, i motsetning til variabler markert med rødt, som følgelig skyver sannsynligheten mot 1. Variablene tilknyttet kunden vist i figur 13 skyver vedkommende lavere enn base value, ned til output value lik 0.39, hvilket betyr at kunde 104 er predikert å ha 39 % sannsynlighet for å velge nytt selskap.

Figur 13: Målvariabel «Nytt Selskap». Individuelt SHAP-plot for kunde 104. Variabler farget i blått påvirker sannsynligheten nærmere 0. Variabler farget i rødt påvirker sannsynligheten nærmere 1. Predikert sannsynlighet for nytt selskap for den aktuelle kunden er 39 %.

6.3 Datasett 3 – Målvariabel: Reddet

I det tredje og siste datasettet, der målvariabel er om en kunde kan reddes eller ikke, vil utfallet i større grad måles i estimert profitt. Dette begrunnes med at det er siste ledd i prosessen og at utfallet av den grunn er endelig. I metodekapittelet ble det introdusert en kostnadsmatrise, altså det å tilegne summer til de ulike verdiene av en confusion matrix. Optimal cutoff er derfor satt til den verdien som maksimerer den økonomisk estimerte profitten, og skiller seg fra de tidligere datasettene der cutoff er satt til å optimere verdien for F1-score.

Testsettet består av 4201 observasjoner og 28 variabler (inkludert målvariabel). Fordelingen av målvariabel er 70/30 for henholdsvis klasse 0 og klasse 1.



Figur 14: Profitt ved ulike cutoff-verdier for målvariabel «Reddet». Høyest verdi er realisert ved cutoff 0.35.

Resultatene fra kostnadsmatrisen gitt modellens prediksjoner er visualisert i figur 14 og viser til en maksimert profitt der cutoff for modellen er lik 0.35. Dette målet er følgelig benyttet for klassifiseringsrapport og confusion matrix vist i tabell 14 og 15.

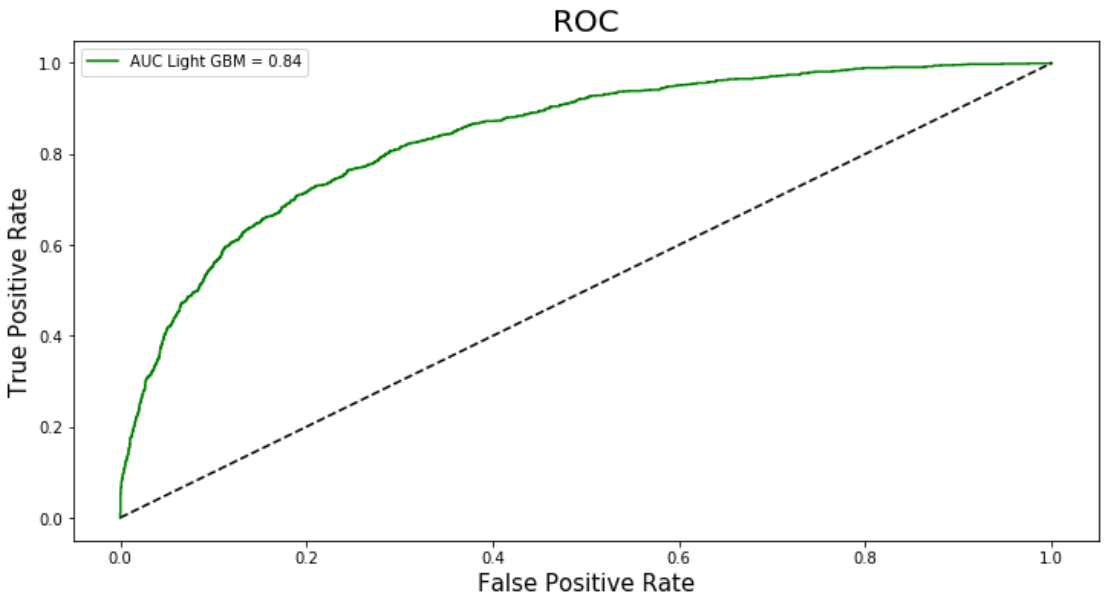
Klassifisering	*Precision*	*Recall*	*F1-score*	*Accuracy*
0	0.87	0.80	0.84	0.78
1	0.59	0.71	0.65	

Tabell 14: Klassifiseringsrapport for målvariabel «Reddet». Cutoff benyttet er 0.35.

		PREDIKERT	
		0	1
REELL	0	2399	597
	1	344	861

Tabell 15: Confusion matrix for målvariabel «Reddet». Cutoff benyttet er 0.35.

Tilhørende AUC for gjeldende prediksjon er målt til 0.84, identisk prediksjon 1. Med dette ser man AUC-verdier som varierer svært lite i forhold til hverandre for de ulike målvariablene. Til motsetning viser klassifiseringsrapportene, en oversikt følsom for hvilken cutoff satt, svært ulike resultater mellom prediksjonene. Forskjellene i resultatene fra klassifiseringsrapportene er sterkt påvirket av ulik cutoff og følgelig kan det være vanskelig å sammenligne prediksjonene. AUC er uavhengig av cutoff og er derfor mer sammenlignbare på tvers av datasettene.



Figur 15: ROC-kurve for målvariabel «Reddet».

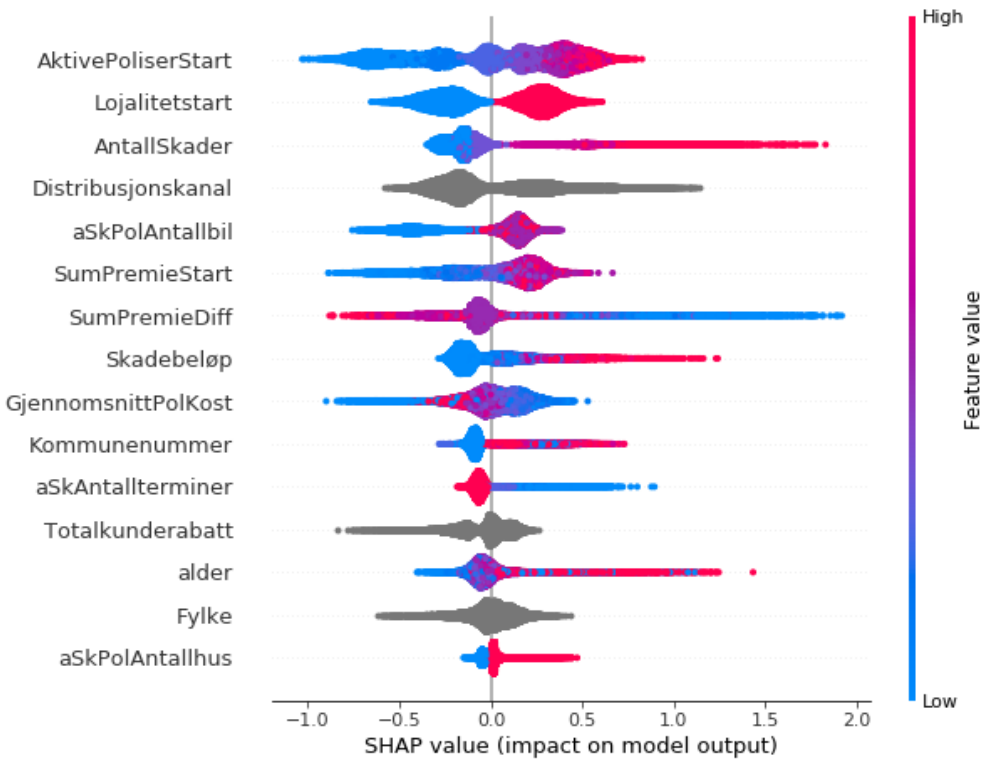
Lavere cutoff går ut over modellens evne til å predikere presist. Recall er her høyere enn precision og underbygger tidligere nevnt scenario om at høyere score innad i enkelte mål ikke kommer uten konsekvenser for andre mål. Modellen klassifiserer ganske enkelt flere tilfeller i klasse 1 fordi den belønnes mer for korrekte predikeringer enn den straffes for feil. I sum vil likevel den totale treffsikkerheten fremdeles være relativt høy, på 78 %, veldig likt verdiene rapportert fra de to foregående datasettene. Fra tidligere utregninger kan profitt per redningsforsøk med og uten modell sammenlignes. Per dags dato tjener Frende i snitt 15 NOK per redningsforsøk, mens man ved hjelp av et veiledet utvalg fra modell i snitt kan tjene 513 NOK per forsøk.

Modellen gjenkjenner ikke alle tilfellene hvor Frende har klart å gjennomføre redning, men kan belage seg på markant høyere redningsprosent. Til sammenligning er redningsprosenten til Frende på om lag 30 %, mens modellen har 59 %, og til tross for færre redninger øker likevel den totale forretningsverdien[2].

[2] Utregningene er basert på tall fra testsettet.

$$Total\ verdi\ for\ alle\ redningsforsøk\ uten\ modell = 15\ kr * 4201 = 63\ 015\ kr$$

$$Total\ verdi\ for\ alle\ redningsforsøk\ med\ modell = 513\ kr * 1458 = 747\ 954\ kr$$



Figur 16: SHAP-plot for målvariabel «Reddet». Variablenes viktighet er presentert i synkende rekkefølge. Blå farge representerer lav verdi innad i variabel. Rød farge representerer høy verdi innad i variabel. Skillelinjen i midten markerer hvorvidt observasjon påvirker utfallet mot klasse 0 eller 1. Observasjoner til venstre påvirker mot klasse 0 og observasjoner til høyre påvirker mot klasse 1.

Til tross for at det er blitt argumentert for at forretningsverdien i stor grad kan måles ved hjelp av økonomiske utregninger, vil det likevel være interessant å danne grupperinger. Fra SHAP-plottet skapes det igjen utsatte kundegrupper. Disse bærer noe preg av all filtreringen som er gjort for å konstruere datasett 3 og er derfor redusert til å gjelde relativt få kunder. Av den grunn har er det blitt foretatt færre og mindre ekstreme restriksjoner for å unngå å lage for små grupperinger. Dette gjelder spesielt de med høy redningsprosent, da ekstremverdier innenfor enkelte variabler fjernet så godt som samtlige observasjoner.

Gruppering	Kjennetegn	N	Klasse 0	Klasse 1	Andel Reddet
Gruppe 1	AktivePoliserStart > 3 Lojalitetstart = 1 SumPremieDiff < 50 AntallSkader > 1	219	74	145	66 %
Gruppe 2	AktivePoliserStart = 1 Lojalitetstart = 0 Antall bilforsikringer < 2	544	493	51	9 %

Tabell 16: Kundegrupperinger med kjennetegn og oversikt over «Andel Reddet».

I tillegg er det interessant å se nærmere på de kategoriske variablene distribusjonskanal, kunderabatt og fylke. For distribusjonskanal kommer det frem at «Eierbank» og «Frendes egne kanaler» står for kundemassene med lavest redningsprosent. «Franchise» og «Partner» er representert av relativt få kunder, og tallene kan av den grunn være mer problematiske å trekke slutninger fra. Fra kunderabatt er funnet at redningsprosenten øker i takt med hvilken rabattkode man har. Høyere rabattkode fører til økt sannsynlighet for å bli reddet. Fylkene har en relativt jevn fordeling, med unntak av Oslo og Vestlandet, som har henholdsvis høyere og lavere redningsprosent enn de andre. Vestlandet har betydelig flere kunder, flest av alle, mens Oslo er et av de med færrest. Dersom en kunde blir reddet i Oslo, vil dette ha mer å si for redningsprosenten, og det blir vanskeligere å sammenligne de to.

7 Diskusjon

7.1 Svar på forskningsspørsmål

Innledningsvis ble spørsmålet stilt om hvordan maskinlæringsmodeller kan bidra til kundebevaring. Rent konkret koker dette ned til bedre allokering av ressurser for forsikringsselskapet.

Uten benyttelse av slike modeller vil selskapet ha en mer reaktiv innstilling til kundebevaring. Det vil si at en kunde forteller at vedkommende har lyst til å avslutte sitt medlemskap, og eventuelle tiltak blir så iverksatt som en reaksjon på opplysningen. Maskinlæringsmodellene gjør derimot at man kan komme i forkant av problemet ved at man sitter med en oversikt over kunder som potensielt vil avgå, hvorfor og om de kan reddes. Tilnærmingen blir med det mer proaktiv. I tillegg er det presentert en oversikt over hvilke karakteristikker som kjennetegner grupperinger innad i klasse 1 og 0 for samtlige målvariabler.

Slik informasjon gjør det mulig å vite hvem man skal fokusere på og hvem det kan være lurt å la være å fokusere på. De ulike grupperingene gir en oversikt over hvor de generelle tiltakene bør fattes. Ved å se på kombinasjoner av karakteristikker finner man sammenhenger det for mennesker kan være vanskelig å oppfatte. I tillegg styrker modellene antagelser det trolig allerede var kjennskap til, som at lengde på medlemskap har stor påvirkning på om kunde avgår og at gode rabattordninger holder på kunder. Det er ikke dermed sagt at rabatt på poliser vil være den ultimate løsningen, da dette kan komme til å koste selskapet dyrt. I delkapittel 2.7 ble det argumentert for at lang varighet på medlemskap er gunstig for bedriften. Selv om argumentet stadig er gjeldende, kan det likevel oppstå tilfeller der bedriften er bedre tjent med å avslutte kundeforholdet, eksempelvis dersom en kunde har meget høy skadefrekvens over tid. Det overordnede målet må være å beholde kunder gitt at de stadig kan anses som inntektskilder.

På den andre siden kan individuelle tiltak innføres for enkeltpersoner. For å maksimere utbytte av en slik løsning vil trolig det mest logiske være å se på kunder med predikert ekstremverdier innenfor samtlige målvariabler. Det vil si at modellen har predikert høy sannsynlighet for avgang, høy sannsynlighet for at avgangen foregår til nytt selskap og høy sannsynlighet for

redning. En slik tilnærming begrunnes med at individuell oppfølging krever store mengder ressurser og med det utgifter. I oppgaven har det blitt estimert at et redningsforsøk koster omtrent 500 NOK. For at en individuell oppfølging skal være økonomisk gunstig, er det derfor viktig at det tas en tilnærming mot de mest utsatte individene og jobbe seg gradvis nedover ettersom predikert sannsynlighet minker.

Et sentralt problem som dukker opp ved å ta stilling til problemer før de forekommer, er hvorvidt dette faktisk ville vært et problem i utgangspunktet. Endelig modell i oppgaven har som vist en god del feilpredikeringer, til tross for at flertallet er korrekt, noe som kan føre til at man benytter verdifull tid og ressurser på å redde kunder det i utgangspunktet ikke ville vært nødvendig å redde. Slike tall vil dog være svært vanskelig å måle, og om det faktisk viser seg å være nyttig, får man ikke endelig svar på før det er testet. Tiltakene må overvåkes nøye, og kostnad i forhold til nytte vurderes. I tillegg er det hensiktsmessig å rette et kritisk blikk mot resultatene fra modellen.

7.2 Kommentar til endelig modell og implementering

I dette delkapittelet vil det bli kommentert vurderinger gjort ved utvikling av modell, samt problemstillinger som kan oppstå dersom nevnte modell blir implementert i praksis.

I de to første stegene av prediksjonsprosessen har det hovedsakelig blitt fokusert på å optimere F1-score. Argumentet var at det var hensiktsmessig å kombinere god precision og recall for å kunne identifisere flest mulig tilfeller med best mulig presisjon. En slik tilnærming fører riktignok til at mye data blir droppet mellom stegene. Fra datasett benyttet i prediksjon 1 var om lag 180 000 observasjoner tilgjengelig. Ved siste steg var antallet redusert til 28 000. En alternativ løsning kunne vært å fokusere på optimal recall for klasse 1. Med dette vil flere tilfeller bli plassert i nevnte klasse, og da kravet for å overføres til neste steg er å havne her, vil vi ende opp med mer data der avgjørelsen om kunde kan reddes blir tatt. Negativt er at scenariet bringer med seg flere feilpredikeringer, og relaterte kostnader vurderes til å være større enn potensielt utbytte av økt dataoverføring.

Hvor stor konsekvens slike kostnader får er sterkt tilknyttet vurderingen av kundens livsløpsverdi. Dersom det skulle vise seg at estimatene presentert i oppgaven er feil, og deres

livsløp i realiteten er lenger, vil en modell som ønsker å maksimere profitt fokusere ytterligere på høyere recall, da slike predikeringer blir belønnet enda mer. Ved motsatt tilfelle, dersom videre livsløp er overvurdert, vil økt fokus på precision være å foretrekke. Dette fordi modellen i et slikt tilfelle ikke lenger belønnes like mye, hvilket vil si at en riktig prediksjon ikke er like mye verdt. Som et biprodukt vil også dataoverføringen henholdsvis øke eller minke i de to scenariene.

Det vises i tillegg til SHAP, som kan benyttes for individuelle prediksjoner, slik at bedrift, og dermed ansatte, får mulighet til å vurdere om prediksjonen er fornuftig. Samspill mellom ansatte og modell blir dermed skapt, og i en startfase anses det som riktig å ha en slik tilnærming. Involvering gjør at utfallene kontinuerlig blir overvåket, og feil blir trolig oppdaget raskere, som igjen fører til at endringer kan implementeres. Det bringer dog med seg stor grad av skjønn og med det øker usikkerhetsmomentene.

Tidsaspektet er et annet tema vurdert i oppgaven. Tidligere ble det argumentert for at det er mindre viktig, da majoriteten av variablene er hentet et halvt år før målvariabel er predikert, og det gjenkjennes likevel betydelig flere tilfeller korrekt enn galt. For dette har vi ikke noe sammenligningsgrunnlag, og det er mulig at uthenting av data nærmere prediksjonsdato kan være å foretrekke. Det blir dog i stor grad spekulering og vil i tillegg gå utover reaksjonstiden fra en kunde blir flagget som utsatt til tiltak iverksettes.

Videre kan et problem oppstå ved registrering av nye kunder. Kunder med medlemstid under 12 måneder har blitt utelatt fra treningen, og modellen har derfor ingen informasjon om deres handlingsmønstre. Det har blitt vurdert slik at oppførselen for nye kunder er likest de med 12 til 24 måneders medlemskap, og ved nye tilfeller vil disse bli gruppert sammen. Dersom det viser seg at deres handlingsmønstre er vesentlig annerledes, vil andelen feilpredikeringer for nyankomne forsikringstakere trolig være høy.

Til sist er det ønskelig å kommentere det målrettede fokuset mot sterk generaliseringsevne for modellene. Modeller som lærer for mye av treningssettet, vil kunne fremstå som gode, men vil i praksis være ubrukelige på ny data. Positivt er derfor at resultatene fra trenings-, validerings- og testsett viser seg relativt like. Fra oversikten over SHAP vurderes det i tillegg at modellen er

trent på riktig grunnlag. Jevnt fordelte SHAP-verdier for variablene forteller samtidig at fullstendig informasjon ikke er til stede.

7.3 Videre forskning

I oppgaven har det blitt sett på hvordan utgangsverdiene til maskinlæringsmodeller kan forstås ved hjelp av SHAP-verdier. Forståelsen gir mennesker grunnlag for å samhandle i beslutningsprosessen, men det har ikke blitt tatt for seg hvordan dette kan gjøres. En videre problemstilling kan derfor være:

- *Hvordan optimalisere samspillet mellom menneske og maskinlæringsmodeller for best mulig kundebevaring?*

Det er ikke dermed sagt at forskning på kundebevaring er ferdig, noe resultatene fra oppgaven er et eksempel på, da de er langt fra perfekte. Som en forlengning av arbeidet vil det være mulig å jobbe videre med hyperparameteroptimering for modellene. Det har ikke blitt rettet stort fokus mot slik optimering i oppgaven, og det gjenstår å se om man kan presse modellene til bedre resultat dersom det legges ned mer tid på dette. Det kan også være interessant å se på om introduksjon av nye variabler kan styrke modellene. Samtidig kan det være nyttig å forsøke nye modeller, spesielt nevrale nettverk. De forkastede modellene i oppgaven viste til dårligere resultater enn gradient boosting algoritmene, og videre fokus på disse er ikke å anbefale. Nevrale nettverk har typisk sterk evne til å gjenkjenne sammenhenger i kompliserte datasett, men vil mulig kreve enda mer data for å nå sitt fulle potensiale. Dersom Frende forsetter sin kundevekst, kan det skje naturlig, eller via samarbeid med andre selskap.

8 Konklusjon

Formålet med oppgaven har vært å belyse hvordan maskinlæring kan redusere kundeavgang i et forsikringsselskap. Problemet ble delt inn i tre steg, henholdsvis avgang, årsak til avgang og om de kan reddes. Modellene har gjort det mulig å identifisere utsatte kundegrupper, slik at firmaets ressurser kan bli brukt på en mer målrettet måte for å forhindre avgang. På denne måten kan det i større grad oppnås en proaktiv tilnærming til kundebevaring.

Dersom de generelle tiltakene ikke fungerer, og kundene velger å avgå, vil det likevel være mulig å foreta et redningsforsøk. Modellen viser til sterke resultater hva gjelder presisjon og gjenkjenning av kunder det er mulig å redde. Økt presisjon for redningsforsøkene fører til høyere realisert verdi for bedriften. I dag ligger gjennomsnittlig gevinst per redningsforsøk på 15 NOK. Til sammenligning kan bruk av modell øke denne verdien til 513 NOK.

Videre kan modellen vise til lignende resultater på trenings-, validerings- og testsettet, og det kan konkluderes med at en sterk generaliseringsevne har blitt oppnådd.

Beslutningsprosessen til modellene kan i tillegg tolkes for individuelle observasjoner, og med det gi bedriften mulighet til å involvere ansatte i endelige vurderinger.

9 Referanseliste

Al Daoud, E. (2019). Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. *International Journal of Computer and Information Engineering, 13*(1), 6-10.

Applied Innovation. (2017). Husky or Wolf? Using a Black Box Learning Model to Avoid Adoption Errors. Retrieved from http://innovation.uci.edu/2017/08/husky-or-wolf-using-a-black-box-learning-model-to-avoid-adoption-errors/

Autor, D. (2014). *Polanyi's paradox and the shape of employment growth* (Vol. 20485): National Bureau of Economic Research Cambridge, MA.

Bathaee, Y. (2017). The artificial intelligence black box and the failure of intent and causation. *Harv. JL & Tech., 31*, 889.

CatBoost. (2020). Benchmark. Retrieved from https://catboost.ai/#benchmark

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system.* Paper presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.

Chylinski, M. (2016). *Retentionomics: The Path to Profitable Growth*. Retrieved from Forbes Insights:

de Oliveira, J. G. (2019). A study on Gradient Boosting algorithms.

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology: Theory and Applications* (pp. 3-11). Cham: Springer International Publishing.

Farzanfar, E., & Delafrooz, N. (2016). Determining the Customer Lifetime Value based on the Benefit Clustering in the Insurance Industry. *Indian Journal of Science and Technology, 9*. doi:10.17485/ijst/2016/v9i1/72307

Forsikringsavtaleloven. (2006). Lov om forsikringsavtaler (LOV-1989-06-16-69). Retrieved from https://lovdata.no/dokument/NL/lov/1989-06-16-69

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Gjensidige Forsikring ASA. (2019). *Årsrapport 2019*. Retrieved from

Günther, C.-C., Tvete, I. F., Aas, K., Sandnes, G. I., & Borgan, Ø. (2014). Modelling and predicting customer churn from an insurance company. *Scandinavian Actuarial Journal, 2014*(1), 58-71.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*: Springer.

Hayes, A. (2019). Combined Ratio Definition. Retrieved from https://www.investopedia.com/terms/c/combinedratio.asp

Insr Insurance Group ASA. (2019). Årsrapport 2019.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T.-Y. (2017). *Lightgbm: A highly efficient gradient boosting decision tree.* Paper presented at the Advances in neural information processing systems.

Keitakurita. (2018). LightGBM and XGBoost Explained. Retrieved from https://mlexplained.com/2018/01/05/lightgbm-and-xgboost-explained/

Lundberg, S., & Lee, S.-I. (2016). An unexpected unity among methods for interpreting model predictions. *arXiv preprint arXiv:1611.07478*.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence, 267*, 1-38.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). *CatBoost: unbiased boosting with categorical features.* Paper presented at the Advances in neural information processing systems.

Punj, G., & Stewart, D. W. (1983). Cluster Analysis in Marketing Research: Review and Suggestions for Application. *Journal of Marketing Research, 20*(2), 134-148. doi:10.2307/3151680

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development, 3*(3), 210-229.

Topdanmark Forsikring A/S. (2019). Annual Report 2019.

Torkzadeh, G., Chang, J. C.-J., & Hansen, G. W. (2006). Identifying issues in customer relationship management at Merck-Medco. *Decision Support Systems, 42*(2), 1116-1130.

Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory, 55*, 1-9.

10 Figurliste

11 Tabelliste

12 Vedlegg

12.1 Formler

12.1.1 Gradient boosting algoritme for klassifikasjonsproblem

1. $Initialize\ f_{k0}(x) = 0, k = 1,2, \dots, K\ \ (11)$

2. $For\ m = 1\ to\ M$:

$(a)\ Set$

$$p_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^{K} e^{f_l(x)}}, k = 1,2, \dots, K$$

$(b) For\ k = 1\ to\ K$:

$i. Compute\ r_{ikm} = y_{ik} - p_k(x_i), i = 1,2, \dots, N$

$ii. Fit\ a\ regression\ tree\ to\ the\ targets\ r_{ikm}, i = 1,2, \dots, N$

$giving\ terminal\ regions\ R_{jkm}, j = 1,2, \dots, J_m$

$iii. Compute$:

$$\gamma_{jkm} = \frac{K-1}{K} \frac{\sum_{x_i \epsilon R_{jkm}} r_{ikm}}{\sum_{x_i \epsilon R_{jkm}} |r_{ikm}|(1 - |r_{ikm}|)}, j = 1,2, \dots, J_m$$

$$iv. Update\ f_{km}(x) = f_{k,m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} I\left(x \in R_{jkm}\right)$$

3. $Output\ \hat{f}_k(x) = f_{kM}(x), k = 1,2, \dots, K$

Ovenfor er algoritmen til Gradient Boosting for et klassifiseringsproblem vist. I første steg settes en startprediksjon for alle observasjonene, som algoritmen deretter går videre fra. Innledningsvis settes verdien til 0 og derfor vil alle observasjoner bli predikert til klasse 0 i første steg.

Steg to er der hvor trærne vil bygges, hvor «M» er antall trær og «m» er hvert enkelt tre. Stegene under punkt 2. vil bli gjort «M» ganger. I (a) blir modellens faktiske prediksjon utredet. Videre i steg (b) i. blir pseudo residualene utregnet. Dette gjøres ved å ta målvariabelen sin faktiske verdi minus predikert verdi, hvilket blir gjort for alle observasjoner.
I steg ii. vil et regresjonstre bli bygd fra variablene i datasettet ved hjelp av pseudoresidualene

utregnet i forrige steg.

I steg iii. blir en Gamma-verdi som tilnærmet minimerer tapsfunksjonen utregnet for hvert enkelt blad. Grunnen til at løsningen bare er hva som tilnærmet minimerer tapsfunksjonen er at formelen dette steget kommer fra ikke har en endelig løsning, og derfor blir løsningen bare tilnærmet korrekt.

I steg iv. blir prediksjonene oppdatert i henhold til hva som er predikert tidligere. Bladene oppdateres ved å benytte den forrige verdien og legge til Gamma-verdien utregnet i steg iii.

Når steg to er gjennomført «M» ganger går algoritmen over til steg tre. Siste verdi utregnet i steg 2(b)iv. vil følge over til dette steget og blir med det endelig utgangsverdi.

12.1.2 Shapley-verdier og SHAP

Shapley-verdier:

$$\emptyset_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}\left(x_{S \cup \{i\}}\right) - f_S(x_S) \right] (12)$$

SHAP:

$$g(z') = \emptyset_0 + \sum_{i=1}^{M} \emptyset_i Z'_i = bias + \sum bidrag\ fra\ variabel\ (13)$$

12.1.3 Statistiske mål

For klasse 1:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}\ (14)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}\ (15)$$

For klasse 0:

$$Precision = \frac{True\ Negative}{True\ Negative + False\ Negative}\ (16)$$

$$Recall = \frac{True\ Negative}{True\ Negative + False\ Positive}\ (17)$$

Uavhengig av klasse:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}\ (18)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}\ (19)$$

12.2 Databehandling

Slik nevnt i innledningen blir det foretatt tre ulike prediksjoner relatert til datasettet. Dette krever at i tillegg til å lage modeller med individuelt optimerte parametere, må det konstrueres ulike datasett. Fra det opprinnelige datasettet er antall observasjoner (rader) og variabler (kolonner) blitt justert basert på vurderinger fra analyse av dataen.

Felles for de tre datasettene er at inngangsvariablene vil være like etter databehandling er gjennomført. Dette kan gjøres fordi informasjonen tilknyttet hver enkelt variabel er tilgjengelig før hver prediksjon skal gjennomføres. Med andre ord er det ingen variabler som skaper fullstendig informasjon ved å bli inkludert. Forskjellene oppstår når konstruksjon av målvariabel blir gjennomført, hvilket vil bli kommentert videre i eget delkapittelet. Endringene i datasettet kan oppsummeres som *endring på eksisterende variabler*, *konstruksjon av nye variabler*, *fylle inn for manglende verdier, fjerning av opprinnelige variabler* og *konstruksjon av målvariabler*.

12.2.1 Endring på eksisterende variabler

Den første endringen er gjort etter undersøkelse av variablene *kundeførstegang* og *totaltmnd*. De to viser henholdsvis til når en kunde er blitt registrert og hvor mange måneder vedkommende har vært hos Frende. Som følge av metoden Frende har benyttet for å hente ut datasettet, består opprinnelig versjon utelukkende av avgåtte forsikringstakere registrert som kunde første gang i 2019. Med andre ord er det ikke tatt med kunder registrert i 2019 som fortsatt er medlem. Videre er det derfor ingen registrerte kunder med medlemstid under 12 måneder som *ikke* er avgått. Dette skaper en svært skjev fordeling og vil trolig føre til at maskinlæringsmodellene vil lære å se sammenheng mellom kort medlemstid og avgang fra bedriften. Ved fremtidige predikeringer kan derfor nye kunder automatisk bli kategorisert som avgått, hvilket er et uønsket scenario. For å takle den ujevne fordelingen er det blitt valgt å droppe samtlige kunder registrert i 2019 og medlemstid under 12 måneder. I tillegg er det blitt foretatt en gruppering av *totaltmnd*-variabelen, fordelingen kan sees fra tabell 17. Grupperingen skjer for å gi modellen en metode å kategorisere nye kunder på og følger en antagelse om at handlingsmønstre for kunder med lav medlemstid er relativt lik. I tillegg vil det være gunstig for generaliseringsevnen til modellen.

Variabel: totaltmnd	
Gruppe	*Antall måneder i bedrift*
0	Mindre enn 25 måneder
1	25 til 48 måneder
2	49 til 72 måneder
3	73 til 96 måneder
4	97 til 120 måneder
5	Over 120 måneder

Tabell 17: Lengde på medlemskap som inngår i de ulike gruppene for variabelen totaltmnd.

12.2.2 Konstruksjon av nye variabler

Videre er det blitt konstruert en ny binær variabel relatert til om vedkommende er blitt reddet før eller ikke, ved å se på verdier tilknyttet den opprinnelige variabelen *AntallRedninger*. Med denne variabelen kommer det frem dersom en kunde er blitt reddet og i så fall hvor mange ganger dette har forekommet. Dersom en kunde er blitt reddet 2 eller flere ganger vil vedkommende få status som «reddet tidligere» og følgelig bli tildelt verdi 1 i ny binær variabel, kalt *reddetFør*, ellers 0. Grunnen til at verdier med 2 eller flere blir sett på, dreier seg igjen om å unngå og gi modellen fullstendig informasjon. Dersom *AntallRedninger* viser at de kun er reddet én gang kan det vise til nåværende tilfellet og dermed føre til at modellene trenes på fullstendig informasjon, hvilket igjen skaper bias og lite robuste modeller. Av kunder som er reddet tidligere er det omtrent 50 % som forlater bedriften, dette i motsetning til omtrent 30 % blant resterende.

I tillegg er det blitt sett på ulikheter mellom variabler som måler forskjeller i henholdsvis premiepenger og antall livsforsikringer, først målt for 6 måneder siden, deretter for 1 måned siden. Den nye variabelen *SumPremieDiff* er laget fra forskjellen mellom *SumPremie1mnd* og *SumPremieStart*, mens *AntallLivDiff* er laget fra forskjellen mellom *AntallLiv1mnd* og *AntallLiv*. Til slutt er det blitt laget en variabel relatert til gjennomsnittlig polisekostnad basert på *SumPremieStart* og *AktivePoliserStart*.

Variabelnavn	*Type*
ReddetFør	Binær
SumPremieDiff	Numerisk
AntallLivDiff	Numerisk
GjennomsnittPolKost	Numerisk

Tabell 18: Nye variabler konstruert fra eksisterende variabler i opprinnelig datasett.

12.2.3 Manglende verdier

Opprinnelig datasett inneholder i tillegg en del manglende verdier. I enkelte tilfeller kan det føre til at store mengder data faller bort dersom man ikke har en fornuftig metode for å innsette nye verdier. Både fordi tilfeldig påfyll av nye verdier vil føre til at modellene trenes på feil grunnlag, og fordi de simpelthen ikke lar seg trene, som følge av restriksjoner innad i programmeringsspråk, dersom manglende verdier er til stede. Manglede verdier ble ikke notert ned under opprinnelig konstruksjon av datasettet, men vist som blank eller «nan». Eksempelvis dersom en kunde ikke har bilforsikring vil dette vises uten verdi. Løsningen ble derfor ganske enkelt å erstatte de manglede feltene med verdien 0.

12.2.4 Droppede variabler

Valget har også falt på å droppe en hel del variabler. Dette gjøres nok en gang for å unngå fullstendig informasjon, eksempelvis ved å droppe *DatoSK_KundeAvgangAktivert*, som forteller om når kunden hadde avgang, informasjon som åpenbart ikke skal være til stede under prediksjon av kundeavgang. Det er også flere overflødige variabler som har identiske verdier, eksempelvis *AktivePoliser1mnd* og *AktivePoliserStart*, der førstnevnte er blitt droppet. Resterende variabler som er blitt droppet av samme grunn vil ikke kommenteres videre, en fullstendig liste kan sees i tabell 19.

Det er likevel én droppet variabel det vil bli kommentert videre på, nemlig *kundeførstegang*. Som navnet tilsier forteller denne om når kunden første gang ble registrert hos Frende. Det kan være fare for at modellen vil lære å gjenkjenne sammenhenger mellom *kundeførstegang* og *totaltmnd*, altså lengde på medlemskap. I seg selv er ikke variabelen et problem da modellen ikke kjenner til tidspunktet for dagen den foretar prediksjonen. Kombinert med lengde på medlemskap er problemet at lik registreringsdato, men ulik lengde på medlemskap, gir garanti for at minst én har avgått fra selskapet. Variabelen er derfor droppet.

Variabler	*Type*
SumPremieNå	Numerisk
SumPremie1mnd	Numerisk
AktivePoliser1mnd	Numerisk
aSkPolAntallbil1mnd	Numerisk
aSkPolAntallhus1mnd	Numerisk
aSkPolAntallinnbo1mnd	Numerisk
aSkPolAntallreise1mnd	Numerisk
AntallLiv1mnd	Numerisk
AntallRedninger	Numerisk
MedIRedningsrapport	Kategorisk
Redningsunderkategori	Kategorisk
Kundeførstegang	Dato
DateSK_KundeAvgangAktivert	Dato
Redningsdato	Dato

Tabell 19: Variabler droppet fra opprinnelig datasett.

12.2.5 Konstruksjon av målvariabel

Til sist er målvariabler blitt konstruert. Fra figur 1 i innledningen ble det presentert en forenklet oversikt over hvordan disse henger sammen, men det vil nå bli forklart en mer detaljert beskrivelse om hvordan dette er blitt gjort.

Første del i prosessen er å predikere hvorvidt en kunde avgår eller ikke, der målvariabelen kalles *avgang*. Denne målvariabelen er ikke blitt endret fra opprinnelig datasett og den inneholder samtlige observasjoner, sett bort fra de fjernet av grunner listet ovenfor.

Andre del i prosessen dreier seg om å predikere hvorvidt en kunde avgår til nytt selskap eller ikke. For at kunder skal ha en verdi i denne kolonnen er de allerede nødt til å ha avgått, det vil si at samtlige kunder som ikke har avgått vil bli fjernet ved konstruksjon av datasettet til gjeldende prediksjon. I tillegg til «nytt selskap» er det listet opp en hel del andre begrunnelser for avgang, som «behov opphørt». Etter samtale med Frende er det dog blitt bestemt at alle som har nytt selskap som årsak vil få tildelt verdien 1, ellers 0. På denne måten vil målvariabelen igjen bli binær.

Siste steg omhandler hvorvidt kunder kan reddes eller ikke. Målvariabelen er konstruert fra kunder som er med i redningsrapport, en variabel inkludert i opprinnelig datasett. Av kundene som *ikke* blir reddet har om lag 75 % av disse årsak til avgang lik nytt selskap. Ideelt sett, for å skape en best mulig sammenheng mellom prediksjonene, ville alle hatt denne årsaken. For å skape et større datasett, og med det mer informasjon å trene på, er likevel de resterende 25 %

som ikke har dette som sin avgangsårsak blitt inkludert. Hva gjelder kundene som *er* reddet vil ikke disse ha forlatt selskapet, og det er dermed ingen kjennskap til årsaken de ville hatt for å avgå, da denne informasjonen først registreres etter avgang. Det er dog rimelig å anta at hovedandelen ville avgått til nytt selskap, da slike kunder er i fokusområdet til Frende, og er disse man i praksis vil ha mulighet til å redde.

Med dette er alle endringer gjort, og nytt datasett er klart til bruk. Det består av 27 inngangsvariabler, en oversikt kan sees fra tabell 20.

Variabler	***Min***	***Maks***	***Gjennomsnitt***	***Antall kategorier***	***Type***
ASkAntallterminer	1	12	8.18	-	Numerisk
AntallPerioder	1	10	1.14	-	Numerisk
alder	18	107	50.08	-	Numerisk
SumPremieStart	29	151632	10132.92	-	Numerisk
AktivePoliserStart	1	33	3.28	-	Numerisk
aSkPolAntallbil	0	8	0.78	-	Numerisk
aSkPolAntallhus	0	10	0.47	-	Numerisk
aSkPolAntallinnbo	0	8	0.61	-	Numerisk
aSkPolAntallreise	0	3	0.59	-	Numerisk
AntallLiv	0	10	0.52	-	Numerisk
AntallSkader	0	20	0.74	-	Numerisk
Skadebeløp	0	13600003	18820.65	-	Numerisk
AntallSkader6mnd	0	10	0.15	-	Numerisk
Skadebeløp6mnd	0	10037900	4000.47	-	Numerisk
totaltmnd	0	5	1.98	-	Numerisk
SumPremieDiff	-20268	37926	176.20	-	Numerisk
AntallLivDiff	-4	6	0.01	-	Numerisk
GjennomsnittPolKost	29	37728	3211.65	-	Numerisk
Kjønn	-	-	-	2	Kategorisk
Distribusjonskanal	-	-	-	4	Kategorisk
Selvbetjent	-	-	-	2	Kategorisk
aSkBetalingsmåte	-	-	-	2	Kategorisk
Fylke	-	-	-	12	Kategorisk
Kommunenummer	-	-	-	16	Kategorisk
Totalkunderabatt	-	-	-	3	Kategorisk
Lojalitetstart	-	-	-	2	Kategorisk
ReddetFør	-	-	-	2	Kategorisk

Tabell 20: Fullstendig oversikt over inngangsvariablene for ferdig datasett. For numeriske variabler vises min, maks og gjennomsnitt. For kategoriske variabler vises antallet kategorier innad i variabelen.

12.3 Resultater

12.3.1 Resultater fra samtlige modeller

Prediksjon 1 – Målvariabel «Avgang»

Kandidatmodeller						
Modell	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
XGBoost	Klasse 0	0.82	0.89	0.85	0.78	0.83
	Klasse 1	0.70	0.53	0.60		
LightGBM	Klasse 0	0.82	0.91	0.86	0.79	0.84
	Klasse 1	0.71	0.53	0.61		
CatBoost	Klasse 0	0.80	0.92	0.85	0.78	0.82
	Klasse 1	0.71	0.46	0.56		
Forkastede modeller						
Modell	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
Logistic Regression	Klasse 0	0.81	0.67	0.73	0.69	0.76
	Klasse 1	0.49	0.72	0.58		
Random Forest	Klasse 0	0.84	0.69	0.75	0.70	0.77
	Klasse 1	0.51	0.71	0.59		
Decision Trees	Klasse 0	0.84	0.65	0.73	0.67	0.76
	Klasse 1	0.48	0.72	0.58		
KNN	Klasse 0	0.82	0.66	0.73	0.66	0.72
	Klasse 1	0.46	0.68	0.55		

Tabell 21: Resultater fra samtlige modeller for prediksjon 1 – målvariabel «Avgang».

Prediksjon 2 – Målvariabel «Nytt Selskap»

Kandidatmodeller						
Modell	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
XGBoost	Klasse 0	0.80	0.71	0.75	0.76	0.85
	Klasse 1	0.74	0.80	0.77		
LightGBM	Klasse 0	0.80	0.72	0.76	0.77	0.86
	Klasse 1	0.74	0.82	0.78		
CatBoost	Klasse 0	0.75	0.80	0.77	0.76	0.85
	Klasse 1	0.77	0.74	0.75		
Forkastede modeller						
Modell	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
Logistic Regression	Klasse 0	0.75	0.62	0.68	0.71	0.80
	Klasse 1	0.70	0.80	0.75		
Random Forest	Klasse 0	0.78	0.62	0.69	0.72	0.82
	Klasse 1	0.69	0.83	0.75		
Decision Trees	Klasse 0	0.77	0.67	0.71	0.74	0.81
	Klasse 1	0.72	0.80	0.76		
KNN	Klasse 0	0.71	0.76	0.73	0.71	0.79
	Klasse 1	0.75	0.70	0.73		

Tabell 22: Resultater fra samtlige modeller for prediksjon 2 – målvariabel «Nytt Selskap».

Prediksjon 3 – Målvariabel «Reddet»

Kandidatmodeller						
Modell	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
XGBoost	Klasse 0	0.85	0.80	0.82	0.77	0.83
	Klasse 1	0.59	0.69	0.64		
LightGBM	Klasse 0	0.87	0.80	0.84	0.78	0.84
	Klasse 1	0.59	0.71	0.65		
CatBoost	Klasse 0	0.83	0.89	0.86	0.79	0.82
	Klasse 1	0.66	0.54	0.60		
Forkastede modeller						
Modell	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
Logistic Regression	Klasse 0	0.87	0.69	0.77	0.71	0.80
	Klasse 1	0.51	0.76	0.61		
Random Forest	Klasse 0	0.88	0.66	0.75	0.70	0.79
	Klasse 1	0.49	0.79	0.61		
Decision Trees	Klasse 0	0.86	0.65	0.74	0.68	0.76
	Klasse 1	0.48	0.75	0.58		
KNN	Klasse 0	0.83	0.72	0.77	0.70	0.74
	Klasse 1	0.50	0.65	0.56		

Tabell 23: Resultater fra samtlige modeller for prediksjon 3 – målvariabel «Reddet».

12.3.2 Sammenligning av resultater fra trenings-, validerings- og testsett for LightGBM

LightGBM – målvariabel «Avgang»						
Datasett	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
Trening	Klasse 0	0.83	0.92	0.87	0.82	0.88
	Klasse 1	0.77	0.57	0.66		
Validering	Klasse 0	0.81	0.90	0.85	0.79	0.84
	Klasse 1	0.71	0.53	0.61		
Test	Klasse 0	0.82	0.91	0.86	0.79	0.84
	Klasse 1	0.71	0.53	0.61		

Tabell 24: Resultatene fra trenings-, validerings- og testsett for endelig modell, LightGBM. Målvariabel «Avgang».

LightGBM – målvariabel «Nytt Selskap»						
Datasett	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
Trening	Klasse 0	0.82	0.74	0.78	0.79	0.88
	Klasse 1	0.77	0.84	0.80		
Validering	Klasse 0	0.79	0.71	0.75	0.76	0.85
	Klasse 1	0.74	0.82	0.78		
Test	Klasse 0	0.80	0.72	0.76	0.77	0.86
	Klasse 1	0.74	0.82	0.78		

Tabell 25: Resultatene fra trenings-, validerings- og testsett for endelig modell, LightGBM. Målvariabel «Nytt Selskap»

LightGBM – målvariabel «Reddet»						
Datasett	**Klasse**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC**
Trening	Klasse 0	0.91	0.86	0.88	0.84	0.92
	Klasse 1	0.71	0.80	0.75		
Validering	Klasse 0	0.87	0.81	0.84	0.78	0.84
	Klasse 1	0.62	0.71	0.66		
Test	Klasse 0	0.87	0.80	0.84	0.78	0.84
	Klasse 1	0.59	0.71	0.65		

Tabell 26: Resultatene fra trenings-, validerings- og testsett for endelig modell, LightGBM. Målvariabel «Reddet».

12.4 Cluster-analyse for prediksjon 2 «Nytt Selskap»

Variabelnavn	Cluster 1 mean	Cluster 2 mean	Global mean	Type
Proba_1.0	**0.87**	**0.24**	**0.5**	**Sannsynlighet**
AktivePoliserStart	5.19	1.28	2.42	Numerisk
AntallLiv	0.74	0.23	0.43	Numerisk
AntallLivDiff	0.01	0.02	0.01	Numerisk
AntallPerioder	1.16	1.16	1.17	Numerisk
AntallSkader	0.85	0.24	0.44	Numerisk
AntallSkader6mnd	0.30	0.07	0.15	Numerisk
GjennomsnittPolKost	3474.20	1588.01	3478.49	Numerisk
Skadebeløp	23544.07	3625.86	11774.86	Numerisk
Skadebeløp6mnd	6515.85	1040.41	3919.21	Numerisk
SumPremieDiff	254.31	35.12	104.63	Numerisk
SumPremieStart	17026.00	2078.35	7891.63	Numerisk
aSkAntallterminer	10.11	3.33	7.77	Numerisk
aSkPolAntallbil	1.30	0.10	0.67	Numerisk
aSkPolAntallhus	0.89	0.01	0.32	Numerisk
aSkPolAntallinnbo	0.89	0.27	0.41	Numerisk
aSkPolAntallreise	0.74	0.75	0.48	Numerisk
alder	49.00	38.90	45.64	Numerisk
totaltmnd	1.29	0.45	1.17	Numerisk
Lojalitetstart	0.38	0.07	0.35	Binær
ReddetFør	0.06	0.00	0.03	Binær
Selvbetjent	0.01	0.40	0.12	Binær
Eierbank	0.57	0.50	0.57	Binær
Frendes egne kanaler	0.35	0.44	0.36	Binær
Partner	0.04	0.04	0.04	Binær
Franchise	0.04	0.02	0.02	Binær
Fylke=Vestland	0.25	0.33	0.30	Binær
Fylke=Viken	0.12	0.14	0.14	Binær
Fylke=Agder	0.14	0.10	0.12	Binær
Fylke=Nordland	0.15	0.04	0.9	Binær
Fylke=Oslo	0.03	0.15	0.08	Binær
Fylke=Rogaland	0.09	0.07	0.08	Binær
Fylke=Trøndelag	0.07	0.05	0.06	Binær
Fylke=Troms og Finnmark	0.05	0.04	0.04	Binær
Fylke=Vestfold og Telemark	0.04	0.04	0.04	Binær
Fylke=Innlandet	0.04	0.03	0.03	Binær
Fylke=Møre og Romsdal	0.02	0.02	0.02	Binær
Kjonn=Mann	0.79	0.54	0.68	Binær
Kjonn=Kvinne	0.21	0.46	0.32	Binær
Totalkunderabatt=A	0.10	0.91	0.66	Binær

Totalkunderabatt=B	0.52	0.07	0.22	Binær
Totalkunderabatt=C	0.37	0.01	0.12	Binær
aSkBetalingsmåte=Avtalegiro	0.75	0.21	0.54	Binær
aSkBetalingsmåte=Giro	0.25	0.79	0.46	Binær

Tabell 27: Gjennomsnittlige verdier for tilhørende variabler i cluster 1 og cluster 2. «Cluster 1 mean» og «cluster 2 mean» referer til gjennomsnittlig verdi for variabel innad i gjeldende gruppe. «Global mean» referer til gjennomsnittlig verdi for datasettet 2, målvariabel «Nytt Selskap». «Proba_1.0» viser sannsynligheten for nytt selskap ved avgang for de respektive gruppene.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway