Stockholms universitet

Premieskattning med hjälp av mer tolkningsbara trädbaserade metoder

Filip Axelsson

Masteruppsats i försäkringsmatematik
Master Thesis in Actuarial Mathematics

Masteruppsats 2023:15
Försäkringsmatematik
Juni 2023

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematisk statistik
Stockholms universitet
Masteruppsats **2023:15**
http://www.math.su.se/matstat

Premieskattning med hjälp av mer tolkningsbara trädbaserade metoder

Filip Axelsson*

Juni 2023

Sammanfattning

Under många år har samma metod använts för prissättning inom sakförsäkring. Denna metod är de så kallade generaliserade linjära modellerna. Dessa modeller är robusta, pålitliga och tolkningsbara, vilket är en av anledningarna till att de fortfarande används idag. Dock ser vi en tydlig utveckling inom maskininlärning där metoder blir allt bättre och ger därmed mer precisa prediktioner. I denna rapport kommer två maskininlärningsmetoder att användas: Gradient Boosting Machine (GBM) och Random Forest. Båda dessa modeller är trädmodeller, men även black-box modeller. Detta innebär det går att observera dess indata och utdata, men själva processen som sker däremellan är oklar eller komplex. Detta leder till deras nackdel, de är svårtolkade och det finns olika sätt att öka förståelsen för modellerna, men dessa ger oftast bara en lokal perspektiv eller gäller endast en del av en större datamängd. Ett alternativt sätt att göra dessa maskininlärningsmetoder mer tolkbara är att använda surrogatmodeller. I denna rapport kommer enkla beslutsträd att anpassas på black-box-modellernas prediktioner. Genom att använda det befintliga datamaterialet "MCcase" med färdiga tariffer kan vi undersöka om någon av black-box-modellerna presterar bättre än GLM och även bedöma hur bra dessa surrogatmodeller blir. Resultatet visade att endast Random Forest-modellen och dess surrogatmodell presterade bättre än GLM för detta specifika datamaterial. Dock presterade alla modeller på en liknande nivå. En möjlig anledning till att GBM presterade sämst kan vara bristen på korrelation i datamaterialet, eftersom en av GBMs främsta egenskaper är att hantera sådana korrelationer. Trots detta kunde vi visa att surrogatmodellerna uppfyllde sitt syfte och kan eventuellt erbjuda en ny metod för premieskattningar.

*Postadress: Matematisk statistik, Stockholms universitet, 106 91, Sverige.
E-post: filip.axee@gmail.com. Handledare: Mathias Millberg Lindholm.

Abstract

For many years, the same method has been used for pricing in property insurance. This method is known as generalized linear models (GLM), which are robust, reliable, and interpretable, making them still used today. However, we are witnessing a clear development in machine learning where methods are becoming increasingly better, thus providing more accurate predictions. In this report, two machine learning methods will be used: Gradient Boosting Machine (GBM) and Random Forest. Both of these models are tree-based models but also black-box models. This means that while you can observe their input and output, the process that occurs in between, for a specific dataset is har to interpret. There are different ways to increase understanding of the models, but these often provide only a local perspective or apply to only a part of a larger dataset. An alternative way to make these machine learning methods more interpretable is to use surrogate models. In this report, simple decision trees will be fitted to the predictions of the black-box models. By using the existing dataset 'MCcase" with predefined tariffs, we can investigate if any of the black-box models perform better than GLM and also assess the quality of these surrogate models. The results showed that only the Random Forest model and its surrogate model outperformed GLM for this specific dataset. However, all models performed at a similar level. One possible reason for GBM performing the worst could be the lack of correlation in the dataset, as one of GBM's main strengths is handling such correlations. Despite this, we were able to demonstrate that the surrogate models served their purpose and could potentially offer a new method for premium estimations.

Förord

Jag vill rikta ett stort tack till min handledare Mathias Millberg Lindholm för all vägledning och stöd som han har gett mig under mitt examensarbete. Jag vill även tacka min familj, men framförallt min farmor för hennes stora intresse för min skolgång och arbete.

Innehåll

1 Introduktion **3**
- 1.1 Inledning . . . 3
- 1.2 Syfte & metod . . . 4

2 Teori **4**
- 2.1 Prissättning inom sakförsäkring . . . 4
  - 2.1.1 Premie och begrepp . . . 5
  - 2.1.2 Modellantaganden . . . 7
  - 2.1.3 Väntevärde och varians . . . 7
- 2.2 Generaliserade linjära modeller (GLMs) . . . 7
  - 2.2.1 Exponentiella spridningsmodeller . . . 8
  - 2.2.2 Tweedie modeller . . . 9
  - 2.2.3 Länkfunktion . . . 9
  - 2.2.4 Parameterskattning, maximum-likelihood . . . 10
  - 2.2.5 Generaliserade additiva modeller (GAM), splines . . . 11
- 2.3 Trädbaserade modeller . . . 11
  - 2.3.1 Beslutsträd . . . 11
  - 2.3.2 Bagging . . . 13
  - 2.3.3 Random forest . . . 14
  - 2.3.4 Boosting . . . 15
  - 2.3.5 Gradient boosting machine (GBM) . . . 16
- 2.4 Utvärdering av modell . . . 17
  - 2.4.1 Bias och varians tradeoff . . . 17
  - 2.4.2 Korsvalidering . . . 18
  - 2.4.3 Black-box modell . . . 19
    - 2.4.3.1 Variable Importance . . . 19
    - 2.4.3.2 Partial dependence plots (PDP) . . . 20
  - 2.4.4 GLM vs Black-box modell . . . 20
    - 2.4.4.1 MSE . . . 21
    - 2.4.4.2 Devians . . . 21
- 2.5 Surrogatmodell . . . 22
  - 2.5.1 Fidelity . . . 22

3 Data **22**

4 Modellanpassning **26**
- 4.1 GLM . . . 26
  - 4.1.1 Skadefrekvens . . . 27

4.2 GBM . . . 28
4.2.1 Skadefrekvens . . . 28
4.2.1.1 Surrogatmodell . . . 30
4.3 Random forest . . . 35
4.3.1 Skadefrekvens . . . 36
4.3.1.1 Surrogatmodell . . . 36

5 Resultat **40**

6 Sammanfattning & diskusion **46**

7 Appendix **48**
7.0.1 GLM . . . 48
7.0.1.1 Likelihood-kvot test . . . 48
7.0.1.2 AIC . . . 48
7.1 Figurer . . . 49

Referenser **54**

1 Introduktion

1.1 Inledning

Försäkringar har funnits i olika former, och sträcker sig hela vägen tillbaka till antiken. Grunden för en försäkring är att sprida risken mellan individer i en (stor) population (för att gemensamt kunna täcka kostnader som en enskild individ inte kunde klara av). Det fanns till exempel byinvånare som hjälpte till att bygga om ens hus ifall det brann ned. Det som främst av allt förknippas med den moderna försäkringsindustrin, som för Europa har sina rötter under 1600-talet är försäkringar för sjöfart. Genom att sälja försäkringar mot att fartygen inte skulle återvända (kapsejsa), och genom att försäkra flera fartyg spreds riskerna ut men även möjliggjorde att fler expeditioner kunde genomföras. Det var även under denna tid som de aktuariella beräkningarna började tas fram. De som beräknade risken att ett fartyg inte återvände bäst, kunde på så sätt genera lönsamma affärerna.

Inom sakförsäkring har prissättningen betraktat olika egenskaper hos det försäkrade objektet men även hos försäkringstagaren. Detta ledde fram till så kallade tariffer, där den huvudsakligen källan till beslut för att skapa en modell kring skadekostnaden kommer från försäkringsbolagets historiska data om försäkringar och skador, men kan även kompletteras med data från andra håll. Det sedan på 90-talet som de brittiska aktuarierna introducerade generaliserade linjära modeller (GLM) som ett verktyg för tariffanalyser. Denna metod har blivit en standard i många länder och används även i störst utsträckning även idag.

Metoder inom statistisk inlärning har blivit alltmer populära senaste åren, eftersom maskininlärnings-algoritmer lyckas hantera kontinuerliga kovariater, samspelseffekter och kolinjära prediktorer på ett bra sätt. Detta är även något som lyfts fram i artikeln “Toward an explainable machine learning model for claim frequency: a use case in car insurance pricing with telematics data” (Maillart, 2021). Artikeln lyfter även fram att den generaliserade linjära modellen fortfarande används i stor utsträckning, på grund av att dessa är lätta att tolka. Dock visar det sig att GLMs ofta inte är lämpat för data när förhållandet mellan kovariaten och responsvariabeln till exempel inte är linjärt, vilket medför att det blir nödvändigt att diskretisera de kontinuerliga variablerna (Maillart, 2021). Nackdelen med maskininlärning är att dessa modeller är svåra att tolka, och oftast ligger fokus på modellens prediktiva förmåga. Det finns olika sätt att få en förståelse för maskinlärningsmodeller, dock är det ofta metoder som endast ger ett lokalt perspektiv, vilket innebär att fokus ligger på att förstå funktioner som är specifika för en viss del av en större datamängd (som exempelvis LIME eller SHAP) (Maillart, 2021). Det andra problemet med dessa metoder är att de ger för lite information för att få en någorlunda bra förståelse över modellen (exempelvis VIP) (Maillart, 2021) .

I denna rapport kommer fokus ligga på att undersöka hur bra de trädbaserade modeller-

na Random forest och gradient boosting trees egentligen är i jämfört med den klassiska generaliserade linjära modellen. Finns det någon fördel att använda dessa black-box modeller, och om det finns en fördel, är den så pass stor att det är värt förlora den åtråvärda egenskapen kring tolkningsbarhet som GLM erhåller? En annan aspekt som kommer undersökas är om det på något sätt går att utöka förståelsen till dessa modeller genom så kallade surrogatmodeller, och hur pass tolkningsbara dessa i så fall blir. En nackdel med att skapa surrogatmodeller är att förmågan att få en bra prediktion eventuellt förloras i takt med konverteringen till en mer tolkningsbar modell. Om detta skulle vara fallet, kan det då optimeras så att den förlorade prediktionsförmågan minimeras medan det ändå blir en tolkningsbar surrogatmodell?

1.2 Syfte & metod

Syftet med denna rapport är att få en ökad förståelse för maskininlärningsmetoder och deras potentiella användning för att skapa bättre prediktioner av premie än populära generaliserade linjära modeller. Det är även av intresse att undersöka om black-box-modeller som GBM och Random Forest kan konverteras till mer tolkningsbara modeller genom surrogatmodeller utan att förlora alltför mycket i sin prediktionsförmåga. Detta skulle möjliggöra ersättning av GLM med maskininlärningsmetoder samtidigt som de uppfyller regelverk som gäller för premieberäkningar, till exempel att kunna förklara varför en kund får en specifik premie.
Genom att använda statistikprogrammet R (R Core Team, 2020) kan metoder som GLM, GBM och Random Forest anpassas för att sedan utvärderas i jämförelse med GLM. Förutom de standardfunktioner som finns i R kommer även funktioner som “gbm” (Greenwell, Boehmke, Cunningham & Developers, 2022), “randomForest” (Liaw & Wiener, 2002) och “splines” (R Core Team, 2022) att användas.
Vanliga beslutsträd kommer också att användas som surrogatmodeller och anpassas med hjälp av funktionen ‘rpart” (Therneau & Atkinson, 2022). Detta kommer att bidra till ökad tolkningsbarhet. För att kunna anpassa dessa modeller krävs även en förståelse för olika anpassningsmått såsom MSE, devians och fidelity.

2 Teori

2.1 Prissättning inom sakförsäkring

Innan vi går in på detaljer kring modellering och analys så kommer vi i detta kapitel gå igenom problemet bakom prissättning och definiera grundläggande koncept och antaganden. En icke-liv försäkringspolicy (sakförsäkring) är ett kontrakt mellan en försäkringsgivare (ett försäkringsbolag) och en försäkringstagare (kund). Dessa parter har kommit överens om att försäkringstagaren kommer motta en ersättning av försäkringsgivaren om en skadehändelse som skapar ekonomisk förlust inträffar. Notera att denna ersättning består av pengar, samt

att denna skada ska ha skett inom en specifik period, oftast ett år och omfattar vissa typer av skadehändelser. För att få tillgång till detta skydd så betalar försäkringstagaren en avgift till försäkringsgivaren, det är detta som kallas premie. En sakförsäkring kan omfatta till exempelvis skador på en bil, hus, eller annan egendom. Det kan även omfatta personer, och juridiska kostnader. I stora drag är sakförsäkring all annan försäkring som inte går under livförsäkring. Genom detta försäkringskontrakt, så förflyttas den ekonomiska risken från försäkringstagaren till försäkringsgivaren (Ohlsson & Johansson, 2010). Eftersom premien sätts före en skada har inträffat, så är skadekostnaden okänd vilket gör att försäkringsgivaren måste ha en god förståelse för risken i sin portfölj. Genom "De stora talens lag", så är förlusten hos försäkringsbolaget summan av många jämförelsevis små oberoende förluster, mycket mer förutsägbar än för endast en individ. Detta leder till att den relativa kostnaden inte bör vara långt ifrån det förväntade värdet (Ohlsson & Johansson, 2010). Detta ger oss den allmänt tillämpade principen att premien ska baseras på den förväntade kostnaden som överförs från försäkringstagaren till försäkringsgivaren (Ohlsson & Johansson, 2010). På grund av att de förväntade kostnaderna varierar mellan olika försäkringar så finns det ett behov av statiska modeller. Skadefrekvensen är inte densamma för alla försäkringar, och om en skada uppkommer så är den förväntade skadekostnaden inte heller samma för alla försäkringar. Exempelvis så kan de flesta hålla med om att en brandförsäkring för en stor vila bör ha en större premie än en liten stuga. I dagens samhälle har marknaden i många länder blivit avreglerade vilket har lett till fri konkurrens. Detta är på grund av att om ett företag tar ut för hög premie för vissa typer av försäkringar, så kommer dessa kunder flytta sin försäkring till en konkurrent med en mer rättvis premie. I fallet då ett försäkringsbolag istället tar ut en för låg premie för grupp 1 med hög risk och för hög av grupp 2 med låg risk. Det skulle leda till att grupp 2 lämnar för konkurrenter och fler av typen grupp 1 attraheras. Detta är även detta som kallas för "adverse selection" och kommer leda till ekonomiska förluster, både genom att attrahera underprissatta försäkringar och förlora lönsamma (Ohlsson & Johansson, 2010).

2.1.1 Premie och begrepp

För att prissätta en sakförsäkring, så måste försäkringsgivare som prediktera kostnaden Y. Detta kan genomföras baserat på olika kategoriska kovariater X (rating factors) som sedan används i en funktion f, $\hat{Y} = f(X)$. Dessa faktorer brukar oftas tillhöra någon av följande kategorier (Ohlsson & Johansson, 2010):

- **Egenskaper för försäkringstagaren**, exempelvis ålder eller kön på försäkringstagaren
- **Egenskaper för det försäkrade objektet**, exempelvis ålder eller modell på bil
- **Egenskaper för den geografiska regionen**, exempelvis inkomst per person eller befolkningstäthet i försäkringstagarens bostadsområde

Försäkringskontrakt som har samma klasser för varje kovariat, sägs tillhöra samma tariffcell och ges samma premie. Ofta modelleras premien genom att skapa två separata modeller för skadefrekvensen och skadekostnaden, vilket ger oss:

$$\text{Riskpremie} = \text{Förväntad skadefrekvens} \cdot \text{Förväntad medelskadekostnad} \tag{1}$$

Notera att i (1) är riskpremien samma sak som den förväntade totala skadekostnaden delat på duration. Det är även viktigt känna till dessa begrepp (Ohlsson & Johansson, 2010):

duration av ett kontrakt, detta är tiden som kontraktet har varit aktivt och mäts oftast i år.

Skada, detta är händelse som har blivit rapporterad av försäkringstagaren där den begär en ekonomisk kompensation.

Skadefrekvens, detta är antalet skador dividerat på durationen för en grupp kontrakt som är aktiva under en specifik period.

Medelskadekostnad, detta är totala kostnaden dividerat med antalet skador, det vill säga medelkostnaden för ett kontrakt.

Det finns även så kallade nyckeltal, vilket är av samma typ som ovan, det vill säga ett förhållande mellan utfallet av en slumpmässig variabel och ett volymmått. Detta volymmått är det som sedan kommer kallas exponering. Exponeringen w kommer i uttryck i responsen Z, exempelvis antalet skador eller skadebeloppet. Analysen är sedan utförd på nyckeltalet $Y = Z/w$, istället för enbart responsvariabeln själv. Exponeringen har stor fundamental roll i analysen, eftersom mer exponering i data leder till mindre variation mellan nyckeltalen (Ohlsson & Johansson, 2010). I Tabell 1 visas exempel på viktiga nyckeltal och hur de beräknas.

Exponering w	Respons Z	Nyckeltal $Y = Z/w$
duration	Antal skador	Skadefrekvens
duration	Skadekostnad	Riskpremie
Antal skador	Skadekostnad	Medelskadekostnad
Intjänad premie	Skadekostnad	Skadeprocent
Antal skador	Antal stora skador	Andel stora skador

Tabell 1: Exempel på viktiga nyckeltal

2.1.2 Modellantaganden

I detta kapitel kommer det presenteras tre stycken antaganden som kommer ge en grund för att anpassa statistiska modeller (Ohlsson & Johansson, 2010).

- **Oberoende kontrakt**, anta n kontrakt. Givet någon responstyp från Tabell 1, låt Z_i definiera responsen för kontrakt i. Då är $Z_i, \ldots, Z_n$ oberoende.
- **Tidsoberoende**, anta n disjunkta tidsintervall. Givet någon responstyp från Tabell 1, låt Z_i definiera responsen för kontrakt i. Då är $Z_i, \ldots, Z_n$ oberoende.
- **Homogenitet**, anta två kontrakt inom samma tariffcell, som har samma exponering. Givet någon responstyp från Tabell 1, låt Z_i definiera responsen för kontrakt i. Då har Z_1 och Z_2 samma sannolikhetsfördelning.

2.1.3 Väntevärde och varians

En analys på en tariff fokuserar på det förväntade värdet i en tariffcell men för att kunna en uppfattning kring precisionen i skattningarna behövs även kunskap kring variansen. Utgå från antaganden som är listade i Kapitel 2.1.2, samt anta godtyckligt nyckeltal $Y = Z/w$ för en grupp kontrakt i en tariffcell med total exponering w och total respons Z. Anta dessutom att w är antalet skador så att Z kan skrivas som en summa av w individuella responser $Z_1, \ldots Z_w$. Notera att antaganden om oberoende kontrakt och tidsoberoende medför att alla Z_k är oberoende, eftersom skador som erhålls från olika kontrakt eller tidpunkter. Homogenitet medför identisk fördelning, detta innebär att väntevärdet kan skrivas enligt följande $\mathrm{E}[Z_k] = \mu$ och variansen $\mathrm{Var}(Z_k) = \sigma^2$ för något μ och σ^2 (Ohlsson & Johansson, 2010). Detta tillsammans med elementära regler kring väntevärde och varians ger följande:

$$\begin{aligned} \mathrm{E}[Z] &= w\mu, \quad \mathrm{Var}(Z) = w\sigma^2, \\ \mathrm{E}[Y] &= \mu, \quad \mathrm{Var}(Y) = \sigma^2/w. \end{aligned} \tag{2}$$

Notera att (2) även gäller ifall exponeringen är uttryckt i intjänad premie eller duration (Ohlsson & Johansson, 2010).

2.2 Generaliserade linjära modeller (GLMs)

Generaliserade linjära modeller (GLMs) är modeller som generaliserar de klassiska regressionsmodellerna, som antar normalfördelning. Denna modell specificeras av tre komponenter (Agresti, 2002):

- **Slumpmässig komponent:** Responsvariabeln Y har oberoende observationer $(y_1, \ldots, y_N)$ där fördelningen tillhör den exponentiella familjen.

- **Systematisk komponent:** Sammankopplar en vektor $(\eta_1, \ldots, \eta_N)$ med p kovariater genom en linjär modell. Låt x_{ij} vara värdet på kovariat j, $j =, 1, 2, \ldots, r$ för subjekt i, då får vi följande:
$$\eta_i = \sum_j x_{ij}\beta_j, \quad i = 1, \ldots, N.$$
Denna linjära kombination av kovariat, kallas även för den linjära prediktorn.

- **Länkfunktion:** En så kallad länkfunktion kopplar samman de slumpmässiga och systematiska komponenterna. Låt $\mu_i = \mathrm{E}[Y_i], i = 1, \ldots, N$. Modellen länkar μ_i till η_i genom $\eta_i = g(\mu_i)$. Där g är en monoton och differentierbar länkfunktion. Detta ger att g länkar $\mathrm{E}[Y_i]$ till kovariaten genom formeln:
$$g(\mu_i) = \eta_i = \sum_j x_{ij}\beta_j, \quad i = 1, \ldots, N.$$
Länkfunktionen $g(\mu) = \mu$, kallas för identitetslänk och ger $\eta_i = \mu_i$. Detta är även länkfunktionen som används vid vanliga regressionen där Y är normalfördelad.

För att sammanfatta; en GLM är en linjär modell för ett transformera medelvärdet av responsvariabeln som har en fördelning som tillhör den exponentiella familjen(Agresti, 2002).

2.2.1 Exponentiella spridningsmodeller

Exponentiella spridningsmodeller även kallade Exponential dispersion models (EDMs) används för att generalisera normalfördelningen i linjära modeller. Genom att antaganden om att $Y_1, \ldots, Y_N$ är oberoende, vilket även är ett krav generellt för GLM teori, se Kapitel 2.1.2. Ger att sannolikhetsfördelningen av en EDM är given av följande frekvensfunktion, som specialiserar sig till en sannolikhetsdensitetsfunktion i det kontinuerliga fallet och en sannolikhetstäthetsfunktion i det diskreta fallet (Ohlsson & Johansson, 2010):

$$f_{Y_i}(y_i; \theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi/w_i} + c(y_i, \phi, w_i)\right\}.$$

Där θ_i är en parameter som är tillåten att bero på i, medan spridningsparametern $\phi > 0$ är konstant för alla i. $b(\theta_i)$ är den så kallade kumulativa funktionen, denna antas vara differentierbar två gånger med en inverterbar förstaderivata. Funktionen $c(\cdot)$ beror inte på θ_i, har ett väldigt litet intresse inom teorin för GLM och kommer därav inte ges några detaljer kring (Ohlsson & Johansson, 2010). För varje val av en sådan funktion, erhålls en familj av sannolikhetsfördelningar som exempelvis normal, Poission och gamma fördelningar. Notera att det går att härleda variansen och väntevärdet för EDMs till följande (Ohlsson & Johansson, 2010):

$$\begin{aligned} \mathrm{E}[Y] &= b'(\theta) = \mu \\ \mathrm{Var}(Y) &= b''(\theta)\phi/w \end{aligned} \tag{3}$$

Eftersom $\mu = \mathrm{E}[Y] = b'(\theta)$, samt att det antas vara en inverterbar funktion kan inverterade förhållandet skrivas $\theta = b'^{-1}(\mu)$ i $b''(\theta)$ för att erhålla den så kallade variansfunktionen $v(\mu) = b''(b'^{-1}(\mu))$. Detta leder till att variansen $\mathrm{Var}(Y)$ kan uttryckas som en produkt av variansfunktionen $v(\mu)$, en skalnings- och viktfaktor ϕ/w (Ohlsson & Johansson, 2010).

2.2.2 Tweedie modeller

Inom sakförsäkring är det ofta önskvärt att arbeta med sannolikhetsfördelningar som är sluten med avseende på skaltransformeringar, eller skalärt oberoende. Låt c vara en positiv konstant och Y en slumpmässig variabel från en specifik familj av fördelningar. Denna familj uppfyller att skalning är invariant om cY följer en fördelning inom samma familj. Denna egenskap är något som är önskvärt om Y mäts i en monetär enhet, exempelvis ska det inte ha någon betydelse om skadefrekvensen mäts i promille eller procent (Ohlsson & Johansson, 2010).
Det visar sig att de enda EDMs som har denna egenskap, att skalningen är invariant är så kallade Tweedie modeller. Dessa modeller är definierade att ha följande variansfunktion för något p

$$v(\mu) = \mu^p.$$

s Fallet då $p = 0$ ger normalfördelningen, $p = 1$ ger en poissonfördelning vilket oftast används vid modellering av skadefrekvensen. Vid modellering av medelskadekostnaden är ett populärt val $p = 2$ vilket motsvarar en gammafördelning, och de fall där riskpremien modelleras direkt används ett $1 < p < 2$ som i detta fall är sammansatt Poissonfördelning (Ohlsson & Johansson, 2010).

2.2.3 Länkfunktion

Länkfunktionen är ett fundamentalt objekt hos en GLM, eftersom den länkar ihop medelvärdet till en linjär struktur. Genom att anta r parametrar $\beta_1, \beta_2, \ldots, \beta_r$, ges sammankopplingen av följande (Ohlsson & Johansson, 2010):

$$g(\mu_i) = \eta_i = \sum_{j=1}^{r} x_{ij}\beta_j \tag{4}$$

Notera att detta även har beskrivits i föregående kapitel och att x_{ij} i (4) precis som tidigare motsvarar till värdet av kovariat x_j för observation i. Det ska även poängteras att det finns olika länkfunktioner, och valet av länkfunktion beror oftast på vilken typ av analys som genomförs. Inom sakförsäkring är den mest vanliga länkfunktionen, log-länkfunktion,

eftersom den ger en multiplikativ modell (oftast de mest rimliga modellerna), men för exempelvis analyser av proportioner är det vanligt att använda en logit-länkfunktion (Ohlsson & Johansson, 2010). Nedanför visas log-länkfunktionen:

$$g(\mu_i) = \log(\mu_i). \tag{5}$$

En annan länkfunktion som används är den så kallade identitets-länken och som används av linjära modeller

$$g(\mu_i) = \mu_i. \tag{6}$$

Notera att länkfunktionen i både (5) och (6) inte beror på i, detta är på grund av att en länkfunktion tillåts nämligen inte bero på i (Ohlsson & Johansson, 2010).

2.2.4 Parameterskattning, maximum-likelihood

Det viktigaste steget är parameterskattningen av kovariaten β, vilket även ger de så kallade relativiteterna, som är grundstenarna i en tariff. I det generella fallet av att beräkna MLEs för β-kovariaten i en GLM, så är de estimerade på urval av n observationer. De inviduella observationerna följer EDM-fördelningen som är given enligt formen i Kapitel 2.2.1, dessutom på grund genom oberoende, är log-likelihood som en funktion av $\boldsymbol{\theta}$ följande:

$$l(\boldsymbol{\theta}; \boldsymbol{\phi}, \mathbf{y}) = \frac{1}{\phi} \sum_i w_i(y_i\theta_i - b(\theta_i)) + \sum_i c(y_i, \phi, w_i) \tag{7}$$

Det visar sig tydligt att överspridningsparametern ϕ påverkar inte maximeringen av l med avseende till $\boldsymbol{\theta}$, vilket även bör observeras för den linjära regressionsmodellen, där ϕ betecknas σ^2. Detta leder till att ϕ kan bortses i denna beräkning. En likelihood som betraktar en funktion av $\boldsymbol{\beta}$ istället för $\boldsymbol{\theta}$, kan erhållas genom inversen av förhållandet $\mu_i = b'(\theta)$, i kombination med att länken $g(\mu_i) = \eta_i = \sum_j x_{ij}\beta_j$. Derivatan av l med avseende till β_j kan erhållas genom kedjeregeln och ges av följande (Ohlsson & Johansson, 2010):

$$\frac{\partial l}{\partial \beta_j} = \sum_i \frac{\partial l}{\partial \theta_i}\frac{\partial \theta_i}{\partial \beta_j} = \frac{1}{\phi}\sum_i (w_i y_i - w_i b'(\theta_i))\frac{\partial \theta_i}{\partial \beta_j} = \frac{1}{\phi}\sum_i (w_i y_i - w_i b'(\theta_i))\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j} \tag{8}$$

Eftersom vetskapen kring relationen $\mu_i = b'(\theta_i)$ erhålls $\partial\mu_i/\partial\theta_i = b''(\theta_i)$. Inversens derivata av denna relation, fås genom att ta inversen av derivatan, det vill säga $\partial\theta_i/\partial\mu_i = 1/v(\mu_i)$, och enligt definition $v(\mu_i) = b''(\theta_i)$. Dessutom har vi att $\frac{\partial \mu_i}{\partial \eta_i} = \left[\frac{\partial \eta_i}{\partial \mu_i}\right]^{-1} = \frac{1}{g'(\mu_i)}$ och från $\eta_i = \sum_j x_{ij}\beta_j$ ger att $\partial\eta_i/\partial\beta_j = x_{ij}$. Genom att sätta ihop alla dessa resultat ger då följande:

$$\frac{\partial l}{\partial \beta_j} = \frac{1}{\phi} = \sum_i w_i \frac{y_i - \mu_i}{v(\mu_i)g'(\mu_i)}x_{ij} \tag{9}$$

Ekvation (9) är den så kallad score-funktionen (Ohlsson & Johansson, 2010). Genom att sätta alla r partiella derivator till noll samt multiplicera med ϕ ger ML ekvationerna

$$\sum_i w_i \frac{y_i - \mu_i}{v(\mu_i)g'(\mu_i)} x_{ij} = 0, \quad j = 1, 2, \ldots, r. \tag{10}$$

Det kan helt enkelt se ut som att lösningen till (10) helt enkelt är $\mu_i = y_i$, dock stämmer inte detta eftersom att $\mu_i = \mu_i(\boldsymbol{\beta})$ måste även uppfylla sambandet som ges av regressionen på x-värdena, det vill säga följande (Ohlsson & Johansson, 2010):

$$\mu_i = g^{-1}(\eta_i) = g^{-1}(\sum_j x_{ij}\beta_j) \tag{11}$$

Notera att det endast är den så kallade mättade (saturated) modellen, där antalet parametrar är lika med antalet observationer som tillåter lösningen $\mu_i = y_i$ (Ohlsson & Johansson, 2010).

2.2.5 Generaliserade additiva modeller (GAM), splines

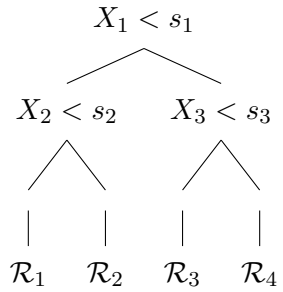
I det generella fallet behandlar GLM endast diskreta kovariater. Det ska dock poängteras att inom nästan all prissättning för sakförsäkring behandlas kontinuerliga kovariater, som exempelvis ålder genom att gruppera värdena i intervall och därefter behandla värden inom samma intervall som identiska. Denna åtgärd är också den vanligaste för att hantera detta problem. Denna uppdelning kan baseras på risk, men även på duration. Nackdelen med denna metod är att premien för två försäkringar med olika men närliggande kovariatvärden kan erhålla en stor variation om dessa värden skulle råka tillhöra olika intervall. Detta problem leder till att en bättre modell bör användas, där den kontinuerliga effekten inkorporeras. Ett sätt är att använda polynomregression eller så kallade "splines". I boken (Ohlsson & Johansson, 2010) beskrivs splines i Kapitel 5. Metoden kan beskrivas kortfattat som följande: anpassa ett antal polynom, definierade på en sammankopplad uppsättning disjunkta intervall, så att ändpunkterna i polynomen binds ihop på ett sådant sätt att ett visst antal derivator i dessa punkter matchar (Ohlsson & Johansson, 2010).

2.3 Trädbaserade modeller

2.3.1 Beslutsträd

Beslutsträd är en icke-linjär modell som kan appliceras på både klassificerings- och regressionsproblem. Modellen delar upp data baserat på ja- och nej-frågor, varpå vi sedan erhåller olika regioner $\mathcal{R}_i$. Inom dessa regioner gör vi sedan en prediktion för de givna observationerna. För att göra trädet mer begripligt och lätt att förstå beskrivs regionerna $\mathcal{R}_i$ som löv ("terminal nodes") av trädet. Observera att beslutsträd oftast ritas upp-och-ner, vilket innebär att löven befinner sig längst ner på trädet. När trädet delar på sig kallas "split", och

dessa punkter kallas "internal nodes". Varje segment av trädet som sammankopplar noderna kallas "grenar" (James, Witten, Hastie & Tibshirani, 2013). Nedan visas ett exempel på hur ett träd kan se ut.



Processen av att bygga ett regressionsträd kan i grova drag delas upp i två steg (James m. fl., 2013):

1. Dela upp kovariatrummet, det vill säga, uppsättningen av möjliga $x_1, x_2, \ldots, x_p$ i J distinkta och icke-överlappande regioner $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_J$.
2. För varje observation som tillhör region $\mathcal{R}_j$, ges samma prediktion. Detta beror på vilken förlustfunktion som minimeras.

Notera att denna prediktor kan skrivas på följande sätt (Hastie, Tibshirani, Friedman & Friedman, 2009):

$$f(x) = \sum_{j=1}^{J} \gamma_j I\{x \in \mathcal{R}_j\} \tag{12}$$

I (12) är γ_j konstanter. Genom att utgå ifrån den vanligaste förlustfunktionen, minsta-kvadratfelet, kommer dessa konstanter betraktas som medelvärdet av responsvärdet av träningsobservationerna i $\mathcal{R}_j$. Detta ger oss att den bästa skattningen ges av följande (Hastie m. fl., 2009):

$$\hat{\gamma}_j = \text{ave}(y_i | x_i \in \mathcal{R}_j) \tag{13}$$

Om vi utgår från det enklaste problemet, minsta-kvadratfelet som kriteriet för minimering, får vi att den bästa binära partitionen av detta kriterium är i allmänhet beräkningsmässigt omöjligt. Därav kommer vi använda oss av följande algoritm:
Börja med att all data, utse en split-variabel j och splitpunkt s, och definiera paret av halvplan:

$$\mathcal{R}_1(j, s) = \{X | X_j \leq s\} \text{ och } \mathcal{R}_2(j, s) = \{X | X_j > s\} \tag{14}$$

Detta leder till att vi söker split-variabeln j och splitpunkten s som minimerar

$$\arg\min_{j,s}(\min_{\gamma_1} \sum_{x_i \in \mathcal{R}_1(j,s)} (y_i - \gamma_1)^2 + \min_{\gamma_2} \sum_{x_i \in \mathcal{R}_2(j,s)} (y_i - \gamma_2)^2) \tag{15}$$

Oavsett val av j och s så ges lösningen till den inre minimeringen av (13) (Hastie m. fl., 2009). Notera att (15) kan skrivas på ett mer allmänt sätt:

$$\arg\min_{j,s}(\min_{\gamma_1}\sum_{x_i\in\mathcal{R}_1(j,s)}L(y_i,\gamma_1)+\min_{\gamma_2}\sum_{x_i\in\mathcal{R}_2(j,s)}L(y_i,\gamma_2)) \tag{16}$$

Där $L(\cdot)$ är en förlustfunktion baserad på ett lämpligt val, som tidigare nämnts, där den enklaste och vanligaste är minsta-kvadratfelet.
Något som bör beaktas är hur stort trädet ska vara. Ett träd som är mycket stort kan överanpassa data, medan ett alltför litet träd kan leda till att strukturer missas. Trädstorleken är en så kallad hyperparameter ("tuning parameter") och bestämmer modellens komplexitet. Ett stort träd leder ofta till en komplicerad modell som är svår att tolka och instabil, medan små träd oftast är mer tolkningsbara (Hastie m. fl., 2009). Notera att vi inte kommer att visa klassifikationsträd här, men dessa är snarlika regressionsträd förutom att de används för att förutsäga en kvalitativ respons istället för en kvantitativ respons (James m. fl., 2013). Vi kommer att fokusera på att modellera skadefrekvensen i denna rapport, vilket faller under regressionsträd.

2.3.2 Bagging

Bagging eller bootstrap aggregation är en teknik som används för att reducera variansen i en estimerad prediktionsfunktion. Denna teknik fungerar ofta mycket bra för högvariansprocesser med låg bias, som trädmodeller. Vilket innebär att en modell har en hög grad variation i modellens prestanda när den appliceras på olika datamängder, samt har en låg grad av systematisk felaktighet. Mer om detaljer om detta framgår i senare kapitel. Genom att använda bagging-tekniken på en högvarians, lågbias-process, som en trädmodell, kan man förbättra modellens prestanda genom att minska variationen i modellens prediktioner. För att applicera bagging-tekniken på en trädmodell, anpassas samma träd flera gånger på olika versioner av bootstrap-data från träningsdatan, och resultatet blir sedan medelvärdet av de olika träden (Hastie m. fl., 2009).
Om vi antar n oberoende observationer $y_1,\ldots,y_n$ med en varians på σ^2 för varje observation. Vidare antas att variansen för medelvärdet $\bar{y}$ av dessa observationer är σ^2/n. Detta innebär att genom att ta medelvärdet av en uppsättning observationer minskar variansen (James m. fl., 2013). Det är därför naturligt att reducera variansen och öka prediktionens träffsäkerheten för en statistisk inlärningsmetod genom att använda många uppsättningar av data (träningsset), konstruera en separat prediktionsmodell baserat på varje träningsset och sedan ta medelvärdet av prediktionerna. Detta kan beskrivas genom att beräkna $\hat{f}^1(x),\hat{f}^2(x),\ldots,\hat{f}^B(x)$ där B är antalet separata träningsset. Sedan används medelvärdet för att erhålla en enda lågvarians statistisk inlärningsmodell som ges av följande formel (James m. fl., 2013):

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(x).$$

Notera att detta inte är praktiskt möjligt eftersom det normalt sett inte finns tillgång till flera träningsset. Det är här bootstrap kommer in i bilden genom att ta observationer upprepade gånger från endast ett träningsset. Denna process genererar B olika bootstrap-träningsset. Genom att sedan använda det b:te bootstrap-träningssetet kan metoden tränas för att erhålla $\hat{f}^{*b}(x)$, varvid medelvärdet sedan kan beräknas för alla prediktioner. Detta ger oss därav följande, som även kallas för bagging (James m. fl., 2013):

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

2.3.3 Random forest

Random forest anses vara en förbättring jämfört med bagging genom en liten justering som dekorrelerar träden. Precis som i bagging så anpassas ett antal beslutsträd på bootstrap-data. Skillnaden vid denna modellering av dessa beslutsträd, är att vid varje split i ett träd så kan split-kovariaterna endast väljas bland ett slumpmässigt urval av m kovariater från hela uppsättningen av r kovariater (James m. fl., 2013). Spliten kan enbart använda en av dessa m kovariater, vid nästa split slumpas m nya kovariater ut. Notera att det är vanligt att $m = \sqrt{r}$. Random forest kan även beskrivas som att algoritmen vid varje split i trädet inte tillåts att välja bland majoriteten av alla tillgängliga kovariater (James m. fl., 2013). Notera att detta förebygger att ett fåtal starka kovariater enbart används, det vill säga de som har en hög korrelation med responsvariabeln. Låt oss anta att det finns en mycket stark kovariat i datasetet tillsammans med ett antal måttligt starka kovariater. Detta leder till att vid bagging-träd kommer alla eller de flesta av dessa träd att använda den starka kovariaten som sin första split. Konsekvensen blir då att alla bagging-träd blir väldigt lika varandra. Prediktionerna från dessa träd är därför starkt korrelerade. Notera att medelvärdet av många högt korrelerade kovariat inte leder till lika stor minskning av variansen som medelvärdet av många okorrelerande kovariat. Detta innebär att bagging inte kommer leda till en avsevärd minskning av variansen för ett enskilt träd utifrån denna process (James m. fl., 2013). Genom att använda random forest kan detta problem undvikas eftersom algoritmen tvingas att endast välja bland en delmängd av kovariater vid varje split. Därför kommer i genomsnitt $(r - m)/r$ av splittarna inte att ta hänsyn till den starka kovariaten, och på så sätt kommer de resterande kovariaterna ha en större chans att bli använda. Denna process kan betraktas som att dekorrelera träden, vilket gör att medelvärdet av de resulterande träden blir mindre varierande och därmed mer tillförlitligt (James m. fl., 2013). Det är också värt att notera att den största skillnaden mellan bagging

och random forest är valet av storleken m för delmängden kovariat. Att använda $m = r$ i random forest är helt enkelt samma sak som bagging (James m. fl., 2013).

2.3.4 Boosting

Boosting är en ytterligare metod som syftar till att förbättra prediktionen av ett beslutsträd. På samma sätt som bagging är boosting en generell metod som kan tillämpas på många statistiska inlärningsmetoder. Som tidigare nämnts, innebär bagging att man skapar flera kopior av originalträningsdatan genom bootstrap, anpassar enskilda beslutsträd och sedan kombinerar alla dessa träd för att skapa en enda prediktionsmodell (James m. fl., 2013). Notera att varje träd anpassas till en datamängd som skapas av bootstrap oberoende av de andra träden. Boosting fungerar på liknande sätt, förutom att träden växer sekventiellt. Detta innebär att varje träd växer genom att använda information från det föregående trädet. Notera att boosting inte använder bootstrap, utan istället är varje enskilt träd anpassat till en modifierad version av originaldatamängden. Boosting innebär att kombinera ett stort antal beslutsträd $\hat{f}^1, \hat{f}^2, \ldots, \hat{f}^B$, och dess algoritm beskrivs i slutet av detta kapitel (James m. fl., 2013).
Idén bakom denna process är att, till skillnad från att anpassa enbart ett stort beslutsträd till data, vilket potentiellt skulle leda till överanpassning, lär sig istället boosting-metoden långsamt. Givet den nuvarande modellen anpassas ett beslutsträd till residualerna från modellen. Med andra ord anpassas ett träd med de nuvarande residualerna istället för resultatet Y som responsvariabel. Det nya beslutsträdet läggs sedan till i den anpassade funktionen för att uppdatera residualerna. Notera att varje träd kan vara relativt små, med endast ett fåtal terminalnoder, och antalet bestäms av parametern d i algoritmen. Genom att anpassa små träd på residualerna förbättrar vi långsamt $\hat{f}$ i områden som inte presterar/predikterar (perform well) bra. Den så kallade shrinkage-parametern λ saktar ner processen ytterligare, vilket möjliggör att fler och olika träd kan anpassas till residualerna. Generellt sett tenderar statistiska inlärningsmetoder som använder principen att lära sig långsamt att prestera bra. Notera att i boosting är konstruktionen av varje träd starkt beroende av föregående träd, vilket skiljer sig från bagging. Boosting använder sig av tre stycken hyperparametrar, vilket är följande (James m. fl., 2013):

1. Antalet träd B. Till skillnad från bagging och random forest, så kan ett för stort B leda till överanpassning i boosting. Det ska dock noteras att överanpassningen tenderar att att ske långsamt, om ens överhuvudtaget.

2. Shrinkage parametern λ, ett litet positivt tal. Denna kontrollerar vilken hastighet som boosting lär sig. Vanliga värden tenderar att vara 0.01 eller 0.001, och det korrekta valet beror på problemet. Små värden på λ kan kräva att väldigt stora värden på B används för att uppnå goda resultat.

3. Antalet splittar i varje träd d, kontrollerar komplexiteten i ensemblen. Något som har fungerat väl är när $d = 1$, detta innebär att varje träd en stubbe (stump) som

består av enda split. I detta fall ensemblen anpassar en additiv modell eftersom varje term endast involverar en enda kovariat. Mer generellt är d 'interaction depth" och styr ordningen för interaktion av modellen. Detta på grund av att d splittar kan som högst involvera d kovariater.

En gemensam faktor för hyperparamterar är att de bestäms med hjälp av korsvalidering, vilket beskrivs i Kapitel 2.4.2.

1. Sätt $\hat{f}(x) = 0$ och $r_i = y_i$ för alla i i träningsdata
2. För $b = 1, 2, \ldots B$, repetera:
   (a) Anpassa ett träd $\hat{f}^b$ med d splittar ($d + 1 =$ antalet terminalnoder) till träningsdata (X, r).
   (b) Uppdatera resiudalerna, $r_i \leftarrow r_i - \lambda \hat{f}^B(x_i)$.
3. Framställ den slutliga modellen

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

2.3.5 Gradient boosting machine (GBM)

Gradient Boosting Machine (GBM) grundar sig på metoden "steepest descent", där idén är att minimera en given förlustfunktion L genom att använda en ensemble av träd. Processen börjar med en terminalnod och lägger sedan till nya träd rekursivt. Rotnoden byggs vidare genom att röra sig i riktningen av den negativa gradienten av förlustfunktionen. Detta innebär att en godtycklig förlustfunktion $L(y_i, f(x_i))$ kan optimeras med följande gradient (Hastie et al., 2009):

$$g_{im} = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}} \tag{17}$$

Det är viktigt att notera att den negativa gradienten i ekvationen (17) refereras till som pseudo residualer och definieras som $r_{im} := -g_{im}$. Dessa residualer används sedan som värden vid anpassningen av de nya träden. Processen upprepas för ett förbestämt antal iterationer M. Algoritmen kan beskrivas med följande iterationer (Hastie et al., 2009):

1. Anpassa $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
2. För $m = 1, \ldots, M$:
   (a) För $i =, 1, \ldots, N$ beräkna

$$r_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}$$

(b) Anpassa ett regressionsträd för värdena r_{im} vilket ger terminal regionerna R_{jm}, $j = 1, 2, \ldots, Jm$.

(c) För $j = 1, 2, \ldots, J_m$ beräkna

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

(d) Uppdatera $f_m(x) = f_{m-1}(x) + \eta \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

3. Sätt till sist $\hat{f}(x) = f_M(x)$.

Det är även viktigt att notera att denna algoritm har tre olika hyperparametrar som bör optimeras: antalet iterationer M, storleken på varje träd J_m och inlärningshastigheten (shrinkage) η. Inlärningshastigheten styr inflytandet av varje nytt träd vid varje iteration och är en skalningsfaktor där $0 < \eta < 1$. Dessa hyperparametrar kan optimeras genom korsvalidering (Hastie et al., 2009).

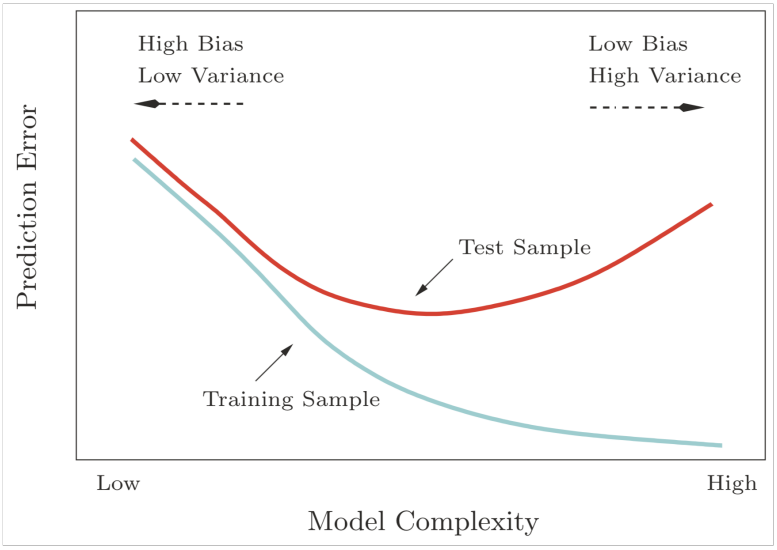
2.4 Utvärdering av modell

En viktig del av modelleringen är att utvärdera modellens prediktiva förmåga. Inom sakförsäkring är det främsta målet att få en så bra prediktion av skadefrekvensen som möjligt för nästkommande års kontrakt. En bra tolkning av modellen ger ett steg i rätt riktning i den processen, samt för att avgöra om det behövs någon ändring i den nuvarande tariffen. Förutom dessa aspekter finns det även andra som kan vara till stor nytta, exempelvis att kunna förklara för en kund orsaken till att de erhåller en viss premie, samt för skadeförebyggande åtgärder. Kunskap om kostnaden/skadefrekvensen för ett riskbeteende är även värdefullt eftersom försäkringsbolaget kan upplysa om dessa risker genom kampanjer och därmed försöka ändra försäkringstagarens beteende och på så sätt minska skadekostnaderna. Dessa upplysningar kan till exempel innebära att försäkringstagaren ska installera brandlarm, köra säkrare bilar eller vidta andra åtgärder som minskar risken (Johansson & Ohlsson, 2022). Detta leder in till följande kapitel där olika metoder kan hjälpa till att analysera modellens prestation, men också kovariaterna. Det kommer även att redovisas hur olika modeller kan jämföras mot varandra, även om modellerna är anpassade med hjälp av olika metoder.

2.4.1 Bias och varians tradeoff

Det finns ett generellt problem för prediktionsmodeller, och det är att finna en balans mellan modellens bias och varians. Bias kan beskrivas som modellens fel vid underanpassning på träningsdata. En hög bias kan bero på en för enkel modell, vilket skulle generera en dålig prediktion. Varians beskriver istället modellens överanpassning till träningsdata.

Notera att en överanpassning på träningsdata generar en hög varians vilket i sin tur leder till en dålig prediktion. Generellt så tenderar variansen att öka och bias att minska när modellkomplexiteten ökar, och tvärtom vid minskning av modellkomplexiteten (James m. fl., 2013). Detta är också vad som kallas för "bias-varians tradeoff", det vill säga att en minskning av bias ökar variansen, och vice versa. Som tidigare nämnt beror detta på att komplexa modeller är bättre anpassade till träningsdata, vilket ger låg bias, men leder till dålig prediktion på valideringsdata, hög varians. Detta är även något som visualiseras i Figur 1. Det optimala är att hitta en punkt där denna tradeoff ger bäst prediktionsförmåga för modellen. En metod som ofta används för att hitta den optimala punkten och därmed förbättra modellens anpassning på träningsdata och prediktionsförmåga på valideringsdata är korsvalidering, vilket även kommer att presenteras i nästa kapitel.



Figur 1: Bias-varians tradeoff (Hämtad från (Hastie m. fl., 2009))

2.4.2 Korsvalidering

Konceptet bakom korsvalidering är att dela upp datamängden i två delar: en del för att anpassa modellen, träningsdata, och en annan del för att utvärdera modellens prestation genom att prediktera observationer från den andra delen, valideringsdata eller testdata (Johansson & Ohlsson, 2022). Men ett generellt problem vid prediktion är överanpassning, vilket innebär att modellen är så väl anpassad till träningsdatan att den ger dåliga prediktioner på valideringsdatan. För att undvika överanpassning är det viktigt att välja en modell baserat på resultaten från prediktionen på valideringsdatan.

Metoden för K-delad korsvalidering delar upp datan i K slumpmässiga grupper. Varje grupp av K fungerar sedan som valideringsdata medan de resterande grupperna tillsammans utgör träningsdata. Genom att beräkna medelvärdet av resultaten som genereras från en förlustfunktion över alla dessa steg kan ekvationen erhållas:

$$CV = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(f^k). \tag{18}$$

Notera att i ekvation (18), representerar $\mathcal{L}(f^k)$ den totala förlusten för grupp k när f estimeras på datamängden där grupp k är utesluten, det vill säga de återstående grupperna (Johansson & Ohlsson, 2022).

2.4.3 Black-box modell

Ett enkelt beslutsträd är väldigt tolkningsbart, en anledning är att hela modellen kan representeras av en enkel tvådimensionell grafik som sedan kan visualiseras. Dock förlorar linjära kombinationer av träd denna viktiga egenskap, och därav måste de tolkas genom andra metoder (Hastie m. fl., 2009).

2.4.3.1 Variable Importance

Ett sätt att analysera en GBM, är med hjälp av variabel importance, vilket kan översättas till svenska som "variabelbetydelse". Denna metod används för att bedöma vilka kovariat som är mest betydelsefulla för att göra en korrekt prediktion eller klassificering. Genom att analysera detta ökar förståelsen kring vilka faktorer som har störst påverkan på en modells resultat, och därmed kan man förbättra modellens prestanda genom att fokusera på de mest relevanta kovariat.

För ett enskilt träd f_m som är anpassat vid iteration m och en specifik, men godtycklig kovariat x_j, låt $S_m(x_j)$ vara kollektionen av splittar som är gjorda på x_j vid anpassningen av trädet. En viss split t resulterar i en minskning av den totala förlusten $\mathcal{L}$, vilket betecknas $(\triangle\mathcal{L})_{mt}$ (Johansson & Ohlsson, 2022). Den totala mängden x_j som bidrar till att minska den totala förlusten är följande (Johansson & Ohlsson, 2022):

$$\sum_{t \in S_m(x_j)} (\triangle\mathcal{L})_{mt}.$$

Det ska poängteras att om $S_m(x_j)$ är tom, så är summan definierad som 0.
I det slutliga trädet som har generats, x_j har då bidragit till den totala minskningen enligt

$$\mathcal{I}(x_j) = \sum_{m=1}^{M} \sum_{t \in S_m(x_j)} (\triangle\mathcal{L})_{mt}. \tag{19}$$

I (19) representerar M det totala antalet iterationer. Det går även att normalisera resultatet i (19) genom att skala $\mathcal{I}(x_j)$ så att det största värdet är 100. Det är även detta mått som refereras till gain eller "vinst" och ges av $100 \cdot \mathcal{I}(x_j)/(\max_{j'} \mathcal{I}(x_{j'}))$ (Johansson & Ohlsson, 2022).

2.4.3.2 Partial dependence plots (PDP)

Partial dependence plots ("delvis bereonde plots") brukar vanligtvis förkortas till PDP:er, är en grafisk representation av den förväntade påverkan av en specifik kovariat på modellens prediktion, med hänsyn till alla andra kovariat i modellen. Denna metod visar hur förändringar för den specifika kovariaten påverkar modellens prediktionsförmåga, samtidigt som de resterande kovariaten hålls konstanta.
Låt x_j vara den kovariat som är av intresse, och låt $\mathbf{x_i}^j(z)$ vara den observerade vektorn $\mathbf{x_i}$ förutom att observationen x_{ij} har ersatt av värdet z, samt behåller resterande kovariat konstanta. Låt $f(\mathbf{x}) = f_M(\mathbf{x})$ vara det slutliga trädet. Utifrån detta beräknas sedan medelvärdet över alla observationer för ett valfritt x_j (Johansson & Ohlsson, 2022)

$$\bar{f}_j(x_j) = \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x_i}^j(x_j)). \tag{20}$$

I (20) är värdet på x_j det valda värdet som ersätter det observerade värdet samtidigt som de resterande hålls oförändrade. Det är viktigt att inse att denna summa är över alla observationer och inte enbart de som har det valda värdet x_j. Notera att resultatet inte ger en beskrivning av populationen utan av själva trädet (Johansson & Ohlsson, 2022). Det vill säga vad är prediktionen av skadefrekvensen för just detta värde på x_j, i genomsnitt över alla observerade värden för de resterande kovariaten.
I de fall x_j förekommer multiplikativt i $f(\mathbf{x})$, utan interaktioner, möjliggör följande omskrivning $f(\mathbf{x_i}) = f_j(x_j) \times f_{-j}(\mathbf{x}_i^{(j)})$. Detta leder sedan trivialt till följande:

$$\bar{f}_j(x_j) = f_j(x_j) \frac{1}{n} \sum_{i=1}^{n} f_{-j}(\mathbf{x}_i^{(j)}). \tag{21}$$

En graf av $\bar{f}_j(x_j)$ kommer visualiserar formen på beroendet av y på x_j, i detta fall upp till en mulitplikativ konstant (Johansson & Ohlsson, 2022). Det går även att normalisera detta genom att definiera något värde x_j^* av x_j som en basklass.

2.4.4 GLM vs Black-box modell

Att kunna mäta modeller på deras prediktionsförmåga skapar en möjlighet för jämförelse. Det finns olika mått och olika metoder för att skapa denna möjlighet och detta kapitel kommer ett urval presenteras.

2.4.4.1 MSE

Det vanligaste måttet som används för att bedöma en modells prestanda är det så kallade medelkvadratfelet (MSE - "mean squared error"). Enligt (James m. fl., 2013) beräknas detta mått på följande sätt:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2, \tag{22}$$

här representerar $\hat{f}(x_i)$ prediktionen som ges av $\hat{f}$ för observation i. Om prediktionen är mycket nära det verkliga värdet kommer det att generera en lågt MSE-värde, medan en prediktion som avviker långt från det verkliga värdet kommer att resultera i ett högt MSE-värde.

2.4.4.2 Devians

Genom att låta $l(\hat{\boldsymbol{\mu}})$ beteckna log-likelihooden från den givna datamängden som en funktion av den estimerade medelvektorn $\hat{\boldsymbol{\mu}}$. Om antalet av icke-redundanta parametrar r, är samma som antalet observationer n, erhålls en perfekt anpassning genom att sätta alla $\hat{\mu}_i = y_i$. Detta är även ML-skattningarna, eftersom de trivialt uppfyller ML-ekvationerna som har beskrivits i tidigare kapitel. Fallet när $r = n$ benämns som den mättade modellen, det är även denna modell som används som riktmärke vid mätning av godness-of-fit för andra modeller. Detta eftersom den mättade modellen har en perfekt anpassning. Denna mätning är även kallad skalad devians D^*, som är definierad som en LRT-statistika för den aktuella modellen mot den mättade modellen (Ohlsson & Johansson, 2010). Denna statistika är definierad som följande:

$$D^* = D^*(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = 2[l(\boldsymbol{y}) - l(\hat{\boldsymbol{\mu}})]. \tag{23}$$

I (23) antas det att samma ϕ används i $l(\boldsymbol{y})$ och $l(\hat{\boldsymbol{\mu}})$. Låt h beteckna inversen av b' i relationen $\mu_i = b'(\theta_i)$ så att $\theta_i = h(\mu_i)$. Detta leder i sin tur att D^* kan uttryckas genom följande uttryck (Ohlsson & Johansson, 2010):

$$D^* = \frac{2}{\phi}\sum_i w_i(y_i h(y_i) - b(h(y_i)) - y_i h(\hat{\mu}_i) + b(h(\hat{\mu}_i))). \tag{24}$$

Genom att nu multiplicera ϕ med uttrycket i (24) ges den oskalade deviansen $D = \phi D^*$, vilket ger fördelen att inte bero på ϕ. Det finns olika devianser och de vanligaste kommer från normal, Poisson och gamma fördelningar, respektive devians ges nedanför (Ohlsson & Johansson, 2010):

$$\begin{aligned} &D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = \textstyle\sum_i w_i(y_i - \hat{\mu}_i)^2, \\ &D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = 2\textstyle\sum_i w_i(y_i \log y_i - y \log \hat{\mu}_i - y_i + \hat{\mu}_i), \\ &D(\boldsymbol{y}, \hat{\boldsymbol{\mu}}) = 2\textstyle\sum_i w_i(y/\hat{\mu}_i - 1 - \log(y_i/\hat{\mu}_i)). \end{aligned} \tag{25}$$

2.5 Surrogatmodell

Det är ofta svårt och komplicerat att tolka en så kallad "black-box"-modell, och det finns många olika metoder som syftar till att göra detta möjligt, till exempel LIME (Maillart, 2021). Dock ger dessa metoder oftast bara en lokal bild (Maillart, 2021). I artikeln (Maillart, 2021) presenteras DefragTrees, en metod för att extrahera en surrogatmodell från en stor klass av black-box-modeller. Denna metod bygger på en Bayesiansk modellselektionsstrategi (Hara & Hayashi, 2018) där man först anpassar additiva träd och sedan anpassar en GLM (Maillart, 2021). En liknande metod är att istället anpassa ett klassiskt regressionsträd med ett eventuellt förutbestämt antal löv. Detta skulle resultera i en mer tolkningsbar modell, vilket är målet med surrogatmodellen. På så sätt kan man utifrån black-box-modellens prediktioner anpassa ett vanligt beslutsträd och sedan välja ut regionerna i trädet för att göra modellen tolkningsbar.

2.5.1 Fidelity

För att kunna jämföra surrogatmodellen med black-box modellen introduceras anpassningsmåttet fidelity. Detta mått kan även ses som R^2-måttet mellan prediktionen av black-box modellen och prediktionen för surrogatmodellen (Maillart, 2021). Måttet definieras enligt följande:

$$\text{fidelity} = 1 - \frac{\sum_{i=1}^{n}(f_{bb}(\boldsymbol{x_i}) - f_{sur}(\boldsymbol{x_i}))^2}{\sum_{i=1}^{n}(f_{bb}(\boldsymbol{x_i}) - \bar{y}_{bb})^2} \tag{26}$$

I (26) representerar f_{bb} och f_{sur} black-box modellen respektive surrogatmodellen, och $\bar{y}_{bb}$ medelvärdet för prediktionerna i black-box modellen.

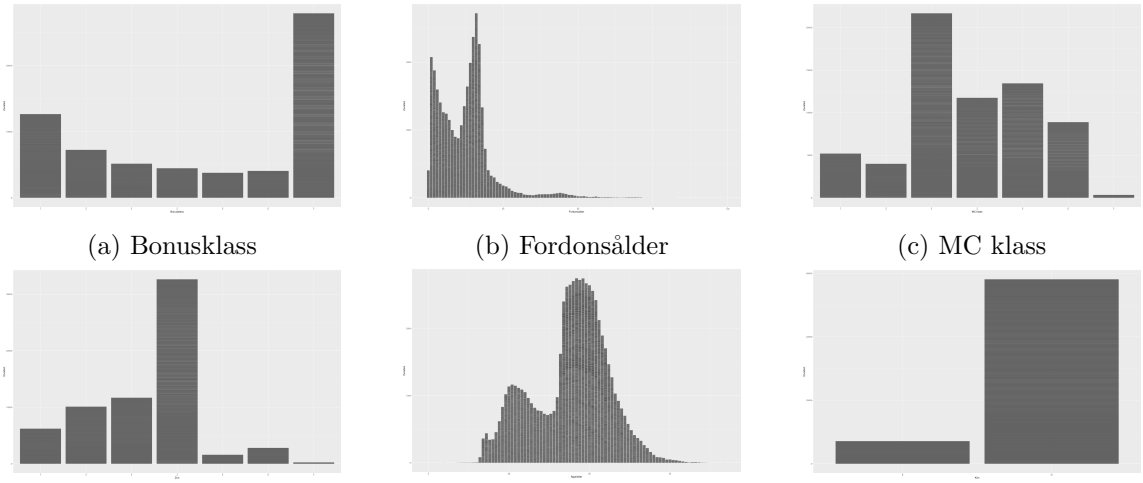
3 Data

Den datamängd som kommer att användas för analysen är "MCcase.txt" som kan hämtas från boken (Ohlsson & Johansson, 2010). I denna datamängd finns sex olika kovariat: Agarald, som bestämmer ägarens ålder och är en kontinuerlig variabel mellan 0 och 99 år, Kon som bestämmer ägarens kön och är en binär variabel som antar värdet M eller K (1 eller 0), MCklass som anger MC-klassen baserat på en EV-ratio, vilket är en kategorisk variabel. Det finns även en variabel som beskriver fordonsåldern, Fordald, som är en kontinuerlig variabel mellan 0 och 99 år. De två sista kovariaterna som kan användas vid analysen är Bonskl, vilket anger ägarens bonusklass baserat på antalet skadefria år, samt Zon som anger den geografiska zonen där ägaren bor. Dessa två kovariater är kategoriska variabler. En sammanfattning av alla kovariater som används i analysen finns i Tabell 2.

Variabel	Beskrivning	Antal nivåer
Agarald	Ägarens ålder, 0-99	100
Kon	Ägarens kön, M (man) och K (kvinna)	2
Zon	Geografiska zoner	7
Mcklass	MC klass baserat på EV ratio	7
Fordald	Fordonsålder, 0-99	100
Bonuskl	Bonusklass, baserat på skadefria år	7

Tabell 2: MCcase data

Genom att studera Figur 2 framgår det att vissa kovariater har klasser med mycket låga durationer. Detta leder till att vissa klasser måste grupperas ihop för att undvika risken att de innehåller för lite data. Dessutom kan vi analysera att vissa kovariater har 100 klasser, och även där finns de med låga durationer. Det är inte lämpligt att använda 100 klasser för varje kontinuerlig kovariat. Dessa variabler måste därför delas in i så kallade tariffceller för att eliminera problemet med för många klasser.



(a) Bonusklass (b) Fordonsålder (c) MC klass

(d) Zon (e) Ägarålder (f) Kön K/M

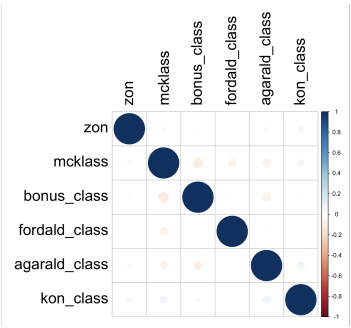
Figur 2: Kovariat mot duration

I boken (Ohlsson & Johansson, 2010) framgår en färdig tariffuppdelning. Denna uppdelning kommer även att användas i denna analys, vilket motsvarar uppdelningen som visas nedanför i Tabell 3. Notera att denna tariffuppdelning enbart kommer att användas vid anpassning av en GLM, eftersom för GBM och random forest krävs inte denna färdiga uppdelning.

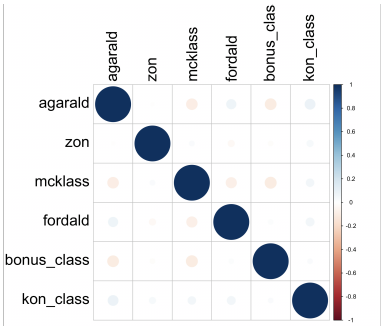
Variabel	Klass	Beskrivning
Zon	1	Centrala och sem-centrala delar av Sveriges största städer
	2	Förorter plus medelstora städer
	3	Mindre orter, förutom de i 5 eller 7
	4	Småorter och landsbygden, förutom de i 5 eller 7
	5	Nordliga orter
	6	Nordlig landsbygd
	7	Gotland
Mcklass	1	EV ratio -5
	2	EV ratio 6-8
	3	EV ratio 9-12
	4	EV ratio 13-15
	5	EV ratio 16-19
	6	EV ratio 20-24
	7	EV ratio 25-
Fordald	1	0-1 år
	2	2-4 år
	3	5- år
Bonuskl	1	1-2
	2	3-4
	3	5-7

Tabell 3: Tariffuppdelning

Förutom det som visas i Tabell 3, delas även Ägarålder upp i klasser. Denna uppdelning sker genom att gruppera så att alla som är under 20 år blir en grupp och alla som är över 70 år blir en annan, medan de som är däremellan delas upp i 5-årsintervaller. Notera att denna uppdelning har tagits fram genom analys av hela datasetet. Därför har tränings- och valideringsset ännu inte tagits fram.
Eftersom det eventuellt kan finnas korrelation mellan kovariaterna är detta en viktig aspekt att undersöka. Genom Figur 3 går det att analysera att oavsett om fordonsålder och ägarålder behandlas kontinuerligt eller är indelade i klasser, så finns det inga tydliga tecken på korrelation mellan kovariaterna. Detta leder till att inga samspel kommer inkluderas vid modellanpassningarna för GLM.



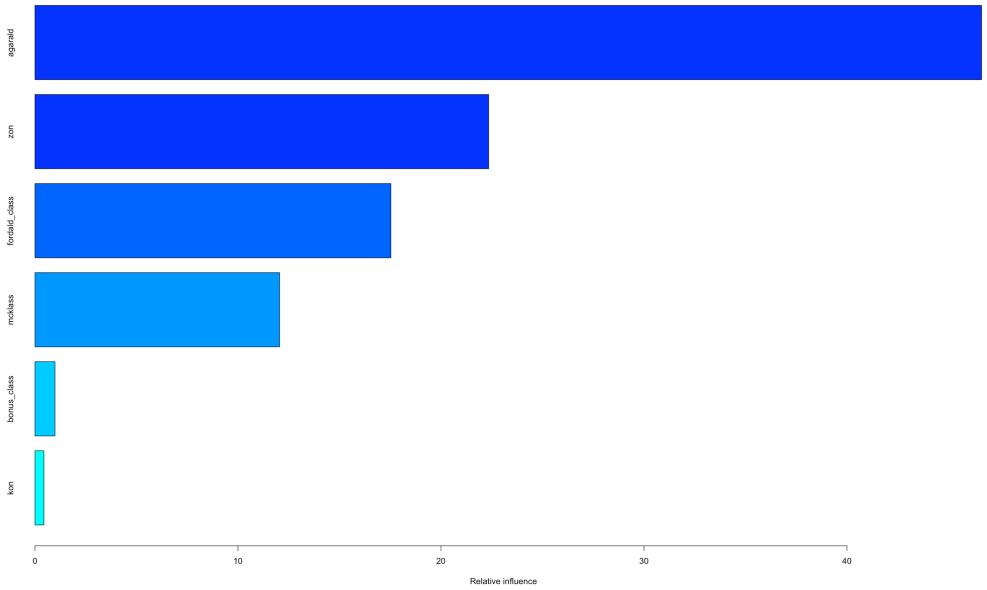
(a) Fordonsålder och Ägarålder indelade i klasser



(b) Fordonsålder och Ägarålder kontinuerlig

Figur 3: Korrelationsmatris

Att anpassa en modell med hjälp av GBM kan även vara till nytta för att få en överblick om vilka kovariat som eventuellt kan tas bort. Nedanför i Figur 4 visas det att kovariaten kön och bonusklass har en svag betydelse vid modellanpassning med hjälp av GBM. Detta leder till att det kommer undersökas om ifall borttagning av dessa kovariat saknar betydelse för modellanpassningen i senare avsnitt. Notera att detta enbart gäller för modeller anpassade med hjälp av GLM.



Figur 4: Variablebetydelse (VIP) för en GBM anpassat på hela datamängden med samtliga kovariater

4 Modellanpassning

I detta avsnitt kommer ett flertal modeller att anpassas och analyseras men olika metoder. Detta kommer genomföras för att skapa en prediktionsmodell för skadefrekvensen, dock ska det poängteras att det även är möjligt att utföra samma process för medelskadekostnaden. Notera även att data kommer delas upp i träningsdata (80%) och valideringsdata (20%). Dessa två datamängder kommer att användas för samtliga modellanpassningar.

4.1 GLM

Vid framtagning av GLM kommer både den färdiga tariffindelningen att användas, där en komplettering görs genom att dela upp Ägaråldern, samt modeller som använder sig av splines för att beskriva de kontinuerliga kovariaterna. Observera att tidigare nämnda modeller även kommer att anpassas genom att kön och bonusklass tas bort som kovariat. Nedan i Tabell 4 visas de olika modellstrukturerna som kommer att användas i GLM-avsnittet.

Modell	Zon	Kön	MC klass	Bonusklass	Ägarålder (klass)	Fordonsålder (Klass)	Ägarålder (Splines)	Fordonsålder (Splines)
Modell 1	X	X	X	X	X	X		
Modell 2	X		X	X	X	X		
Modell 3	X		X		X	X		
Modell 4	X		X	X		X	X	
Modell 5	X		X	X	X			X
Modell 6	X		X	X			X	X

Tabell 4: Modellstrukturer

4.1.1 Skadefrekvens

Genom att anpassa varje modell på träningsdata och sedan göra en prediktion på valideringsdata kan den bästa modellen identifieras. I Tabell 5 presenteras MSE och residual devians för var och en av de sex anpassade GLM-modellerna. Det visar sig att Modell 6 har lägst MSE för valideringsdata och även lägst residual devians för träningsdata. Dessa resultat indikerar att Modell 6 presterar bättre än de andra modellerna. För att ytterligare verifiera detta genomfördes ett likelihood-kvottest där modellerna jämfördes med varandra baserat på träningsdata. Resultaten av detta test visas i Tabell 6, där jämförelsen görs med residual devians som teststatistik. Tabellen visar att det finns en signifikant skillnad i residual devians mellan Modell 3 och Modell 4 samt mellan Modell 3 och de övriga modellerna. Det visar sig även att det finns en signifikant förbättring att använda Modell 6 jämfört med Modell 4, vilket stödjer slutsatsen att Modell 6 är den bäst anpassade modellen på träningsdata.

Modell	MSE - träningsdata	MSE - valideringsdata	Residual devians - träningsdata
Modell 1	0.01146913	0.01225482	3995.2
Modell 2	0.01147156	0.01225527	3999.0
Modell 3	0.01146707	0.01225465	4004.1
Modell 4	0.01146969	0.01224	3994.3
Modell 5	0.01145224	0.01221741	3931.5
Modell 6	0.01145255	0.01221133	3928.2

Tabell 5: MSE och devians för respektive modell - GLM

Jämförelse av modell	Skillnad i devians	Pr(Chi)
Modell 1 vs Modell 2	-3.7313	0.1372
Modell 2 vs Modell 3	-5.1319	0.2178
Modell 3 vs Modell 4	9.7739	0.01577
Modell 3 vs Modell 5	72.551	1.904e-06
Modell 3 vs Modell 6	75.933	2.82e-07
Modell 4 vs Modell 5	62.777	1.054e-05
Modell 4 vs Modell 6	66.159	1.595e-06

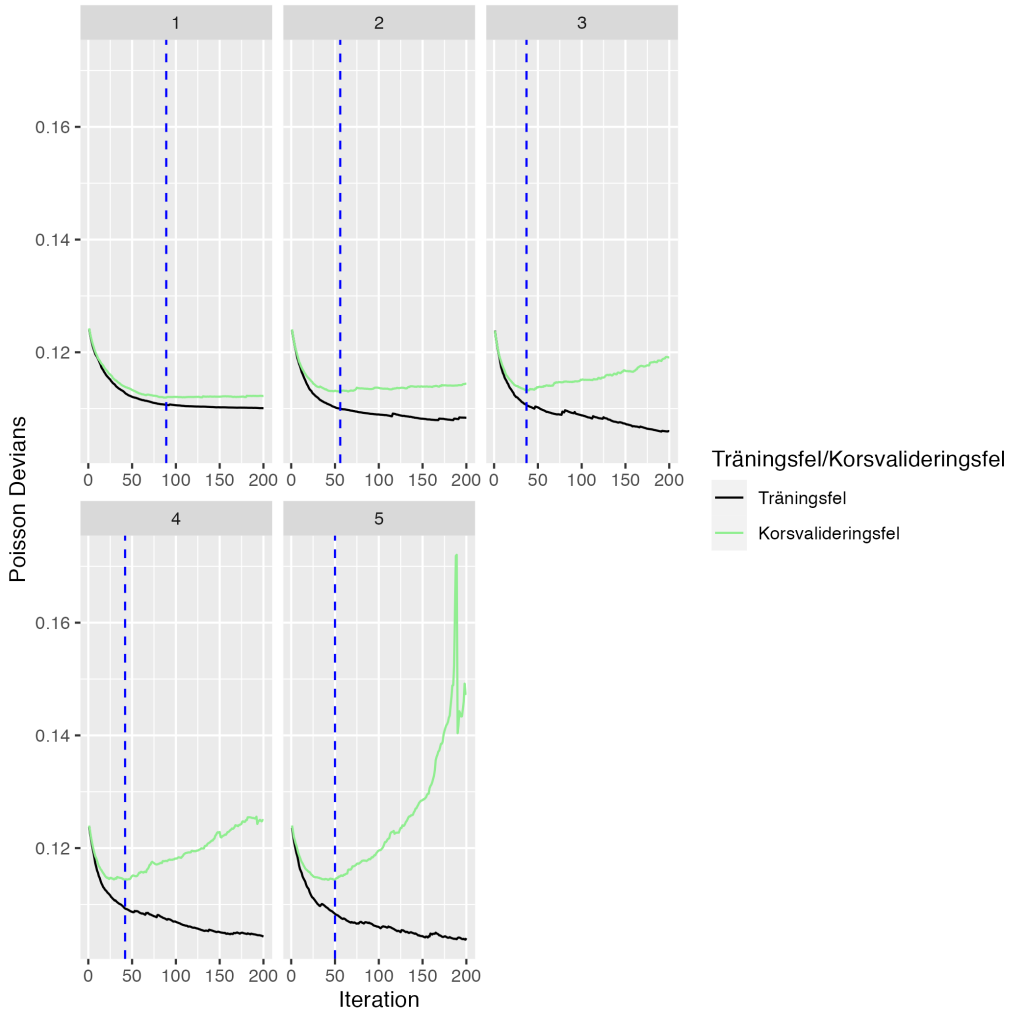
Tabell 6: LRT - GLM

4.2 GBM

Genom att specificera olika kombinationer av hyperparametrarna hos GBM kan dessa optimeras för att anpassa modellen på bästa sätt. De hyperparametrar som kommer att optimeras är "learning.rate" (inlärningshastighet), "n.trees" (antal träd) och "interaction.depth" (trädets djup). De olika värden som kommer att användas är n.trees = 200, 300, 600 eller 1300 beroende på vilken learning rate som används. Antalet träd som ska anpassas väljs sedan med hjälp av korsvalidering. De olika learning rates som används är 0.01, 0.05, 0.1 (standard) och 0.5. Den sista hyperparametern som måste optimeras är interaction.depth, vilket anger hur djupt varje träd ska vara vid varje split. I denna rapport kommer vi att använda värdena 1, 2, 3, 4 och 5 för interaction.depth.

4.2.1 Skadefrekvens

I Figur 5 presenteras den kombination av parametrar som resulterade i lägst korsvalideringsfel för GBM-modellen. Den valda kombinationen består av en learning rate på 0.1 (defualt), en interaction depth på 1 och 200 träd (n.trees). Vid en interaction depth på 1 erhölls alltid det lägsta korsvalideringsfelet oavsett val av learning rate (se Figur 19-21 i appendix). Resultaten av modellerna visas i Tabell 7, vilket indikerar att modellen med interaction depth = 1 och learning rate = 0.1 genererade det lägsta MSE-värdet för både tränings- och valideringsdata. Detta resultat tyder på att modellen har en god anpassning till datan och att det inte finns någon överanpassning.



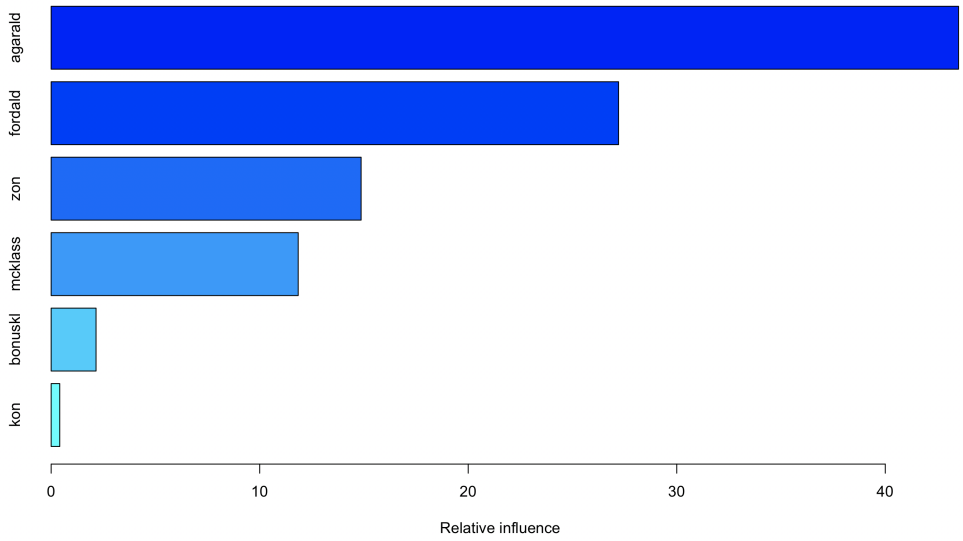
Figur 5: GBM för olika interaction.depth vid n.trees = 200 och learning rate = 0.1

Modell	MSE - träningsdata	MSE - valideringsdata
GBM - learning rate 0.01	0.01160934	0.01232992
GBM - learning rate 0.05	0.01159745	0.01230684
GBM - learning rate 0.1	0.01154601	0.01224391
GBM - learning rate 0.5	0.0116941	0.01247946

Tabell 7: MSE för de olika GBM modellerna, interaction.depth = 1

Baserat på Variabel Importance Ploten (VIP), Figur 6, så kan vi se att ägarålder är den

viktigaste variabeln för vår GBM-modell, följt av fordonsålder och zon. Mcklass, bonusklass och kön anses vara mindre viktiga variabler. Detta innebär att vår modell anser att ägarålder, fordonsålder och zon är de mest signifikanta faktorerna när det kommer till att prediktera skadefrekvensen. Detta kan ge oss värdefull insikt i vilka faktorer som har störst inverkan på skadefrekvensen och kan hjälpa oss att fokusera på att förbättra insamlingen av data för dessa variabler för att förbättra modellens prestanda.



Figur 6: Variable importance GBM, vid interaction.depth = 1, n.trees = 200 och learning rate = 0.1

4.2.1.1 Surrogatmodell

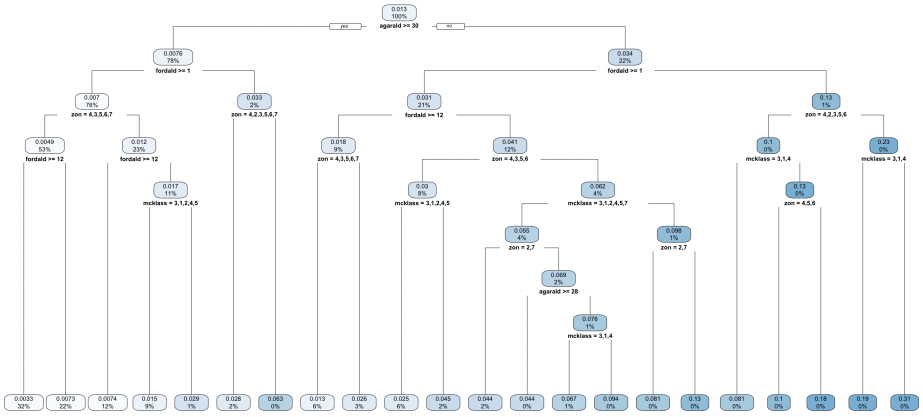
Resultaten som presenteras i Tabell 8 visar att surrogatmodellen, som utvecklades genom att anpassa ett enkelt beslutsträd på GBM modellens prediktionsvärden, har en lägre MSE för valideringsdata än den ursprungliga GBM modellen. Detta innebär att surrogatmodellen är mer tillförlitlig vid förutsägelse av nya observationer som inte ingick i träningsdata. Vidare observerades att surrogatmodellen presterade bäst när komplexitetsparametern (cp) sattes till 0.005. Detta innebär att man bör välja en sådan parameterinställning för att få bästa möjliga prediktionsförmåga för surrogatmodellen.
Notera att surrogatmodellen anpassas genom att använda komplexitetsparametern cp. Detta innebär att varje split måste förbättra resultatet med mer än faktorn cp för att inkluderas. I det här fallet måste förklaringsgraden R^2 förbättras med cp vid varje split. Detta är också en hyperparameter som optimeras genom korsvalidering, vilket tidigare nämnts. Ett

annat alternativ skulle vara att använda “maxdepth”, vilket anger det maximala djupet för det slutliga trädet.
En annan viktig observation är att surrogatmodellen med näst lägst MSE, är även den med bäst fidelity. Detta innebär att den modellen är den bästa på att återge prediktionsvärdena från GBM prediktionerna. Detta beror på att denna surrogatmodell bygger på en mer komplex version och kan därför representera alla de komplexa sambanden mellan variablerna på samma bättre sätt än vad en mer enkel trädmodell kan. Trots detta kan det ändå vara smart att använda sig av surrogatmodellen som genererade den lägsta MSE, och näst lägst fidelity (cp = 0.005) eftersom målet var att öka tolkningsbarheten av GBM modellen. Modellen med bäst fidelity genererade ett mycket stort beslutsträd (se Figur 22 i appendix), vilket ofta kan bli svårtolkat och är därför motsatsen till vad man vill åstadkomma.

Modell	MSE - träningsdata	MSE - valideringsdata	Fidelity - träningsdata	Fidelity - valideringsdata
GBM	0.01154601	0.01224391	1	1
Surrogat - cp = 0.001	0.0115587	0.01221733	0.9319946	0.9086132
Surrogat - cp = 0.005	0.01157873	0.01221491	0.8506386	0.8261026
Surrogat - cp = 0.01	0.01157467	0.01224352	0.7855251	0.770012
Surrogat - cp = 0.015	0.01159002	0.01224849	0.7470471	0.7309986
Surrogat - cp = 0.025	0.01159503	0.01225103	0.7093737	0.6944308
Surrogat - cp = 0.045	0.01158574	0.01227831	0.6043224	0.5893936
Surrogat - cp = 0.055	0.01158574	0.01227831	0.6043224	0.5893936
Surrogat - cp = 0.070	0.01158574	0.01227831	0.6043224	0.5893936
Surrogat - cp = 0.075	0.01160164	0.01230085	0.5294259	0.5216902
Surrogat - cp = 0.1	0.01162683	0.01230214	0.4509997	0.4492558

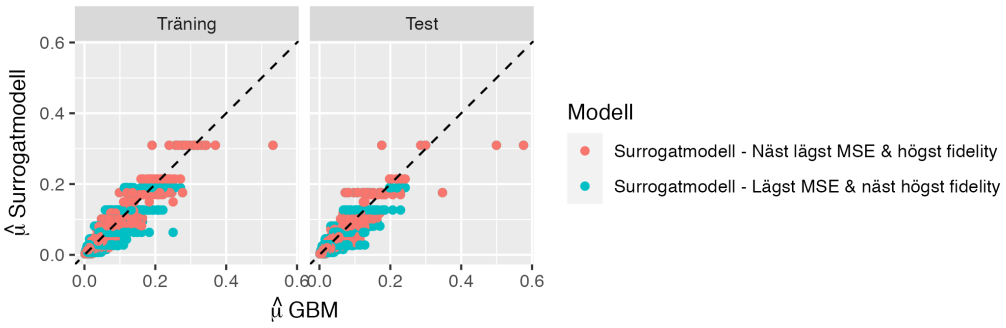
Tabell 8: MSE och fidelity för respektive surrogatmodell - GBM

I Figur 11 kan man se den surrogatmodell som har genererats och det framgår att det beslutsträd som har 22 löv/terminalnoder ger lägst MSE. Dessutom kan man analysera att två terminalnoder innehåller över 50% av alla observationer medan de andra terminalnoderna innehåller mellan 1-9% eller avrundats till 0%.



Figur 7: Surrogatmodell för GBM, cp = 0.005

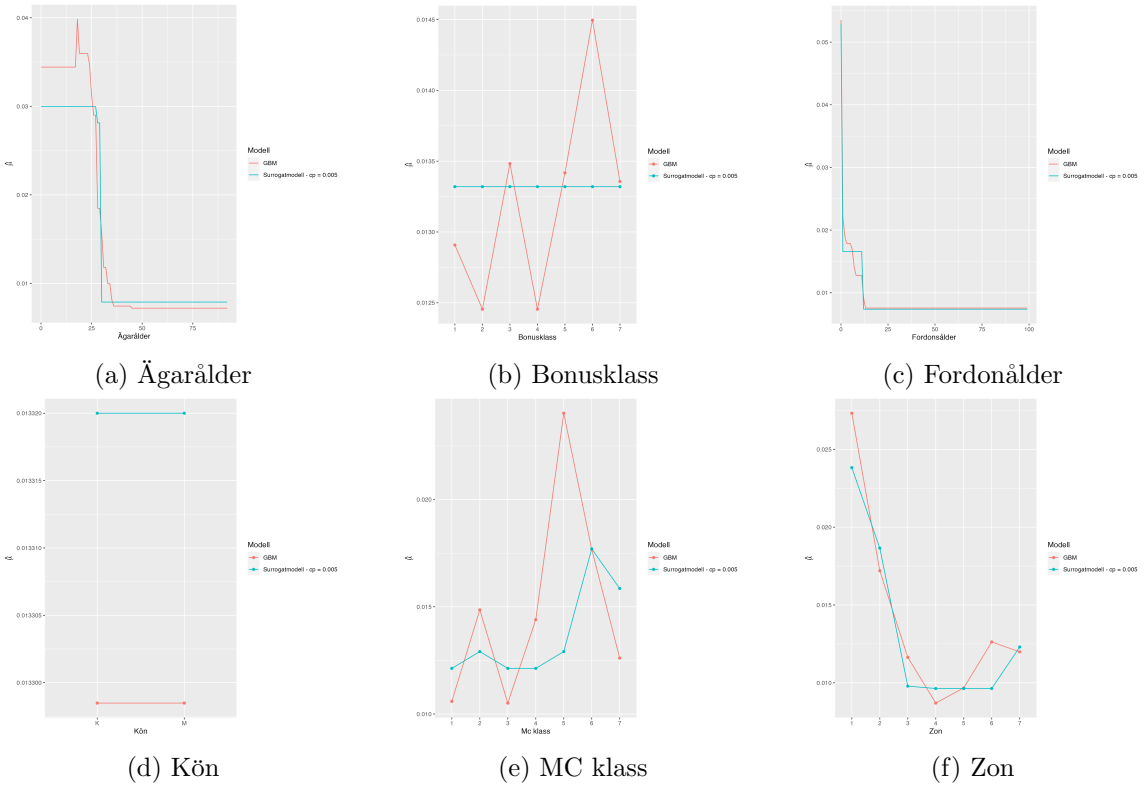
Visualiseringen i figur 8 visar prediktionerna från GBM och de två olika surrogatmodeller - en med lägst MSE och högst fidelity samt den med näst lägst MSE och näst högst fidelity. Både surrogatmodellerna följer i stort sett GBM-prediktionerna, men med några avvikelser. Detta tyder på att båda surrogatmodellerna är användbara för att beskriva GBM på ett bra sätt.



Figur 8: Surrogatmodellers prediktioner mot GBM prediktioner

En annan analys som visar att den givna surrogatmodellen uppvisar samma beteende är Partial dependence plottarna, se Figur 9. De kontinuerliga kovariaterna ägarålder och fordonsålder genererar väldigt liknande effekter för respektive modell. Det finns några få enstaka avvikelser, men överlag följer modellerna varandra mycket nära. Mc-klass och zon uppvisar dock några större avvikelser från varandra, men visar ändå samma trender över klasserna. Den största skillnaden finns i bonusklass, där surrogatmodellen har en konstant trend medan GBM har en mer varierande. Det finns även en stor skillnad i effekt för kön, men det ska dock noteras att denna variabel inte används i surrogatmodellen, vilket tydligt visas i Figur 10. Denna figur visar även att surrogatmodellen rangordnar kovariaternas

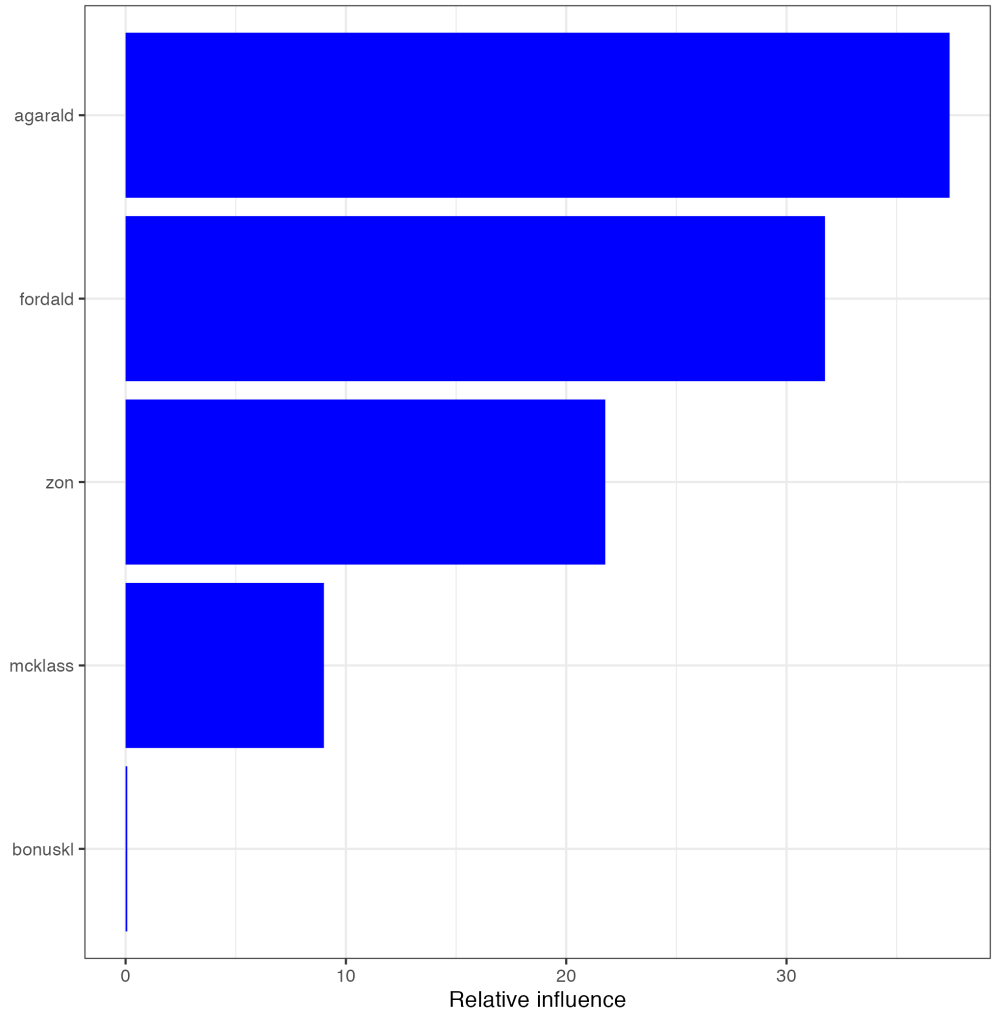
inverkan på samma sätt som GBM. Den största inverkan kommer från ägarålder, följt av fordonsålder, zon, mc-klass och bonusklass. Det som skiljer sig är att surrogatmodellen inte inkluderar kön som en variabel. Med andra ord har den aldrig använt kön vid en uppdelning, vilket leder till att detta kovariat inte visas i figuren.



(a) Ägarålder (b) Bonusklass (c) Fordonålder

(d) Kön (e) MC klass (f) Zon

Figur 9: Partial dependence plots för respektive kovariat och modell (Röd = GBM, BLÅ = Surrogatmodell)



Figur 10: Variable Importance (VIP), surrogatmodell cp = 0.005

4.3 Random forest

För att kunna anpassa en modell med hjälp av random forest, krävs det att optimera olika hyperparametrar. I detta fall handlar det om antalet variabler som slumpmässigt väljs ut som kandidater för varje split (mtry), men även antalet träd som ska modelleras (ntree). I denna rapport kommer mtry vara 1, 2 eller 3 och ntree kommer anta värdena 500 och 1000.

4.3.1 Skadefrekvens

Tabell 9 visar resultatet av en analys av flera olika Random Forest-modeller. Genom att undersöka hur bra modellerna passade träningsdata och valideringsdata genom att beräkna MSE för varje modell. Modellen med lägst MSE för valideringsdata var den som använde hyperparametrarna mtry = 1 och ntree = 1000, vilket betyder att den använde endast en slumpmässigt vald kovariat vid varje nodsplit och byggde 1000 träd. Detta resultat indikerar att modellen med en slumpmässigt vald variabel vid varje nodsplit är en bättre modell för att prediktera data än modeller med fler slumpmässigt valda kovariat. Denna modell kommer att användas för att extrahera en surrogatmodell för att försöka förstå sambanden mellan kovariaten och deras prediktion.

Modell	MSE - träningsdata	MSE - valideringsdata
Random forest - mtry = 2, ntree = 500	0.007099653	0.01248436
Random forest - mtry = 1, ntree = 1000	0.01078074	0.01218043
Random forest - mtry = 2, ntree = 1000	0.007109978	0.01247658
Random forest - mtry = 3, ntree = 1000	0.005331232	0.01283962

Tabell 9: MSE för respektive modell - Random forest

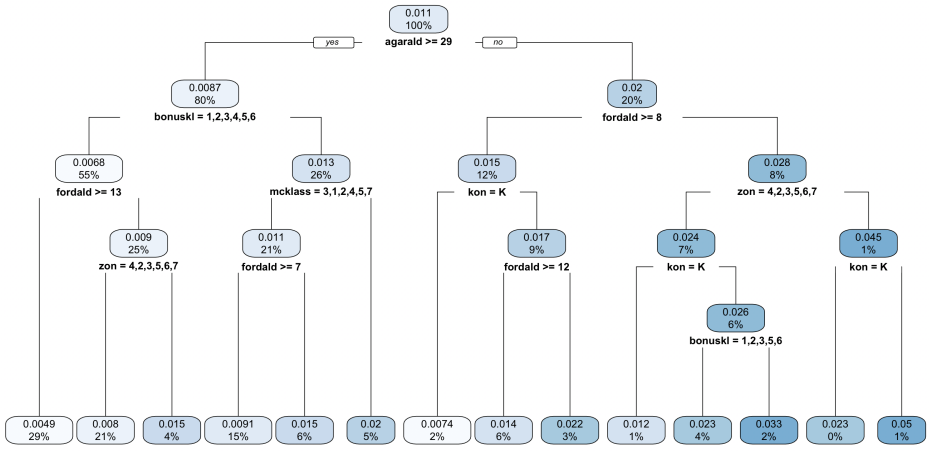
4.3.1.1 Surrogatmodell

Genom att analysera Tabell 10 som visar resultaten för olika surrogatmodeller som genererades från den ursprungliga Random forest-modellen som presenterades i Tabell 9. Surrogatmodellerna utvärderades baserat på MSE och fidelity (träningsdata och valideringdata) i förhållande till Random forest-modellen. Resultaten visar att den ursprungliga Random forest-modellen hade en MSE på 0.0108 för träningsdata och 0.01218 för valideringsdata. En surrogatmodell med cp = 0.01 hade det näst lägsta MSE på 0.011489 för träningsdata och 0.01222 för valideringsdata. Dessutom hade denna surrogatmodell också den näst högsta fidelity på 0.565 för träningsdata och 0.647 för valideringsdata.
En annan surrogatmodell med en lägre MSE och högre fidelity, men gav ett beslutsträd på 71 löv/terminalnoder (se Figur 23 i appendix) , vilket anses svårtolkat. Därför valdes den surrogatmodell som gav det näst lägsta MSE och näst högsta fidelity med cp = 0.01, som även genererade ett beslutsträd på 14 löv.

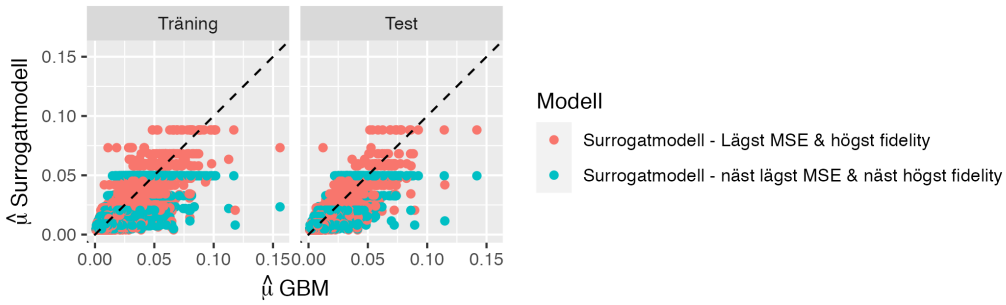
Modell	MSE - träningsdata	MSE - valideringsdata	Fidelity - träningsdata	Fidelity - valideringsdata
Random forest	0.01078074	0.01218043	1	1
Surrogat - cp = 0.001	0.01141039	0.01219113	0.7150293	0.7873216
Surrogat - cp = 0.01	0.01148831	0.01221891	0.5650117	0.6474758
Surrogat - cp = 0.015	0.01150254	0.01223372	0.5139197	0.5922014
Surrogat - cp = 0.025	0.01151483	0.01225193	0.4385194	0.5065831
Surrogat - cp = 0.045	0.01152411	0.01225733	0.4042857	0.4665041
Surrogat - cp = 0.055	0.0115461	0.0122696	0.3507799	0.4089609
Surrogat - cp = 0.070	0.01155951	0.01228112	0.2882284	0.33804
Surrogat - cp = 0.075	0.01155951	0.01228112	0.2882284	0.33804
Surrogat - cp = 0.1	0.01158291	0.0122889	0.2110066	0.2473833

Tabell 10: MSE och fidelity för respektive surrogatmodell - Random forest

I Figur 12 som presenteras visar att de två surrogatmodellernas prediktioner stämmer väl överens med Random Forest-modellens prediktioner. Detta bekräftar att surrogatmodellen kan användas som en tillförlitlig approximativ modell för att förenkla den ursprungliga modellen. Det är också intressant att notera att det finns två löv i beslutsträdet som har 50% av alla observationer, vilket visas i Figur 11. Detta kan vara användbart för att identifiera de mest betydande egenskaperna eller kovariaterna som påverkar modellen.

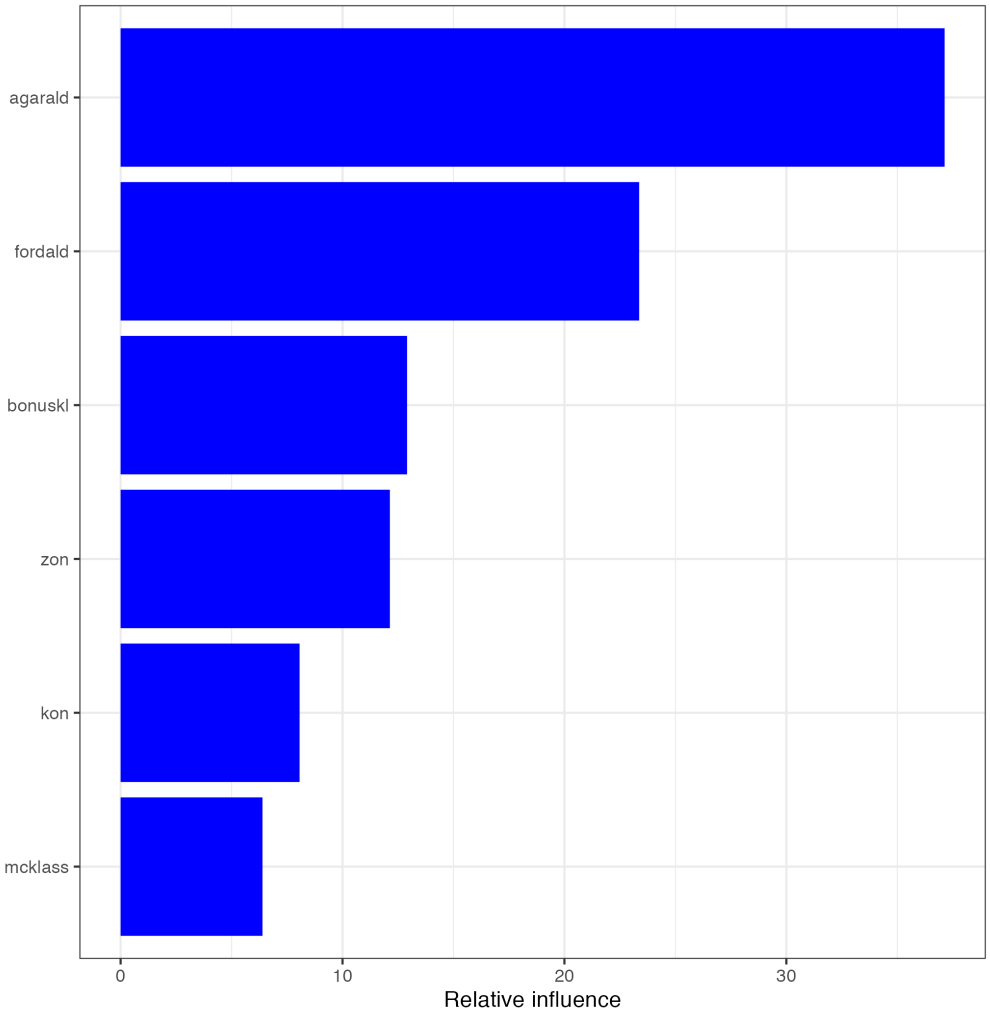


Figur 11: Surrogatmodell för random forest, cp = 0.01



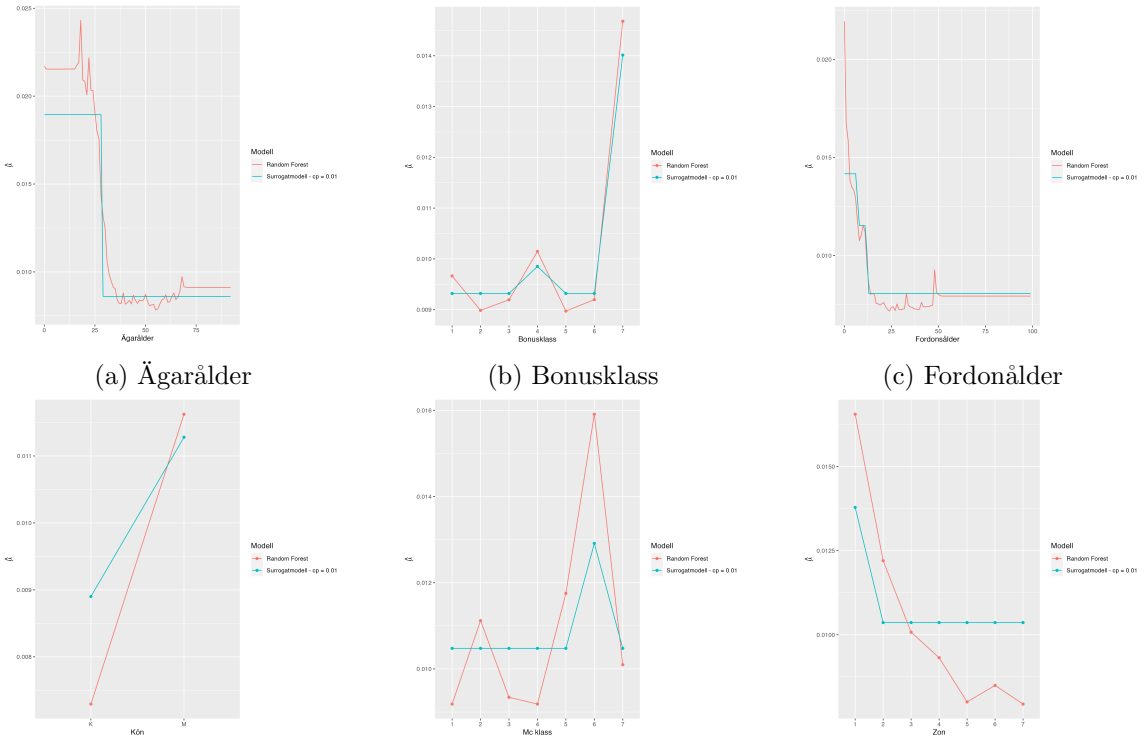
Figur 12: Surrogatmodellers prediktioner mot random forest prediktioner

Kovariaternas inverkan på responsvariabeln är något annorlunda jämfört med surrogatmodellen som togs fram för GBM. Genom att analysera Figur 13 visar det sig att ägarålder fortsätter att ha störst inverkan, följt av fordonsålder. Det som skiljer sig nu jämfört med tidigare är att bonusklass, som tidigare var en av de svagare kovariaterna, nu rankas som den tredje viktigaste. Zon har fortfarande en liknande inverkan som tidigare, men en skillnad är att mc-klass nu har lägre inverkan än kön. Notera att surrogatmodellen för Random Forest har inkluderat samtliga kovariater, medan den för GBM uteslöt kön.



Figur 13: Variable Importance (VIP), surrogatmodell cp = 0.01

Genom att använda Figur 14 kan sambandet mellan kovariaterna och responsvariabeln analyseras. Det framgår att den ursprungliga Random Forest-modellen är mer flexibel och varierar mer än surrogatmodellen. Det bör dock betonas att surrogatmodellen verkar följa alla trender från den ursprungliga modellen. Det som sticker ut mest är effekten i zon, där surrogatmodellen ger samma effekt för alla klasser utom "1", medan Random Forest-modellen ger en avtagande trend.



(a) Ägarålder (b) Bonusklass (c) Fordonålder

(d) Kön (e) MC klass (f) Zon

Figur 14: Partial dependence plots för respektive kovariat och modell (Röd = Random Forest, BLÅ = Surrogatmodell)

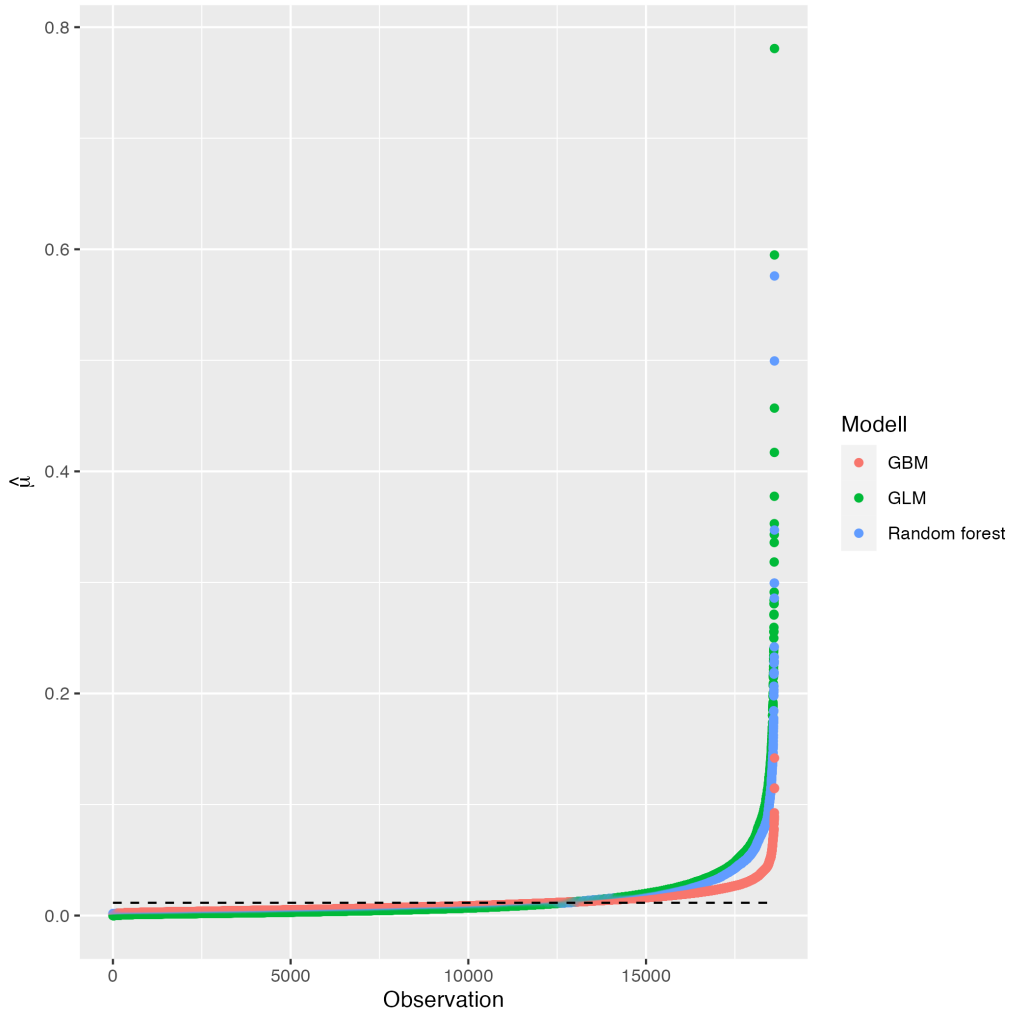
5 Resultat

Resultatet visar att användningen av splines för både ägarålder och fordonsålder genererade en bättre modell jämfört med att dela dem i klasser. En generaliserad linjär modell (GLM) med splines uppvisade näst bäst MSE för valideringsdata, medan Random Forest-modellen presterade ännu bättre med en lägre MSE än GLM. Den enda modellen som presterade sämre än GLM var GBM. Det är viktigt att notera att skillnaderna mellan de olika modellerna inte är särskilt stora, vilket också kan ses i både Tabell 11 och Figur 15. Figur 15, där prediktionerna visas i storleksordning, indikerar att alla tre modellerna följer varandra relativt väl. Det är först i slutet av fördelningen där de olika modellerna tenderar att ge något olika predikterade värden. GLM har prediktionerna med de största värdena, medan för GBM och Random forest ligger de inte på samma nivå.
Tabell 11 presenterar resultaten för de olika modellerna, där MSE är angivet för både träningsdata och valideringsdata. Dessutom finns fidelity-värden för träningsdata och valideringsdata för varje surrogatmodell. Baserat på dessa resultat kan man konstatera att

modellerna med splines för ägarålder och fordonsålder har en bättre prestanda än de övriga modellerna hos GLM. Bland alla modeller i tabellen uppvisar Random Forest det bästa MSE-resultatet för valideringsdata. Detta resultat visar även att surrogatmodellen för random forest med cp = 0.001 var den enda surrogatmodellen som uppvisade en lägre MSE än GLM - modell 6. Dock ska det poängteras att denna surrogatmodell hade över 70 slutnoder, medan surrogatmodellen med cp = 0.01 hade betydligt färre.

Modell	MSE - träningsdata	MSE - valideringsdata	Fidelity - träningsdata	Fidelity - valideringsdata
GLM - Modell 3	0.01146707	0.01225465	-	-
GLM - Modell 6	0.01145255	0.01221133	-	-
GBM	0.01154601	0.01224391	-	-
Random forest	0.005321006	0.01217751	-	-
Surrogat GBM - cp = 0.001	0.0115587	0.01221733	0.9319946	0.9086132
Surrogat GBM - cp = 0.005	0.01157873	0.01221491	0.8506386	0.8261026
Surrogat Random forest - cp = 0.001	0.01141039	0.01219113	0.7150293	0.7873216
Surrogat Random forest - cp = 0.01	0.01148831	0.01221891	0.5650117	0.6474758

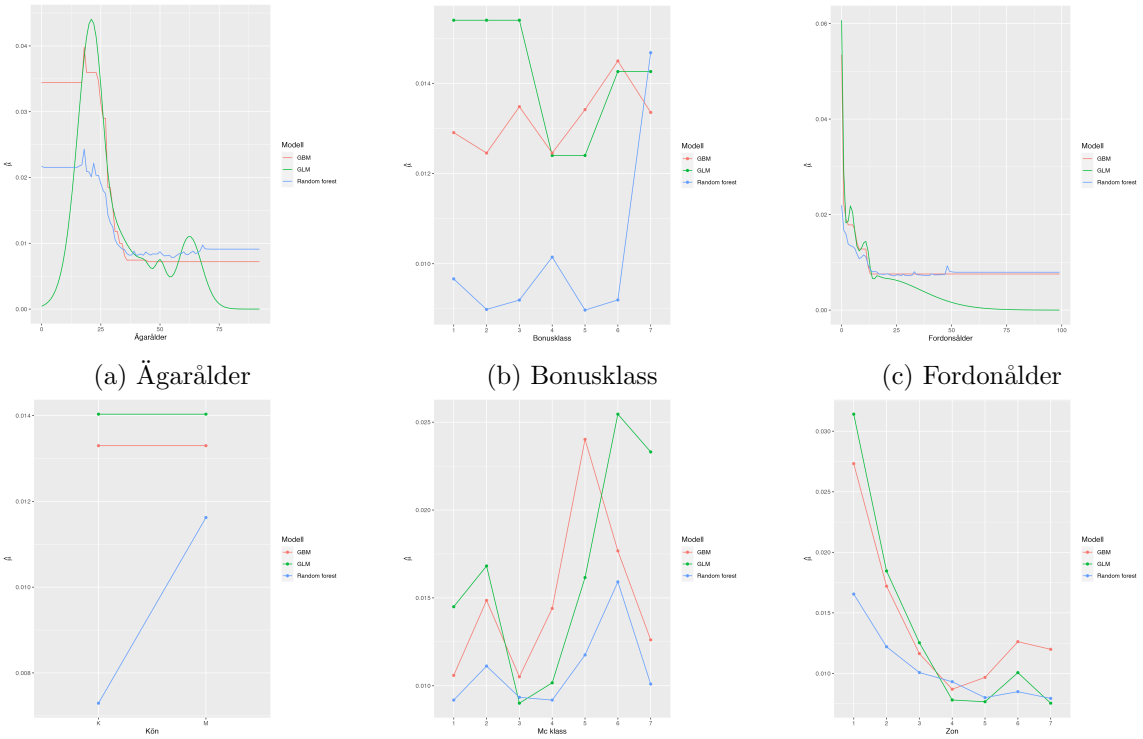
Tabell 11: MSE för respektive modell samt fidelity för respektive surrogatmodell



Figur 15: De olika modellernas prediktioner i storleksordning tillsammans med det observerade medelvärdet

Ännu ett resultat som tyder på att samtliga modeller följer varandra relativt nära är Partial dependence plots (PDP), som visas nedan i Figur 16. Det visar sig att för varje kovariat följer de respektive modellerna varandra i trender, med några undantag. Till exempel kan vi observera att för variabeln Kön är värdena konstanta för GLM och GBM, medan Random Forest uppvisar en växande trend. De största skillnaderna kan ses i Bonusklass, MC-klass och Kön. De kontinuerliga variablerna förhåller sig relativt nära varandra med mindre avvikelser, och detta gäller även för den kategoriska variabeln Zon. Detta stärker resultatet att det egentligen inte finns någon poäng med att använda någon annan modell än GLM,

då den presterade näst bäst och ändå har förmågan att vara tolkningsbar.

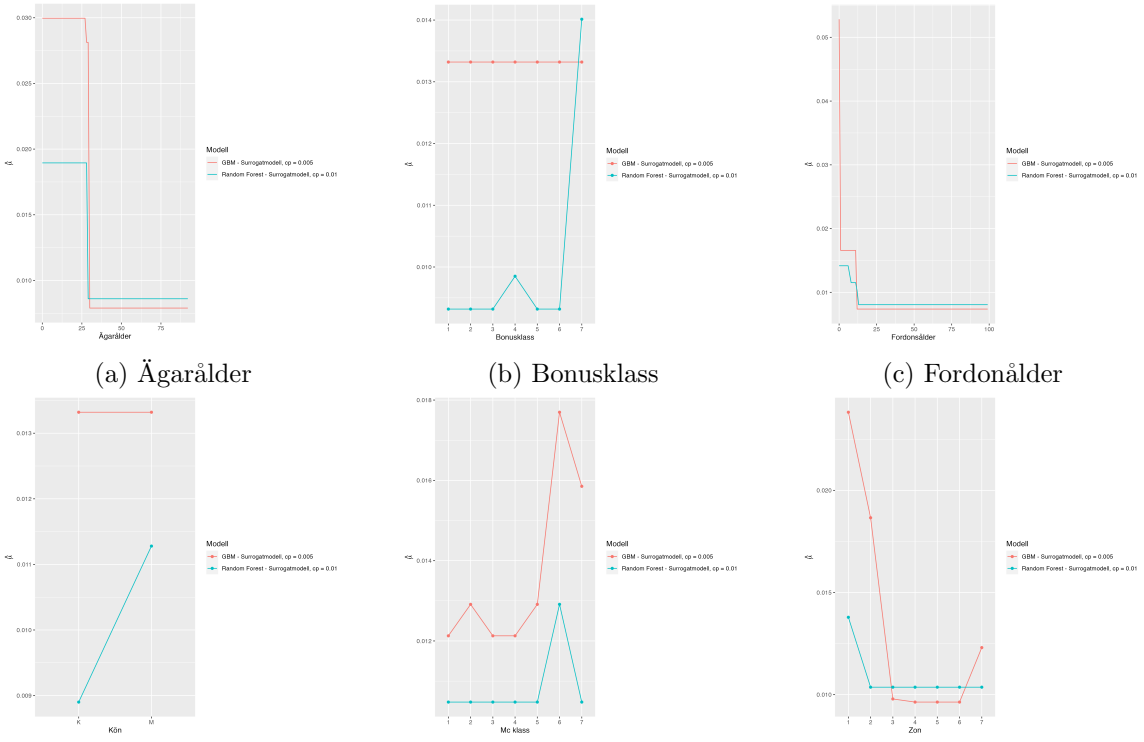


(a) Ägarålder (b) Bonusklass (c) Fordonålder

(d) Kön (e) MC klass (f) Zon

Figur 16: Partial dependence plots för respektive kovariat och modell (Röd = GBM, GRÖN = GLM, BLÅ = Random forest)

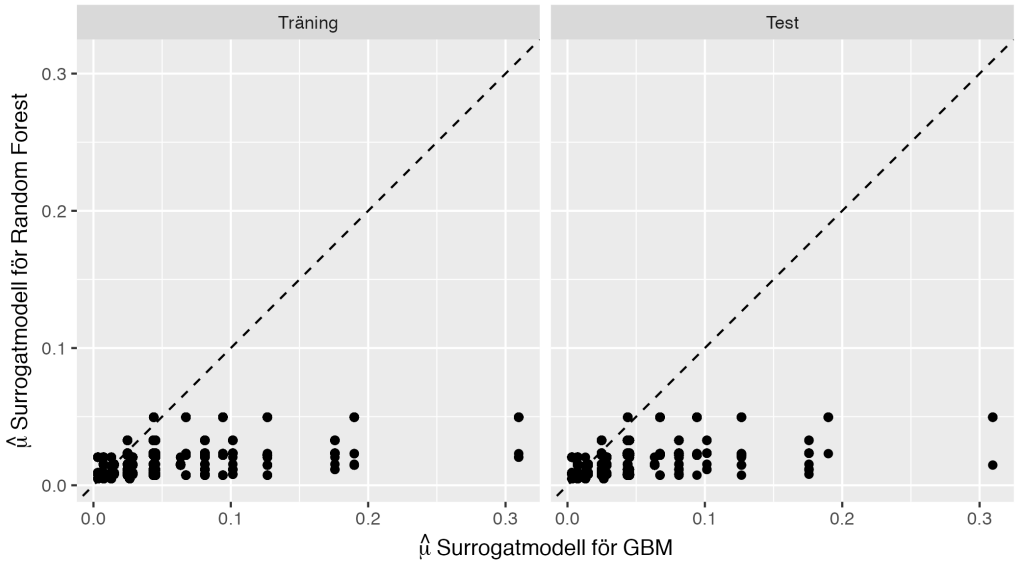
Ett resultat som visas i Figur 17 är att effekterna är lägre för surrogatmodellen för Random Forest jämfört med samtliga kovariater. Det resulterar även i att surrogatmodellen för Random Forest är mindre varierande och tenderar att ha samma effekt för flera klasser, medan surrogatmodellen för GBM har en mer varierande effekt över klasserna. Det bör dock poängteras att surrogatmodellen för Random Forest underestimerar sina prediktioner jämfört med surrogatmodellen för GBM, vilket visas i Figur 18. Det visar sig att prediktionerna är ibland ungefär 6 gånger lägre. Något som tydligt framgår i figuren är att surrogatmodellen för GBM har en större spridning mellan sina prediktionsvärden.



(a) Ägarålder (b) Bonusklass (c) Fordonålder

(d) Kön (e) MC klass (f) Zon

Figur 17: Partial dependence plots för respektive kovariat och surrogatmodell (Röd = GBM, BLÅ = Random Forest)



Figur 18: GBM surrogatmodells (cp = 0.005) prediktioner mot Random Forest surrogatmodells (cp = 0.01) prediktioner

Sammanfattningsvis visar resultaten att användningen av splines för att modellera ägarålder och fordonsålder leder till en förbättrad modell jämfört med att använda klassindelning. Random Forest-modellen presterar bäst när det gäller MSE-resultat för valideringsdata, medan GLM-modellen hamnar på andra plats och sist hamnar GBM-modellen. Det ska dock poängteras att skillnaderna mellan modellerna är inte särskilt stora. Detta illustrerades i Figur 15, vilket visade att modellerna följer varandra väl, med vissa avvikelser bland de högsta värdena. Detta visas även i Figur 24, där dessa observationer har brutits ner i intervall för att få en tydligare bild hur modellerna följs åt.

Resultaten visar också att surrogatmodellerna har ett högt fidelity- och lågt MSE-värde, vilket tyder på att de kan användas som approximativa modeller för att förenkla de ursprungliga black-box-modellerna. Det är dock viktigt att betona att dessa surrogatmodeller kan vara användbara i vissa tillämpningar, såsom att öka tolkningsbarheten. Samtidigt har de också sina begränsningar genom att de kan vara mindre exakta i sina prediktioner och instabila jämfört med den ursprungliga black-box-modellen.

6 Sammanfattning & diskusion

För att sammanfatta allt kan vi konstatera att ett flertal GLM:er har anpassats genom att exkludera eller behandla kovariater som klasser eller splines. Det visade sig att behandling av förarålder och ägarålder som splines genererade den bästa modellen (Modell 6), medan exkludering av kön och bonusklass gav den bästa modellen där alla kovariater behandlades som klasser (Modell 3). Dessa modeller jämfördes sedan med GBM och Random forest, där endast Random forest-modellen presterade bättre. En möjlig förklaring till att Random Forest presterade bäst kan vara att det finns tydligt starka kovariat i data. Detta kunde observeras i Figur 6, där det visades att ägarålder och fordonsålder användes i de flesta splittar. Som tidigare beskrivet hanterar Random Forest-metoden detta genom att slumpmässigt välja kovariat som kan användas vid respektive splitt. Genom att använda denna slumpmässiga urvalsmetod kan Random Forest undvika att överanpassa modellen till specifika kovariat och istället dra nytta av flera variabler för att skapa robusta prediktioner. Denna egenskap hos Random Forest kan vara en av anledningarna till dess bättre prestanda i jämförelse med andra modeller i studien. Det är dock viktigt att notera att samtliga modeller hade låga MSE-värden och var mycket likvärdiga.
Surrogatmodeller anpassades även för dessa black-box-modeller, och förhållandet mellan de två modellerna mättes med hjälp av fidelity. Det visade sig att surrogatmodellerna för GBM genererade högst fidelity-värden, med 0.91 för cp = 0.001 och 0.83 för cp = 0.005 vid anpassning på valideringsdata. Surrogatmodellerna för Random forest genererade något lägre fidelity-värden, med 0.79 för cp = 0.001 och 0.65 för cp = 0.01. Det bör dock påpekas att surrogatmodellerna för Random forest hade lägre MSE än GBM, men cp = 0.001 resulterade i besultsträd med över 50 respektive 70 slutnoder. Detta var något som inte är optimalt, vilket ledde till slutsatsen att cp = 0.005 och cp = 0.01 bör användas. Detta genererade ett mindre träd med 14-22 slutnoder, vilket gjorde det mer lättförståeligt.
En relevant reflektionsfråga är om det inte skulle vara bättre att direkt anpassa ett enkelt beslutsträd. I detta specifika fall med datamaterialet 'MCcase.txt" skulle det förmodligen inte ha haft någon större betydelse om man skapade en surrogatmodell eller om man anpassade ett enkelt beslutsträd direkt. Vi har tidigare sett att den enda modellen som presterade bättre var Random forest-modellen tillsammans med dess surrogatmodell. Detta kan bero på att det inte finns tydliga korrelationer mellan kovariaterna i detta datamaterial, vilket kunde analyseras i Figur 3. Eftersom det inte finns några tydliga korrelationer kan det vara

en förklaring till varför GLM presterade bättre än GBM. Detta i sin tur innebär att det egentligen inte finns någon anledning att anpassa surrogatmodeller. Om det däremot fanns korrelationer mellan kovariaterna skulle GBM förmodligen prestera bättre. Detta på grund av att GBM har en stark förmåga att hantera och dra nytta av sådana korrelationer, vilket kan leda till mer precisa prediktioner och bättre modellprestanda. Notera att baserat på våra nuvarande resultat verkar det som att GBM och anpassningen av en surrogatmodell fungerar bra i frånvaro av tydliga korrelationer, men om korrelationer finns bör GBM tillsammans med en surrogatmodell vara ett bättre alternativ än en GLM.

En annan övervägning som bör beaktas är om vi förlorar något i prediktionsförmåga genom att välja en mer tolkningsbar modell, som en surrogatmodell. Är det värt att kompromissa med prediktionsförmågan för att få en modell som är lättförståelig? Som tidigare nämnts försämrades inte prediktionsförmågan avsevärt, utan den förblev inom en rimlig nivå. Dessa surrogatmodeller visade heller inga tydliga avvikelser från GLM, vilket stärker argumentet att de fungerade bra. Det är dock viktigt att komma ihåg att detta alltid bör tas i beaktning vid modellering. En surrogatmodell som presterar sämre än en vanlig GLM fyller ingen funktion, eftersom en GLM redan är tolkningsbar.

Vi har tidigare nämnt att vanliga beslutsträd oftast är instabila, men vi noterade även att Random Forest visade en betydande ökning av MSE vid beräkning på valideringsdata jämfört med träningsdata. Ytterligare analyser har ännu inte utförts avseende stabiliteten hos trädmodellerna, vilket leder till följande förbättringsförslag. Genom att använda bootstrap eller simulera data kan vi analysera stabiliteten i våra modeller. Det innebär att vi kan bedöma hur väl modellerna bibehåller sin prestanda under olika stressade förhållanden. Vi vet också sedan tidigare att GLM är en robust modell som klarar av variationer i data och ger liknande resultat som om dessa variationer inte fanns. Genom dessa simuleringar kan vi även testa om våra trädbaserade modeller uppvisar samma egenskaper och hur stor variation de kan hantera. Simuleringar ger oss också möjlighet att begränsa antalet observationer som används samt variationen i datan. Detta gör att vi kan precisera varför vissa trädmodeller fungerar bättre än andra under olika situationer. Sammanfattningsvis kan vi med hjälp av simulering analysera frågan om det kommer att ges ett helt annat kvalitativt träd nästa gång analysen genomförs, vilket i sin tur skulle leda till en annorlunda kvalitativ tariff.

I denna rapport har vi lagt störst fokus på prediktionsförmågan, mätt med MSE, men ett annat förslag skulle vara att också beakta modellkomplexiteten vid jämförelse av trädmodellerna, samt den beräkningskapacitet som krävs för att träna och anpassa olika modeller. Detta beror på att modellerna kan variera i beräkningsintensitet och kräva olika resurser, såsom processortid, minne och datalagringsutrymme. Frågan blir då: Är det värt att offra flera timmars kodkörning i hopp om att uppnå en förbättring i prediktionsförmågan, även om det kanske inte resulterar i någon förbättring överhuvudtaget?

Slutligen, en annan intressant aspekt att överväga är att inkludera finansiella parametrar, såsom självrisk och självriskperiod, i modelleringen. Det vore intressant att utforska om det finns någon korrelation mellan dessa parametrar och skador. Genom att analysera samban-

det mellan finansiella parametrar och skadeutfall kan vi få en djupare förståelse för hur de påverkar varandra och därigenom förbättra modellens prediktionsförmåga. Vanligtvis hanteras dessa parametrar separat från modellerna och beräknas på en rent matematisk grund. Detta är också något som rekommenderas i läroböcker. Dock finns det en rimlighet i att anta att personer med högre benägenhet att drabbas av skador tecknar försäkringar med exempelvis lägre självrisk än de med lägre risk. Därför skulle det vara något att överväga att inkludera i framtida studier.

7 Appendix

7.0.1 GLM

7.0.1.1 Likelihood-kvot test

Genom att använda metoden likelihood-kvot test (LRT) kan det analyseras ifall kovariater kan exkluderas från en GLM. LRT bygger på att testa två modeller mot varandra, med och utan en specifik kovariat. Modellerna måste vara hierarkiskt ordnande med en nollhypotes som delmodell av grundmodellen, genom att ange vissa kovariater i grundmodellen till noll (Ohlsson & Johansson, 2010).
Anta två modeller H_r och H_s, så att $H_s \subset H_r$. Låt $\hat{\boldsymbol{\mu}}^{(r)}$ vara MLEs för H_r och likadant för H_s. Detta ger då att LRT statistikan för att testa H_s mot H_r ges av $D^*(\mathbf{y}, \hat{\boldsymbol{\mu}}^{(s)}) - D^*(\mathbf{y}, \hat{\boldsymbol{\mu}}^{(r)})$. Det ska dock poängteras att för att modellerna ska vara hierarkiskt ordnande, krävs det att de kommer från samma EDM familj, med ett gemensamt ϕ. Under de generella förhållanden följer LRTs approximativt en χ^2-fördelning, där antalet frihetsgrader är $f_r - f_s$, det vill säga om modellerna f_r och f_s har icke-redundanta β-parametrar. $D(\cdot)$ motsvarar en deviansfunktion som är given av en fördelningen, för poisson fallet defineras LRT enligt följande:

$$D^*(\mathbf{y}, \hat{\boldsymbol{\mu}}^{(s)}) - D^*(\mathbf{y}, \hat{\boldsymbol{\mu}}^{(r)}) = 2 \sum_i w_i y_i \log \left(\frac{\hat{\boldsymbol{\mu}}_i^{(r)}}{\hat{\boldsymbol{\mu}}_i^{(s)}} \right) + \sum_i w_i (\hat{\boldsymbol{\mu}}^{(s)} - \hat{\boldsymbol{\mu}}^{(r)}) \tag{27}$$

7.0.1.2 AIC

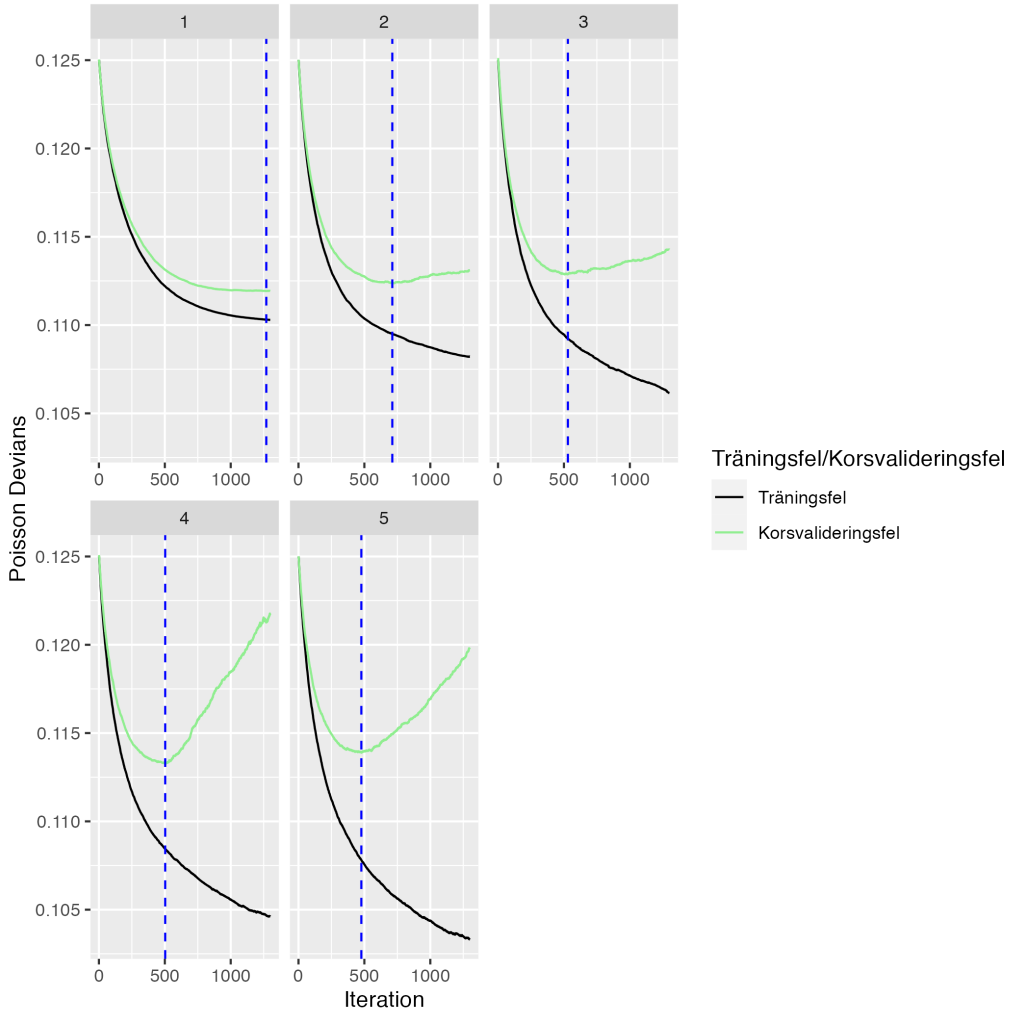
Akaike informationkreiteret (AIC) och LRT är två olika metoder för att avgöra vilken modell som är bäst anpassat till data. I Föregående kapitel beskrevs LRT, vilket visade sig vara en bra metod för att jämföra modeller. Dock visade det sig att denna metod ej tar hänsyn till modellkomplexiteten, detta skulle kunna leda till att den modell med högst komplexitet alltid väljs vilket i sin tur skulle leda till överanpassning. Det vill säga LRT tar ej hänsyn till överanpassning, och ett sätt undvika denna överanpassning genom ett stort antalet kovariater. AIC tar däremot hänsyn till modellkomplexiteten genom att inkluderat antalet kovariater i modellen vid beräkningen. AIC straffar nämligen större modeller genom

att inkludera en term som beror på antalet kovariater, genom denna term ska modeller med snarlika prestations resultat ska modellen med lägst AIC väljas (Agresti, 2002).

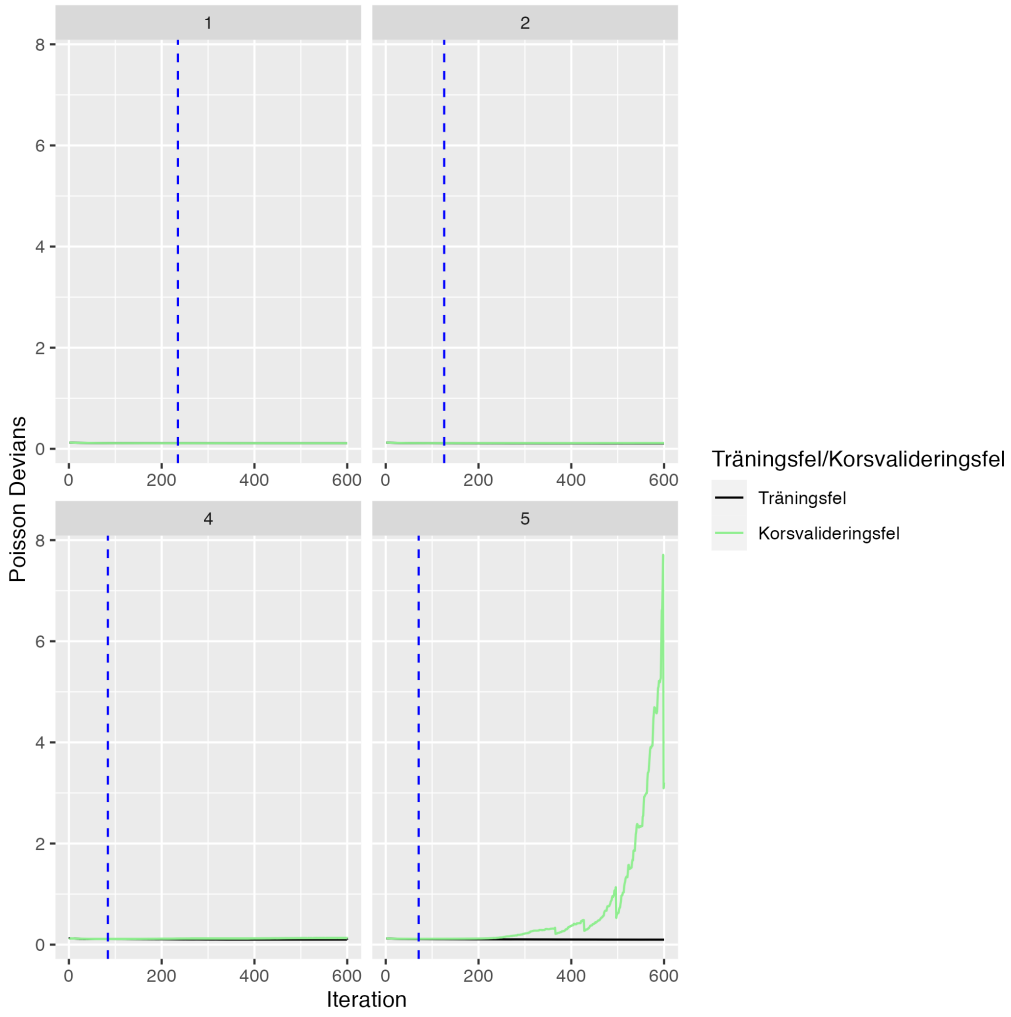
$$AIC = -2(\text{maximum log likelihood} - p) \tag{28}$$

I (28) presenteras hur AIC beräknas. Dock ska modellvalet inte bara grundas på AIC utan det är lämpligt att använda både AIC och LRT.

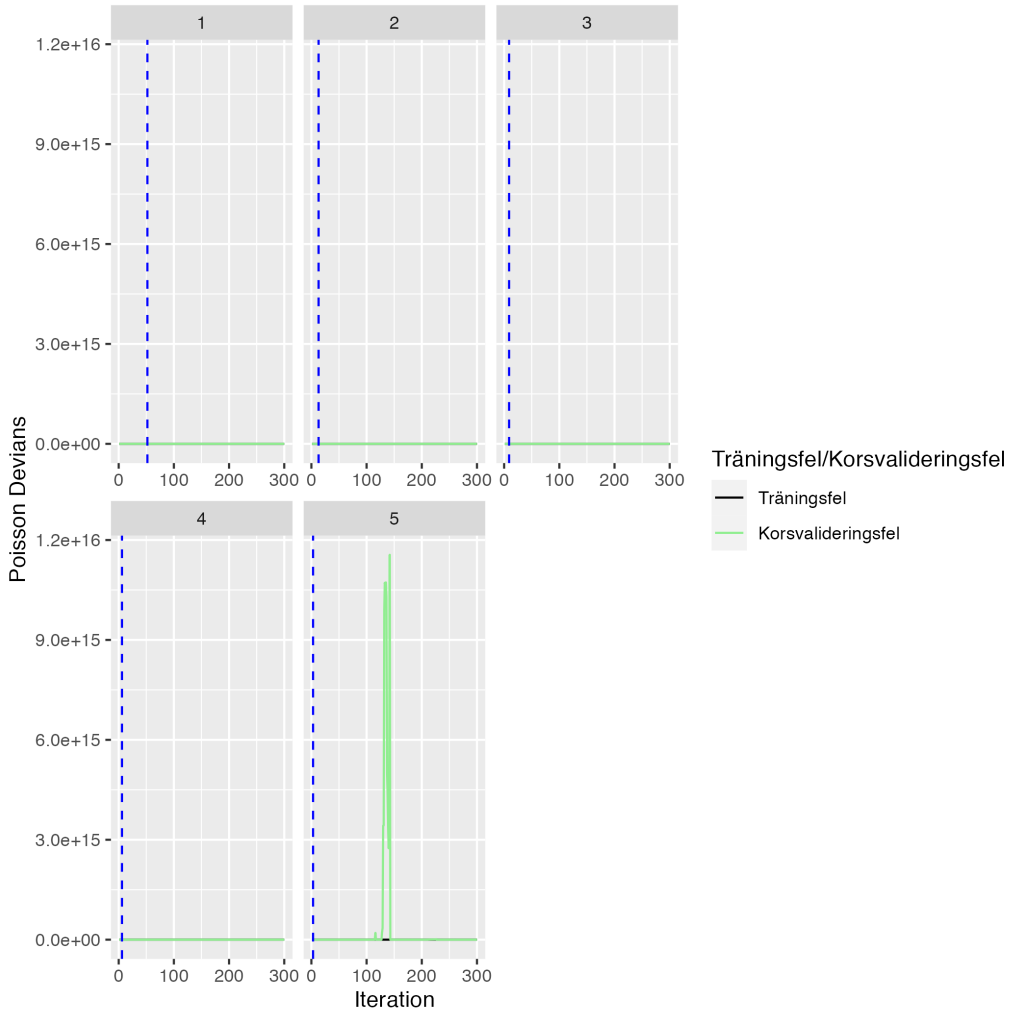
7.1 Figurer



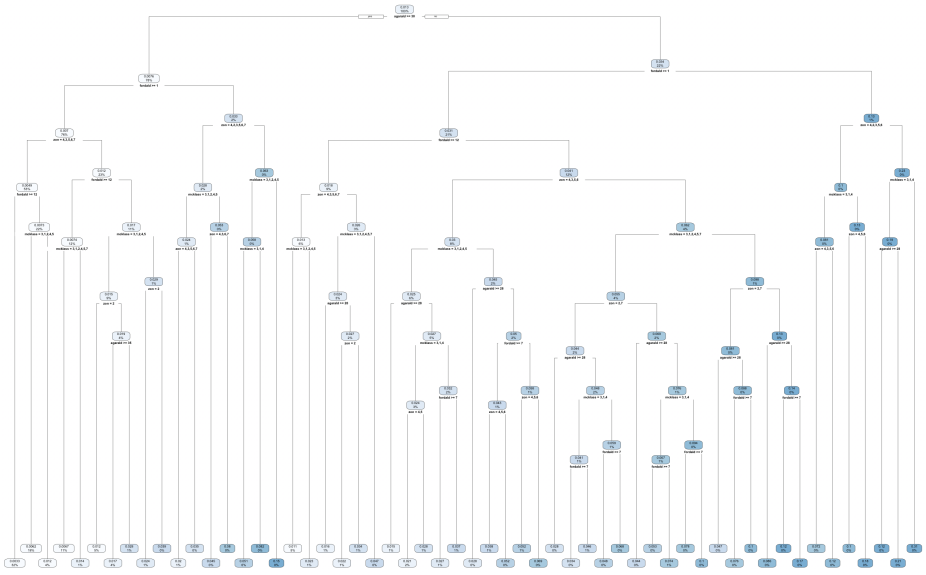
Figur 19: GBM för olika interaction.depth vid n.trees = 1300 och learning rate = 0.01



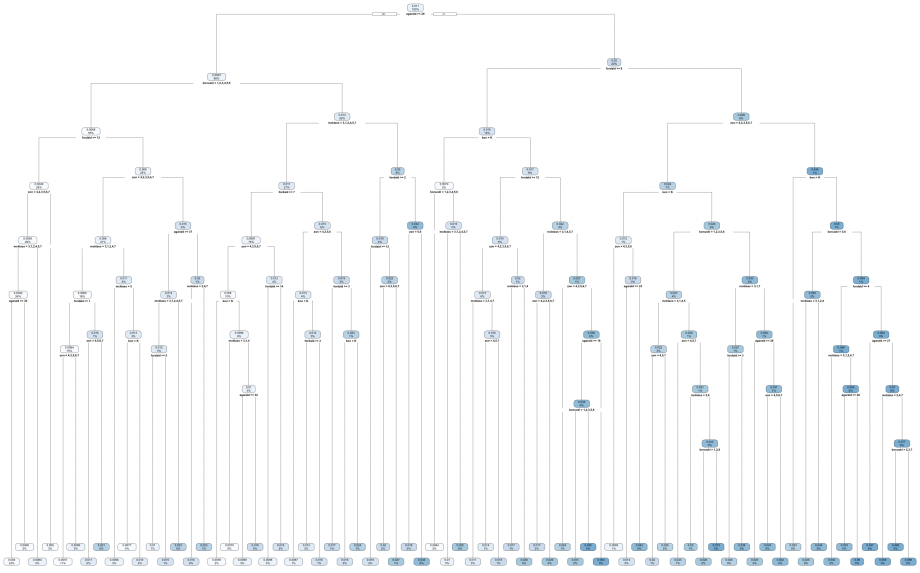
Figur 20: GBM för olika interaction.depth vid n.trees = 600 och learning rate = 0.05



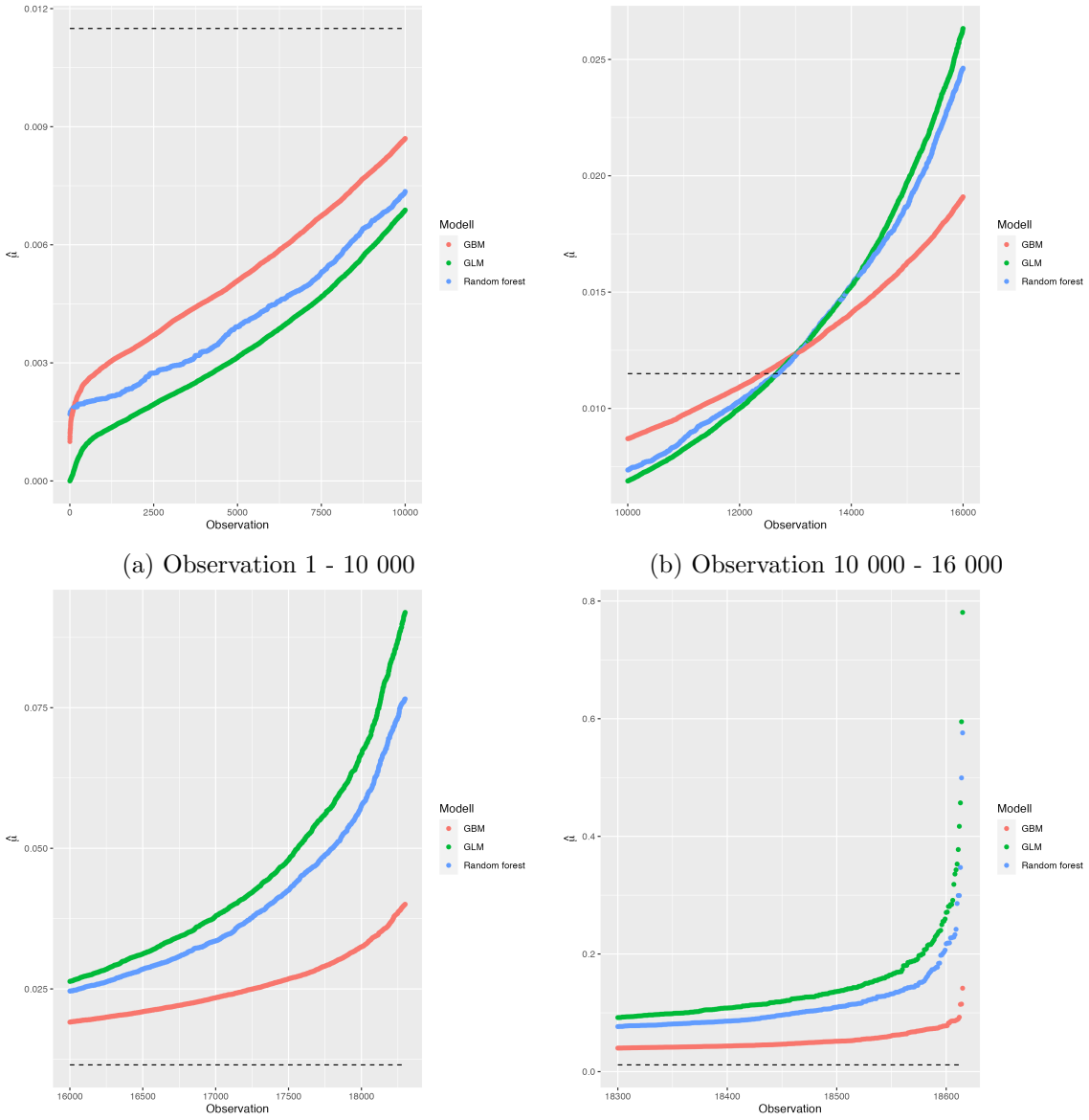
Figur 21: GBM för olika interaction.depth vid n.trees = 300 och learning rate = 0.5



Figur 22: Surrogatmodell för GBM, cp = 0.001



Figur 23: Surrogatmodell för Random forest, cp = 0.001



(a) Observation 1 - 10 000

(b) Observation 10 000 - 16 000

(c) Observation 16 000 - 18 300

(d) Observation 18 300 - 18 615

Figur 24: Modellernas prediktioner storleksordning tillsammans med det observerade medelvärdet

Referenser

Agresti, A. (2002). *Categorical data analysis*. John Wiley & Sons.

Greenwell, B., Boehmke, B., Cunningham, J. & Developers, G. (2022). gbm: Generalized boosted regression models [handbok till mjukvara]. Hämtad från `https://CRAN.R-project.org/package=gbm` (R package version 2.1.8.1)

Hara, S. & Hayashi, K. (2018, 09–11 Apr). Making tree ensembles interpretable: A bayesian model selection approach. I A. Storkey & F. Perez-Cruz (red.), *Proceedings of the twenty-first international conference on artificial intelligence and statistics* (vol. 84, s. 77–85). PMLR. Hämtad från `https://proceedings.mlr.press/v84/hara18a.html`

Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (vol. 2). Springer.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An introduction to statistical learning: with applications in r*. Spinger.

Johansson, B. & Ohlsson, E. (2022). Gradient boosting machines and non-life insurance pricing-lecture notes. *Available at SSRN 4294965*.

Liaw, A. & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18-22. Hämtad från `https://CRAN.R-project.org/doc/Rnews/`

Maillart, A. (2021). Toward an explainable machine learning model for claim frequency: a use case in car insurance pricing with telematics data. *European Actuarial Journal*, 1–39.

Ohlsson, E. & Johansson, B. (2010). *Non-life insurance pricing with generalized linear models* (vol. 174). Springer.

R Core Team. (2020). R: A language and environment for statistical computing [handbok till mjukvara]. Vienna, Austria. Hämtad från `https://www.R-project.org/`

R Core Team. (2022). R: A language and environment for statistical computing [handbok till mjukvara]. Vienna, Austria. Hämtad från `https://www.R-project.org/`

Therneau, T. & Atkinson, B. (2022). rpart: Recursive partitioning and regression trees [handbok till mjukvara]. Hämtad från `https://CRAN.R-project.org/package=rpart` (R package version 4.1.19)