Koneoppimismenetelmät riskihenkivakuutusten raukeamisen ennustamisessa

SHV-työ (suppea)

Olli Poutanen

25. huhtikuuta 2025

Abstract

In this work, the aim is to study selected machine learning methods and apply them in practice to a selected actuarial problem, namely the prediction of life insurance lapse risk. The selected machine learning methods were gradient boosting and neural networks.

First, a theoretical background for machine learning and the selected methods is presented. Second, the determinants of life insurance lapse risk and previous research on the topic are discussed. Finally, the process for applying the selected machine learning methods to life insurance lapse prediction is described and the modeling results are presented.

The results imply that the prediction of life insurance lapse can be improved by using machine learning. Both of the selected machine learning methods improved the lapse prediction results compared to more traditional statistical methods. Gradient boosting (XGBoost) performed slightly better than the neural network model and, especially taking into account the time efficiency of the fitting process and interpretability of the model, the XGBoost model was concluded to be better suited for lapse prediction.

The usefulness of the machine learning models in lapse prediction was also studied by using an economic metric for the lapse management. The result in terms of the economic metric was improved when the model was fit based on regression for the economic gain instead of using a pure classification. The actual economic gain, however, depends on how the parameters for the economic model are determined. Depending on the parameter selection, the resulting economic gain could remain limited or be quite significant. Therefore, the benefits of the lapse prediction depend on how efficient the insurance company is in their lapse management and on the overall profitability of the insurance product.

All in all, the machine learning methods are concluded to be useful in lapse prediction as well as other actuarial applications and their significance is also likely to increase in the future.

Sisällys

"If a machine can learn the value of human life, maybe we can too."

Sarah Connor, Terminator 2: Judgement Day

1 Johdanto

Tilastollisten mallien käyttö ei ole uutta aktuaaritieteissä. Esimerkiksi lineaaristen ja yleistettyjen lineaaristen mallien käyttö tariffien asettamisessa ja varausten laskennassa on ollut yleistä jo pitkään. Tilasto- ja tietojärjestelmätieteiden kehittyessä joustavammat mallit, kuten koneoppimismallit, ovat kuitenkin päihittäneet lineaariset mallit usealla tieteenalalla. Aktuaaritieteissä ne ovat viime vuosiin saakka kuitenkin jääneet vähemmälle huomiolle.

Vaikkakin lineaariset ja yleistetyt lineaariset mallit ovat suhteellisen helppokäyttöisiä, helposti selitettävissä ja niillä on hyviä tilastollisia ominaisuuksia, saattavat ne olla liian rajoittavia monimutkaisten ongelmien kuvaamiseen. Koneoppimisalgoritmit oppivat datan pohjalta säännönmukaisuuksia ilman, että käyttäjän tarvitsee etukäteen määritellä datan ominaisuuksien välistä riippuvuussuhdetta. Esimerkiksi vakuutusriskien takana vaikuttavat ilmiöt ovat usein monimutkaisia ja epälineaarisia, jolloin koneoppimismenetelmät voisivat sopia niiden kuvaamiseen perinteisiä malleja paremmin.

Raukeamisriskillä tarkoitetaan riskiä siitä, että vakuutusyhtiön saama tuotto tai vastuiden arvo muuttuu sen johdosta, että vakuutuksenottajat käyttävät vakuutukseen sisältyviä optioita, kuten mahdollisuutta irtisanoa vakuutus, odotetusta poikkeavasti. Raukeamisriski on yksi merkittävimmistä henkivakuuttamisen riskeistä ja sillä on merkittävä vaikutus vakuutusyhtiön kannattavuuteen ja jopa vakavaraisuusasemaan. Aikaisemmissa tutkimuksissa on tunnistettu useita raukeamisriskiin mahdollisesti vaikuttavia tekijöitä. Tällaisia ovat erilaiset vakuutettuun liittyvät ominaisuudet kuten ikä ja sukupuoli, itse vakuutuksen ominaisuudet kuten voimassaolovuosi ja vuosimaksu sekä erilaiset makromuuttujat kuten korkotaso ja työllisyystilanne.

Tässä työssä esitellään aktuaaritieteisiin parhaiten soveltuvia koneoppimismenetelmiä ja sovelletaan niitä riskihenkivakuutusten raukeamisriskin ennustamiseen. Koneoppimismallien hyötyjä ja ominaisuuksia sekä mallinnuksesta saatavia tuloksia verrataan perinteisempiin malleihin ja niistä saataviin tuloksiin.

Tavoitteena on tutkia voidaanko riskihenkivakuutuksen raukeavuuden ennustettavuutta parantaa koneoppimisen menetelmillä, selvittää mitkä tekijät vaikuttavat raukeavuuteen ja esitellä koneoppimisen menetelmiä yhdessä aktuaaritieteiden merkittävässä sovelluskohteessa. Koneoppimismenetelmillä tehdyn mallinnuksen tuloksia olisi mahdollista hyödyntää esimerkiksi raukeamisriskin hallinnassa kontaktoimalla sellaisia asiakkaita, joilla raukeamisriski on mallien perusteella suurempi tai raukeamiskin arvioimisessa esimerkiksi vakavaraisuuslaskennassa.

2 Koneoppiminen

2.1 Yleistä koneoppimisesta

Koneoppimisella tarkoitetaan laskentamenetelmiä, jotka hyödyntävät "kokemusta" mallin ja sen tuottamien ennusteiden parantamiseksi [14]. Kokemuksella tarkoitetaan tässä analysoitavan datan hyödyntämistä mallin "opettamiseen". Koneoppimisen voikin nähdä samankaltaisena prosessina kuin ihmisen tekemä päättely, joka perustuu menneeseen kokemukseen; yritykseen ja erehdykseen. Esimerkiksi neuroverkoilla, jotka ovat yksi koneoppimisen menetelmistä, voidaan nähdä olevan analogia ihmisen aivojen toiminnalle, jossa neuronit lähettävät synapseja pitkin impulsseja toisilleen. [6], [14]

Koneoppimisella on vahva kytkös tilastotieteeseen, ja tilastollinen oppiminen on sinänsä vain tapa sovittaa tehokkaasti tilastollinen malli opetusdatan perusteella. Apuna tässä hyödynnetään viime vuosina nopeasti kehittynyttä tietokoneiden laskentatehoa ja uusia laskenta-algoritmeja. Koneoppiminen linkittyy siten vahvasti tietojenkäsittelytieteisiin.

Koneoppimista hyödyntäville ongelmille on tyypillistä, että syötemuuttujien, joista käytetään usein myös englanninkielistä termiä feature, avulla pyritään ennustamaan kohdemuuttujan arvoja. Tätä kutsutaan ohjatuksi oppimiseksi. Ohjatussa oppimisessa koneoppimisalgoritmiin syötetään osa analysoitavasta datasta eli opetusdata, jonka avulla malli oppii syötteen perusteella muodostamaan kohdemuut-

tujan arvon, joka opetusdatassa on tiedossa. Tämän jälkeen opetettu malli ennustaa kohdemuuttujan arvoja uudelle datalle eli testidatalle, jossa kohdemuuttujan arvot eivät ole tiedossa. Mallin toimivuutta kuvaa se, kuinka hyvin se onnistuu ennustamaan kohdemuuttujan arvoja uudelle datalle.

Mallin sovituksessa voidaan käyttää apuna vielä kolmatta validointidataa tai ristiinvalidointia. Tällöin voidaan muuttaa mallin valintaa tai hyperparametreja validointidatan perusteella. Tällä pyritään välttämään mallin ylisovittaminen eli mallin rakentaminen liian monimutkaiseksi ja opetusdataa vastaavaksi, mikä yleensä heikentää sen selitysvoimaa uudelle datalle. [6]

Yllä kuvatun ohjatun oppimisen lisäksi koneoppimista voidaan soveltaa ohjaamattoman oppimisen avulla. Tällöin mallin ei ole tarkoitus ennustaa jotakin tiettyä muuttujaa, vaan ennemminkin löytää sille annetusta datasta toistuvia rakenteita esimerkiksi klusterianalyysin tai pääkomponenttianalyysin avulla. Klusterianalyysissä on tarkoituksena löytää havaintojen ryhmiä, joiden sisällä havainnot muistuttavat toisiaan ja eroavat muiden ryhmien havainnoista, jonkin tietyn ominaisuuden tai niiden yhdistelmän suhteen. Pääkomponenttianalyysissä taas on tarkoituksena yksinkertaistaa käytetty muuttujajoukko pienemmällä määrällä uusia alkuperäisistä muuttujista johdettuja muuttujia eli pääkomponentteja käyttäen. [6] Tässä työssä tarkastelu keskittyy kuitenkin ohjatun oppimisen menetelmiin.

Koneoppimista voidaan verrata perinteisempiin tilastotieteen menetelmiin, kuten perinteiseen lineaariseen regressioanalyysiin, jossa estimoidaan riippuvuussuhde selittävien muuttujien ja selitettävän muuttujan välille usein pienimmän neliösumman menetelmää käyttäen. Lineaaristen mallien etuna on niiden helppo tulkittavuus ja sovitettavuus. Tavallisen lineaarisen regression yleistys, yleistetty lineaarinen malli (GLM), on ollut perinteinen menetelmä esimerkiksi vakuutusten tariffihinnan määrittelyssä historiallisen vahinkoaineiston pohjalta. Siinä riippuvuussuhde selittävien muuttujien ja selitettävän muuttujan välillä on lineaarinen linkkifunktion g suhteen:

$$E[Y|X_1, X_2, ..., X_p] = g^{-1}(\beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p).$$

Esimerkiksi kuvattaessa vahinkokappaleiden määrä tietyllä aikavälillä ajatellaan kuvattavan ilmiön noudattavan Poisson-jakaumaa, jolloin linkkifunktio on luonnollinen logaritmifunktio.

Vaikka lineaarinen regressio ja yleistetyt lineaariset mallit pystyvätkin usein kuvaamaan monia ilmiöitä kohtuullisen tarkasti ja ovat koneoppimismalleihin nähden helposti tulkittavia ja helppokäyttöisiä, mahdollistaa koneoppiminen kuitenkin vielä monimutkaisempien ja monesti paremmin todellisuutta kuvaavien riippuvuussuhteiden mallintamisen. Todellisuudessa onkin itse asiassa hyvin epätodennäköistä, että todellinen mallinnuksen kohteena oleva funktio olisi lineaarinen. [6]

Luonnollinen lineaaristen menetelmien laajennus on yleistetty additiivinen malli, jossa kuvattava riippuvuus voidaan mallintaa muutenkin kuin muuttujien lineaarisen kombinaation avulla:

$$E[Y|X_1, X_2, ..., X_p] = g^{-1}(\alpha + f_1(X_1) + f_2(X_2) + ... + f_p(X_p)),$$

missä f voi olla mikä tahansa funktio. Tämä mahdollistaa epälineaaristen riippuvuusuhteiden luomisen yksittäisille syötemuuttujille.

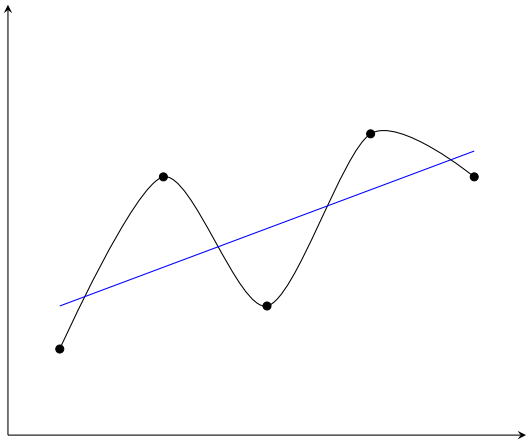
Vieläkin yleistetympi menetelmä on optimiprojektioregressio, jossa käytetty malli on muotoa:

$$E[Y|X_1, X_2, ..., X_p] = \sum_{m=1}^{M} f_m(w_m^T \mathbf{X}),$$

missä $\mathbf{X}$ on syötemuuttujien vektori ja $w_m^T, m = 1, 2, ..., M$ ovat painovektoreita. Optimiprojektioregressiossa epälineaarisen riippuvuusuhteen luominen tapahtuu alkuperäisten muuttujien $X_1, X_2, ..., X_p$ lineaarisille kombinaatioille $V_m = w_m^T \mathbf{X}$. Näin saatavien harjannefunktioiden $f(V_m)$ määrää M ei ole mallia muodostettaessa rajoitettu. Tämä rakenne tekee mallista hyvin yleisen. Valittaessa riittävän suuri M onkin mahdollista approksimoida mikä tahansa jatkuva funktio eli kyseessä on universaali approksimaattori. Optimiprojektioregressio muistuttaa rakenteeltaan yhden piilokerroksen neuroverkkoa. Neuroverkkoja käsitellään tarkemmin kappaleessa 2.4.

Mallin yleisyys mahdollistaa todellisuutta monesti paremmin kuvaavien monimutkaisten riippuvuussuhteiden mallintamisen. Ongelmaksi muodostuu kuitenkin ylisovittaminen. Tätä on kuvattu yksinkertaistetusti kuvassa 1.

Kuviossa on muutaman datapisteen otos, jolle on sovitettu kaksi eri funktiota: lineaarinen ja epälineaarinen malli (splinikäyrä). Kuvan perusteella vaikuttaa siltä, että epälineaarinen malli pystyisi kuvaamaan annettua dataa paremmin, sillä se kulkee kaikkien annettujen datapisteiden läpi. Ongelmana on kuitenkin mallin toimivuus uudelle datalle. Tällöin riippuu kuvattavasta ilmiöstä, toimiiko malli



Kuva 1: Esimerkki mahdollisesta ylisovittamisesta

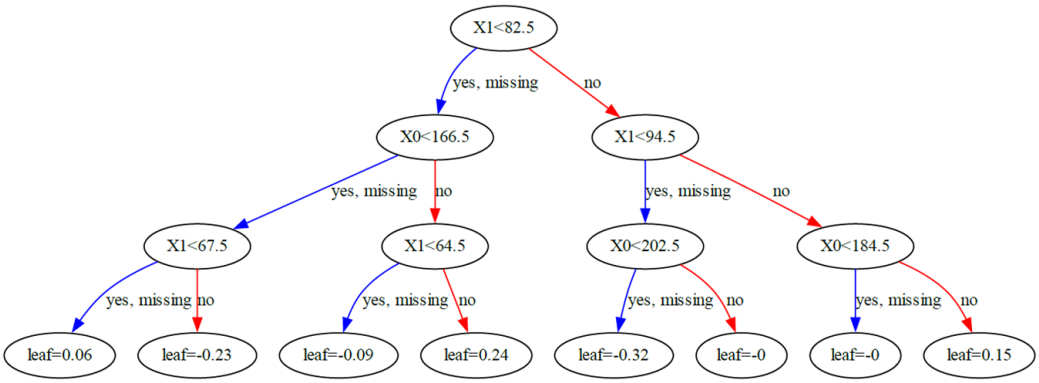
myös uudelle datalle. Voi olla, että data on peräisin hyvinkin lineaarisen riippuvuussuhteen omaavasta ilmiöstä, joka vain sisältää jonkin verran satunnaisuutta. Tällöin lineaarinen malli todennäköisesti toimisi paremmin uudelle datalle, kun taas epälineaarinen malli olisi rankasti ylisovitettu. Koneoppimismallien keskeinen haaste onkin mallin ominaisuuksien ja sovitusmenetelmien valinta siten, että pystytään käyttämään datan ominaisuuksia mahdollisimman hyvin ja välttämään samalla ylisovittaminen.

Mitä enemmän dataa on käytettävissä, sitä paremmin on mahdollista luotettavasti mallintaa datasta löytyviä epälineaarisuuksia. Pienelle datamäärälle lineaarinen malli voi olla paras ja ainoa luotettava tapa ilmiön kuvaamiseen, kun taas suuremmilla datamäärillä esimerkiksi koneoppimismenetelmien hyödyntäminen tulee kannattavammaksi. Saatavilla olevan datamäärän räjähdysmäinen kasvu onkin yksi merkittävä syy koneoppimismallien suosion kasvulle.

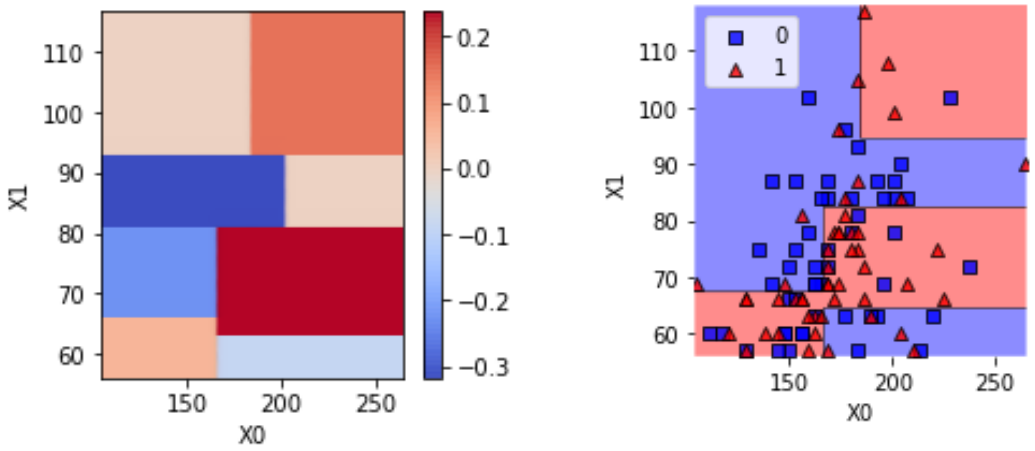
Myös lineaarisessa regressioanalyysissa voidaan esimerkiksi regularisaation, kuten Lasso- tai Ridge-regression avulla, välttää ylisovittamista asettamalla mallin monimutkaisuudelle rajoitus. Koneoppimismallien tapauksessa ylisovittamisen haaste kuitenkin korostuu.

2.2 Päätöspuut

Eräs paljon käytetty koneoppimisen menetelmä ovat päätöspuut. Niiden perusajatus on yksinkertainen ja intuitiivinen, mutta silti niistä johdetut mallit toimivat erinomaisesti moniin erilaisiin koneoppimisongelmiin. Päätöspuussa syötemuuttujien muodostama avaruus jaetaan suorakulmioihin, jonka jälkeen jokaiseen näistä sovitetaan yksinkertainen malli kuten vakio. Mallin sovittamiseen käytetty algoritmi valitsee ensin parhaan mahdollisen muuttujan ja jakopisteen yhdistelmän ja toistaa saman näin syntyvälle kahdelle alipuulle, kunnes haluttu lehtien määrä on saavutettu. [6]



Kuva 2: Esimerkki kahden muuttujan päätöspuusta



(a) Päätöspuun alueet ja ennustekertoimet

(b) Päätöspuussa käytetyn opetusdatan havainnot ja päätöspuun ennustamat arvot

Kuva 3: Esimerkki kahden muuttujan ja kolmen tason päätöspuusta

Kuvissa 2 ja 3 on esitetty esimerkki kahden muuttujan päätöspuusta kahdella tapaa. Kuva 2 kuvaa mallin päätöspuumuodossa. Kuvassa jokainen soikio kuvastaa päätöspuun solmuja, joista jokainen jakaa kyseiseen solmuun päätyvän datan kahtia. Esimerkiksi ne havainnot, joiden osalta muuttujan X1 arvo on pienempi kuin 82.5, päätyvät ensimmäisestä solmusta vasemmanpuoleiseen alipuuhun ja muut havainnot oikeanpuoleiseen alipuuhun. Lopulta jokainen havainto päätyy johonkin puun kahdeksasta lehdestä ja saa mallin ennustetta kuvaavan arvon kyseisen lehden perusteella.

Kuva 3 esittää saman päätöspuun kahden syötemuuttujan kaksiulotteisena avaruuteena. Nähdään, että kahden muuttujan avaruus on jaettu kahdeksaan suorakulmion muotoiseen alueeseen perustuen kuvan 2 mukaisiin jakoihin. Vasemman puoleisessa kuvassa on kuvattu väreillä mallin antama ennustearvo kyseiseen alueeseen päätyville havainnoille. Malli ennustaa niiden havaintojen, joiden ennustearvo on positiivinen, kuuluvan luokkaan 1 ja havaintojen, joiden ennustearvo on negatiivinen, kuuluvan luokkaan 0. Oikean puoleisessa kuvassa on esitetty alueiden väreillä mallin antama ennuste kategoriasta (sininen alue = kategoria 0, punainen alue = kategoria 1). Neliöt ja kolmiot kuvaavat havaintojen todellisia arvoja.

Nähdään, että malli luokittelee suurimman osan havainnoista oikeaan kategoriaan. Osa havainnoista on kuitenkin luokiteltu väärin. Jatkamalla datan jakamista useampaan alueeseen tai lisäämällä muuttujien määrää, on mahdollista sovittaa malli niin, että jokainen opetusdatan havainto tulee oikein luokitelluksi. Tällöin malli olisi kuitenkin todennäköisesti rankasti ylisovitettu, eikä toimisi siten uudelle datalle.

Suosittu algoritmi päätöspuumallin rakentamiseen on CART (Classification and Regression Trees) -algoritmi. Siinä data jaetaan kahteen osaan syötemuuttujien perusteella ja tämän jälkeen kuhunkin osaan valikoitunut data jaetaan jälleen kahteen osaan. Tätä toistetaan, kunnes haluttu jakojen määrä (puun koko) on saavutettu.

Jokaisessa vaiheessa optimaalinen jakoon käytettävä syötemuuttuja ja optimaalinen jakopiste valitaan perustuen epäpuhtausmittaan. Regressio-ongelmissa epäpuhtausmittana voidaan käyttää luontevasti pienintä neliösummaa ja klassifikointiongelmissa esimerkiksi Gini-indeksiä $(1 - \sum_{k=1}^{K} p_k^2)$ tai ristientropiaa $(-\sum_{k=1}^{K} p_k \log p_k)$, missä p_k on havainnon todennäköisyys kuulua luokkaan k kyseisestä solmua vastaavassa alueessa. Algoritmi valitsee pienimmän valitun epäpuhtausmitan arvon antavan syötemuuttujan ja jakopisteen kombinaation. [6]

Optimaalisen puun koon määrittämiseen voidaan käyttää puun karsimista aloittaen mahdollisimman suuresta puusta ja karsimalla puun solmujen määrää minimoimalla kustannuskompleksisuusmittarin arvoa, jossa puun koko mittaa mallin kompleksisuutta. Optimaalinen kompleksisuuden rajoittamisen taso voidaan määritellä esimerkiksi ristiinvalidoinnilla. [6]

Yksinkertainen päätöpuumalli ei ole useinkaan riittävä tai paras malli koneoppimisongelmien ratkaisuun. Niitä voidaan kuitenkin hyödyntää kehittyneempien koneoppimismallien rakennuspalikoina ja siten ne ovat tärkeä osa koneoppimismenetelmien kehitystä. Seuraavassa kuvataan, miten päätöspuita

voidaan hyödyntää osana boosting-malleja, joista on tullut yksi suosituimmista koneoppimismenetelmistä.

2.3 Gradient tree boosting

Boosting on yksi merkittävimmistä koneoppimismenetelmiin liittyvistä kehitysaskeleista. Se kuuluu ensemble-menetelmiin ja sen ideana on yhdistää useita "heikkoja" luokittelualgoritmeja yhdeksi "vahvaksi" luokittelualgoritmiksi. Heikolla luokittelualgoritmilla tarkoitetaan algoritmia, joka on vain hieman parempi kuin satunnainen arvaus. Yksittäinen luokittelualgoritmi gradient boosting -mallissa voi olla esimerkiksi yksinkertainen päätöspuu.

Boosting tree -malli on siten summaus edellä kuvatuista päätöspuista:

$$f_M(x) = \sum_{m=1}^{M} T(x; \Theta_m),$$

missä T on päätöspuufunktio, joka riippuu syötemuuttujista x sekä päätöspuun alueet ja kertoimet muodostavien parametrien joukosta Θ_m. Jokaisessa iteraatiossa m tulee ratkaista:

$$\hat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{N} (L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m)),$$

missä L on mallille määritelty tappiofunktio, joka klassifiointiongelmissa on usein ristientropia.

Gradient tree boostingissa ideana on, summata monta päätöspuuta, siten että seuraavan puun parametrit on sovitettu vastaamaan prosessin edellisessä vaiheessa saadulle mallille lasketun tappiofunktion negatiivista gradienttia. Mallissa jokainen uusi päätöspuu korjaa siten mallia siihen asti saadulle mallille lasketun tappiofunktion negatiivisen gradientin suuntaan eli siihen suuntaan, jossa tappiofunktio pienenee nopeimmin.

Gradient tree boostingin algoritmi voidaan esittää seuraavasti:

1. Initialize $f_0(x) = \arg\min_{\Theta_0} \sum_{i=1}^{N} L(y_i, T(x_i, \Theta_0))$
2. For $m = 1$ to M:

   a)
   $$g_i = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}, i = 1, ..., N$$

   b)
   $$\Theta_m = \arg\min_{\Theta} \sum_{i=1}^{N} (-g_i - T(x_i, \Theta))^2$$

   c)
   $$\rho_m = \arg\min_{\rho} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \rho T(x_i, \Theta_m))$$

   d)
   $$f_m(x) = f_{m-1}(x) + \rho_m T(x, \Theta_m)$$

3. Output $\hat{f}(x) = f_M(x)$

Eräs viime vuosina suosiota saanut gradient boosting -algoritmi on XGBoost-algoritmi. Sen nimi tulee sanoista Extreme Gradient Boosting. Siinä ideana on laajentaa edellä kuvattua gradienttilähestymistapaa ottamalla huomioon myös toisen asteen muutos. Tämä tehdään hyödyntämällä toisen asteen Taylorin approksimaatiota, jolloin yllä olevassa algoritmissa kohta 2. b) tulee muotoon:

$$\Theta_m = \arg\min_{\Theta} \sum_{i=1}^{N} \frac{1}{2} h_i (-\frac{g_i}{h_i} - T(x_i, \Theta))^2,$$

missä

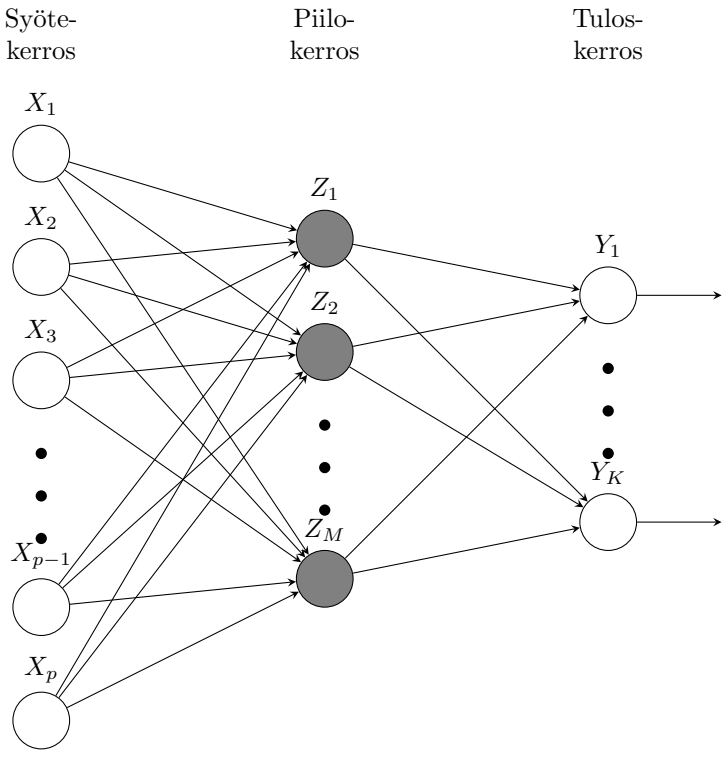
$$h_i = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f=f_{m-1}}, i = 1, ..., N$$

eli tappiofunktion toinen derivaatta funktion f suhteen.

Gradient tree boostingissa ideana on muodostaa useasta pienemmästä päätöspuusta koostuva malli. Tällöin, jos jokaisen malliin lisättävän uuden puun koon annetaan määräytyä rajatta, tulee päätöspuista liian suuria ja ylisovittamisen varaa jokaisen uuden puun kohdalla lisääntyy. Tätä voidaan ehkäistä rajoittamalla jokaisen uuden puun kokoa mallissa esimerkiksi määrittämällä puun lehtien määrä vakioksi. Tällöin malli pyrkii parantamaan ennustetta lisäämällä malliin uusia puita verrattuna yksittäisen puun koon kasvattamiseen. Yksittäisen puun lehtien määrästä ja puiden määrästä tulee siten mallin hyperparametreja. [5, 6]

2.4 Neuroverkot

Neuroverkot ovat saaneet osakseen suuren määrän "hypeä", mikä on saanut ne vaikuttamaan ihmeellisemmiltä kuin todellisuudessa ovat. Tosiasiassa ne ovat, kuten muutkin koneoppimisen mallit, vain epälineaarisia tilastollisia malleja. [6] Osa hypestä liittynee aiemmin esitettyyn analogiaan neuroverkkojen ja ihmisaivojen toiminnan välillä. Neuroverkkojen näkeminen tavallisten lineaaristen mallien luonnollisena yleistyksenä hälventänee osittain tätä mystiikkaa. Neuroverkkomalli voidaan esittää kuvassa 4 havainnollistetun verkkokaavion avulla. Siinä on kuvattu tyypillinen yhden piilokerroksen eteenpäinsyöttävä neuroverkko, jossa kaikki edellisen kerroksen neuronit on kytketty kaikkiin seuraavan kerroksen neuroneihin.

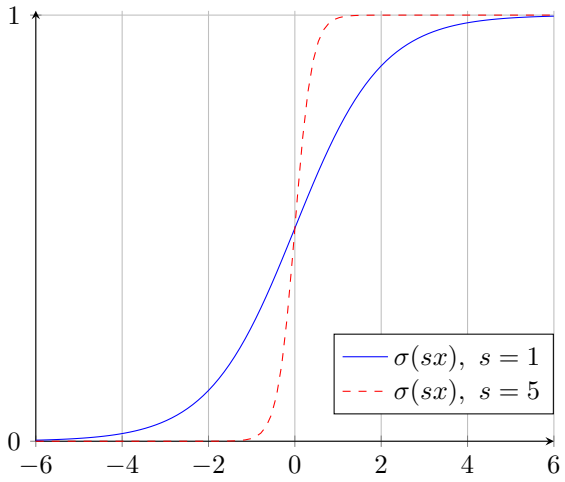


Kuva 4: Havainnollistus neuroverkosta

Kuviossa $X_1, ..., X_p$ ovat mallin syötemuuttujia (features), kun taas $Z_1, ..., Z_M$ ovat muokattuja syötemuuttujia (derived features). Ne saadaan syötemuuttujien lineaaristen kombinaatioiden funktiona. Tulosmuuttujat $Y_1, ..., Y_K$ taas saadaan näiden lineaarisena kombinaationa. Eli

$$Z_m = \sigma(\alpha_{m0} + \alpha_M^T X),\ m = 1, ..., M,$$
$$Y_k = \beta_{k0} + \beta_k^T Z,\ k = 1, ..., K,$$

missä X ja Z ovat syötemuuttujien ja muokattujen syötemuuttujien vektorit. Tässä σ on aktivointifunktio, joka määrittää lähteekö neuronista signaali (ja minkä vahvuisena) seuraavaan neuroniin. Perinteisesti aktivointifunktiona käytettiin kovaa aktivointia, kuten kynnysfunktiota, jolloin neuronista joko lähti signaali tai ei. Hyödynnettäessä neuroverkkoja tilastolliseen mallinnukseen tämä ei ole kuitenkaan optimaalinen, jolloin sileämpi funktio toimii paremmin. Tavallinen valinta aktivointifunktioksi on sigmoidifunktio $\sigma = 1/(1 + e^{-x})$, joka on kuvattu kuvassa 5. [6]



Kuva 5: Havainnollistus aktivointifunktiosta

Kuviosta nähdään, että skaalausparametri s määrittää aktivoinnin kovuuden. Jos aktivointifunktioksi valitaan identiteettifunktio, vastaa malli tavallista lineaarista mallia, mikä selkeyttää kuvaa neuroverkoista lineaaristen mallien yleistyksenä. Aktivoinnin taso riippuu myös painovektorin α normista. Jos tämä normi on pieni, operoi malli aktivointifunktion lineaarisella osalla. Rajoittamalla painojen kokoa lasso- tai ridge-regression tapaan supistuu malli siten kohti lineaarista mallia. Lopulliset tulosmuuttujat P_k saadaan yleensä tulosvektorin Y funktiona:

$$P_k = f_k(X) = g_k(Y),\ k = 1, ..., K,$$

jossa regressiomallin tapauksessa g on useimmiten identiteettifunktio, kun taas klassifioinnin tapauksessa funktioksi g valitaan usein softmax-funktio:

$$P_k = g_k(Y) = \frac{e^{Y_k}}{\sum_{i=1}^{K} e^{Y_i}}.$$

Lopullinen tulosmuuttuja kullekin kategorialle k on siten luku 0 ja 1 välillä ja niiden summa on 1. [6]

Nämä kuvaavat, kuinka vahvasti malli uskoo havainnon kuuluvan kyseiseen kategoriaan. Ne voidaan nähdä todennäköisyyksien kaltaisina pseudotodennäköisyyksinä, mutta toisin kuin esimerkiksi logistisessa klassifioinnissa, ne eivät suoraan vastaa todennäköisyyttä kuulua kyseiseen kategoriaan. Mallin kalibroinnista ja muista ominaisuuksista riippuu, kuinka lähellä nämä ovat todellisia todennäköisyyksiä. Mallin sovelluskohteesta taas riippuu, kuinka tarkasti niiden halutaan kuvaavan todennäköisyyksiä ja kuinka paljon painotetaan muita mallin hyvyyden mittareita.

Neuroverkko voidaan jakaa kolmeen osaan; syötekerrokseen (input layer), piilokerrokseen (hidden layer) ja tuloskerrokseen (output layer). Kuvion palloja kutsutaan neuroneiksi tai solmuiksi (node). Niiden välisten nuolten voidaan ajatella kuvaavan edellisen kerroksen syötteen saamia painoja. Neuronit taas sisältävät aktivointifunktion, jonka perusteella edellisen kerroksen muuttujien lineaarisena kombinaationa saatu syöte lähetetään eteenpäin seuraavaan kerrokseen. Mallissa voi olla useampia piilokerroksia. Piilokerrosten määrällä sekä kuhunkin piilokerrokseen sisältyvällä neuronien määrällä

voidaan siten kontrolloida mallin kompleksisuutta, aivan kuten puiden määrällä ja yksittäisen puun koolla edellä kuvatuissa gradient boosted tree -malleissa.

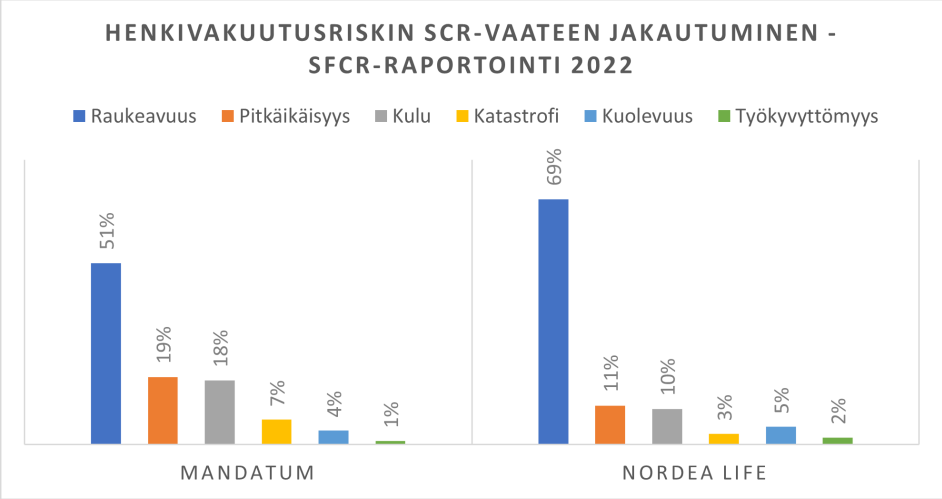
Neuroverkko voi olla myös takaisinkytketty, jolloin malli käyttää informaatioita nykyisen syötteen lisäksi myös aiemmista syötteistä. Tällaiset mallit sopivatkin paremmin esimerkiksi kielen ja tekstin käsittelyyn, eivätkä siten ole tämän työn kohteena.

3 Henkivakuutusten raukeamisriski ja siihen vaikuttavat tekijät

3.1 Yleistä henkivakuutusten raukeamisriskistä

Raukeamisriski on erilaisten markkinariskien jälkeen suurin henkivakuutusyhtiön kohtaamista riskeistä ja sillä on ilmeinen vaikutus henkivakuutusyhtiön kannattavuuteen, vakavaraisuuteen ja likviditeettiin [1, 11]. Raukeaminen on myös yksi Solvenssi II delegoidun asetuksen (2015/35) 26 artiklassa määritellyistä sopimusperusteisista optioista, joka tulee huomioida yhtiön pääomavaateen laskennassa [9]. Verrattuna muihin henkivakuutusriskeihin, kuten pitkäikäisyyteen, kuolevuuteen ja kuluriskiin, se muodostaa yleensä suurimman osan henkivakuutusyhtiön pääomavaateesta.

Alla onkin kuvattu kahden suomalaisen henkivakuutusyhtiön henkivakuutusriskin pääomavaateen jakautuminen komponentteihin. Nähdään, että raukamisriski muodostaa kummankin yhtiön kohdalla yli puolet henkivakuutusriskistä ja onkin selvästi suurempi kuin varsinaiset vakuutetun elämään ja kuolemaan liittyvät pitkäikäisyys-, kuolevuus- ja katastrofiriski.



Kuva 6: Kahden suomalaisen henkivakuutusyhtiön henkivakuutusriskin pääomavaateen (SCR) jakautuminen komponentteihin vuoden 2022 Vakavaraisuutta ja taloudellista tilaa koskevan raportoinnin (SFCR) perusteella [10], [12]

Vakuutusyhtiö odottaa saavansa henkivakuutuksistaan kassavirtoja pitkälle tulevaisuuteen saakka. Tällöin asiakkaiden lopettaessa vakuutusmaksujen maksamisen tai irtisanoessa vakuutuksensa odotettua useammin, jää osa näistä odotetuista kassavirroista saamatta [11]. Etenkin, jos raukeaminen tapahtuu samanaikaisesti usean eri vakuutetun kohdalla (ns. massaraukeaminen), voi vakuutusyhtiö joutua vaikeuksiin ja tämän vuoksi eri raukeamiseen vaikuttavien tekijöiden tunteminen on tärkeää [1].

Raukeamisriski on asiakaskäyttäytymisriski, minkä vuoksi sen taustalla vaikuttavat päätösmekanismit voivat olla hyvinkin kompleksisia ja epälineaarisia. Tutkimuksissa on havaittu, että asiakkailla on jonkinlainen omasta riskistään ja riskipreferensseistään riippuva raja-arvo, oli se sitten rationaalinen tai ei, jonka ylitys saa asiakkaan päättämään vakuutuksensa. Esimerkiksi huonossa taloudellisessa tilanteessa tämän raja-arvon ylittyminen on todennäköisempää. Tämä raukeamisriskin epälineaarisuus tekee siitä mielenkiintoisen kohteen koneoppimismenetelmien soveltamiselle. Raukeamisriski on myös luontevasti esitettävissä koneoppimisongelmana. Ennustettava muuttuja (raukeaa

/ ei raukea) on binäärinen, mikä tekee ennustamisesta selkeän klassifiointiongelman. Tieteellisessä tutkimuksessa on tunnistettu useita raukeamisriskiin potentiaalisesti vaikuttavia tekijöitä, jotka on mahdollista esittää mallin selittävinä muuttujina [1].

3.2 Henkivakuutuksen raukeamisriskiin vaikuttavat tekijät

Raukemisriskiin vaikuttavat tekijät on tieteellisessä tutkimuksessa jaettu usein mikro- ja makrotason tekijöihin. Mikrotason tekijöitä ovat esimerkiksi erilaiset asiakkaaseen ja vakuutussopimukseen liittyvät tekijät. Näitä ovat esimerkiksi vakuutetun ikä ja sukupuoli, vakuutussopimuksen hinta ja sen muutokset, sopimuksen ehdot sekä vakuutetun taloudellinen tilanne [11].

Markotason tutkimuksissa on yleensä keskitytty kahteen keskeiseen hypoteesiin. Hätärahoitushypoteesin mukaan vakuutetut ovat herkempiä irtisanomaan henkivakuutuksiaan taloudellisen epävarmuuden aikoina. Esimerkiksi työttömyyden lisääntyessä asiakkaat saattavat joutua vapauttamaan varoja kaikista saatavilla olevista lähteistä, kuten henkivakuutuksistaan.

Toinen hypoteesi on korkohypoteesi, jonka mukaan korkotason noustessa on markkinoilta saatavissa aiempaa korkeampaa tuottoa ja siten parempia vaihtoehtoisia sijoitustuotteita, mikä saa asiakkaat irtisanomaan olemassa olevia vakuutuksiaan. Makroekonomisten tekijöiden vaikutus kohdistuu useaan vakuutuksenottajaan samanaikaisesti, minkä vuoksi niistä on mahdollista löytää selitysvoimaa massaraukeamiselle eli tilanteelle, jossa suuri osa vakuutuskannasta raukeaa samanaikaisesti. Massaraukeaminen on merkittävä riski vakuutusyhtiölle ja voi aiheuttaa likviditeettihaasteita tai pahimmillaan uhata yhtiön vakavaraisuutta. [15]

Aiemmassa tutkimuksessa on löydetty jonkin verran tukea sekä hätärahoitus- että korkohypoteesille. Näistä etenkin hätärahoitushypoteesi ja työttömyyden vaikutus raukeamiseen on saanut vahvistusta. Mikrotason selittävistä tekijöistä selkein vaikutus on havaittu vakuutetun iällä ja sopimuksen kestolla sekä vakuutusmaksulla. [1, 2]

Esimerkiksi Loisel ym. [11] tutkivat raukeamista kahdella eri koneoppimismenetelmällä, XGBoostilla ja tukivektorikoneella (Support vector machine). Heidän tutkimuksensa keskittyi mikrotason muuttujiin. Heidän tulostensa perusteella koneoppimismallit suoriutuivat raukeamisen ennustamisessa perinteisempiä malleja, kuten logistista regressiota, paremmin. Kun mittarina käytettiin tutkijoiden kehittämää estimaattia raukeamisen ennustamisesta ja asiakaspoistuman hallinnasta saatavasta rahallisesta hyödystä, pärjäsi etenkin XGBoost selvästi muita malleja paremmin. Tutkimuksessa keskityttiin mikrotason muuttujiin. Loisel ym. eivät kuitenkaan raportoineet, mitkä muuttujat selittivät raukeamista parhaiten. [11]

Myös Andrade & Valencia [2] vertailivat koneoppimismalleja raukeamisen ennustamisessa. Hekin keskittyivät työssään mikrotason muuttujiin. Myös heidän tutkimuksensa tuloksena oli, että koneoppimismalleilla pystyttiin selittämään raukeamista perinteisempiä malleja paremmin. Vakuutusmaksun ja vakuutetun iän lisäksi heidän tutkimuksessaan myös jakelukanavalla, maksufrekvenssillä ja myyntipaikkakunnalla havaittiin olevan vaikutusta raukeavuuteen, kun taas esimerkiksi sukupuolella ei havaittu olevan vaikutusta. [2]

Aleandri [1] tutki eri muuttujien vaikutusta raukeamiseen ja ennustetun raukeavuuden vaikutusta kannattavuuteen. Tutkimuksessa vertailtiin koneoppimismenetelmiä logistiseen regressioon. Tutkimuksessa päätöspuupohjainen koneoppimismalli pääsi logistista regressiota selkeästi pienempään ennustevirheeseen. Raukeamiseen vaikuttavista tekijöistä tutkimuksen perusteella merkittävimpiä olivat muun muassa ikä ja vakuutusmaksu. Iän ja raukeamisen välillä todettiin positiivinen korrelaatio, kuten myös vakuutusmaksun ja raukeamisen välillä, mikä on looginen tulos. [1]

Raukeamisriski koskee niin riskihenkivakuutuksia, joissa korvaus maksetaan vakuutetun kuollessa vakuutuksen voimassaoloaikana, kuin säästöhenkivakuutuksia, joissa vakuutettu saa korvausta, jos hän on hengissä sopimuksessa määriteltynä aikana. Tässä työssä keskitytään riskihenkivakuutusten raukeamisriskiin. Eri tekijät saattavat vaikuttaa erilaisiin henkivakuutustuotteisiin eri tavoin [8]. Esimerkiksi erilaiset makrotalouden muutokset, kuten korkotason nousu, saattavat vaikuttaa eri tavalla riskihenkivakuutusten kuin korkotuottoisten tai sijoitussidonnaisten säästöhenkivakuutusten raukeavuuteen, mikä tulee huomioida vertailtaessa tuloksia aiempiin tutkimuksiin.

Tässä työssä tutkitaan sekä mikrotason että makrotason muuttujia. Päähuomio näistä keskittyy mikrotason muuttujiin. Mallinnuksessa käytettyjen muuttujien valinta ja mallinnusprosessin kulku on kuvattu luvussa 4.

4 Raukeamisriskin ennustaminen koneoppimismenetelmillä käytännössä

4.1 Mallinnusprosessin kuvaus

Seuraavassa kuvataan koneoppimismallien soveltaminen riskihenkivakuutusten raukeamisen ennustamiseen käytännössä. Tarkoituksena on testata kuinka hyvin koneoppimismalleilla voidaan ennustaa raukeavuutta, tuovatko koneoppimismenetelmät lisäarvoa verrattuna perinteisempiin menetelmiin suhteessa lisättyyn kompleksisuuteen, mikä kuvatuista malleista toimii parhaiten ja lisäksi mitkä tekijät ennustavat raukeavuutta parhaiten. Työssä käytettiin suomalaisen henkivakuutusyhtiön riskihenkivakuutusdataa.

Mallinnuksessa käytetyt koneoppimismallit olivat gradient boosting (XGBoost) ja neuroverkot. Näitä vertailtiin logistisen regressiomallin antamiin tuloksiin. Mallinnus tehtiin Pythonilla, käyttäen xgboost (gradient boosting), tensorflow (neuroverkot), scikit-learn (logistinen regressio ja suoritusmittarit) ja SHAP (mallien selitys) -kirjastoja Spyder-ympäristössä. Mallinnuksessa pyrittiin löytämään jokaiselle mallille parhaat mahdolliset hyperparametrit. Tähän käytettiin apuna Grid search -menetelmää ja ristiinvalidointia.

Mallinnus jakaantui seuraaviin vaiheisiin, jotka toteutettiin kohtien 3.-6. erikseen XGBoost- ja neuroverkkomalleille sekä logistiselle regressiolle:

1. Datan ominaisuuksien tarkastelu ja datan validointi
2. Muuttujien muokkaaminen mallinnuksen kannalta optimaalisemmiksi (feature engineering)
3. Optimaalisen muuttujien määrän valinta
4. Mallin hyperparametrien optimointi
5. Mallin sovittaminen
6. Mallin tulosten arviointi

Näitä vaiheita toistettiin osittain iteratiivisesti siten, että esimerkiksi alustavien tulosten jälkeen palattiin aiempiin vaiheisiin ja esimerkiksi muokkaamaan muuttujia paremmin mallinnukseen sopivaksi. Tärkeää tässä on kuitenkin noudattaa jossain määrin yllä olevaa järjestystä, sillä muuttujien ja niiden määrän sekä hyperparametrien muokkaaminen liiaksi tulosten perusteella voi johtaa ylisovittamiseen. Jotta esimerkiksi ilmeisimmät virheet datassa saadaan eliminoitua ja malli saadaan ylipäätään toimimaan suhteellisen järkevillä lähtöasetuksilla, on tällainen iteratiivinen tapa kuitenkin perusteltu.

4.2 Datan ominaisuuksien tarkastelu ja muuttujien muokkaaminen

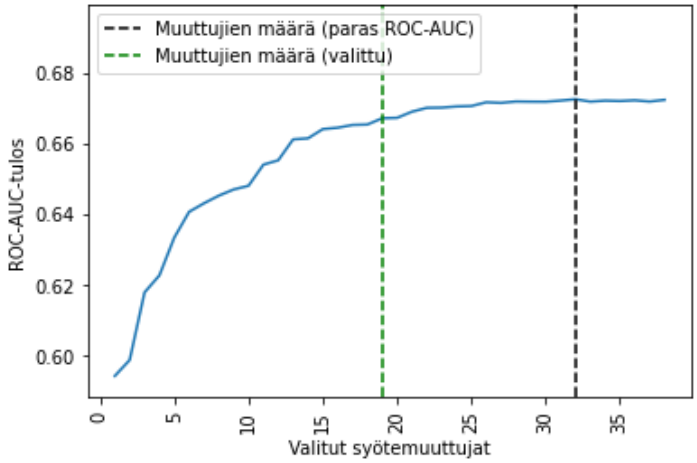
Mallinnus aloitettiin tarkastelemalla datan ominaisuuksia ja tekemällä alustavaa muuttujien muokkausta (feature engineering), jotta ne toimisivat mahdollisimman hyvin mallinnuksen kannalta. Muuttujia tarkasteltiin niiden jakaumien avulla, minkä perusteella pystyttiin varmistamaan, että mallinnukseen käytettävä data näyttää järkevältä ja korjaamaan datassa esiintyviä virheitä. Jakaumien avulla pystyttiin myös varmistamaan, että jokaisessa kategoriassa on mallinnuksen kannalta riittävä määrä havaintoja. Muuttujia pyrittiin tämän perusteella myös pyöristämään esimerkiksi jättämällä maksuista ja niiden muutoksista pois turhat desimaalit. Kategoriamuuttujat, kuten asuinpaikka, muunnettiin dummy-muuttujiksi, jotka saavat arvon 1 tai 0 (esim. pääkaupunkiseutu $\in \{0, 1\}$).

Raukeamisen ennustamiseen käytetyt muuttujat voidaan jakaa asiakkuuteen, tuotteeseen, maksuihin ja vakuutettuun liittyviin muuttujiin sekä makromuuttujiin. Asiakkuuteen liittyviä muuttujia olivat esimerkiksi vakuutuksen kesto, tuoteprofiili ja myyntikanava. Tuotteeseen liittyviä muuttujia esimerkiksi tuotteen tyyppi. Maksuihin liittyviä muuttujia olivat vakuutusmaksut ja niiden muutokset. Vakuutettuun liittyviä muuttujia olivat esimerkiksi vakuutetun ikä, asuinpaikka ja erilaiset elämäntilannetta kuvaavat muuttujat. Makromuuttujista mallinnuksessa käytettiin työttömyysastetta ja korkotasoa sekä näiden muutoksia. Datan sensitiivisyyden takia tulosten osalta muuttujien nimet on piilotettu ja eri muuttujiin viitataan ainoastaan tyylillä Muuttuja 1, Muuttuja 2 ja niin edelleen.

4.3 Muuttujien valinta ja hyperparametrien optimointi

Seuraavaksi tehtiin lopullisessa mallissa käytettävien muuttujien valinta. Tähän käytettiin apuna rekursiivista muuttujien eliminointia ristiinvalidoinnilla (RFECV). Siinä muuttujat järjestetään niiden tärkeyden (feature importance) perusteella. Ensin malli sovitetaan käyttämällä kaikkia muuttujia. Sen jälkeen muuttujia poistetaan yksi kerrallaan niin, että vähiten tärkeät muuttujat poistetaan ensin. Mallin tuloksia arvioidaan jonkin valitun mittarin avulla. Tässä työssä on käytetty arviointimittarina ROC-AUC -pistettä (Receiver Operating Characteristic - Area Under Curve), joka on selitetty myöhemmin luvussa 4.4.2. Muuttujia poistetaan mallista niin kauan kun valitun mittarin antama tulos ei olennaisesti heikkene.

Mallin sovittamisen käytetään jokaisen muuttujien kombinaation osalta ristiinvalidointia eli opetusdata jaetaan osiin, joista yksi osa kerrallaan valitaan validointidataksi ja loppuosaa opetusdatasta käytetään mallin sovittamiseen. Esimerkiksi kymmenkertaisessa ristiinvalidoinnissa sovittaminen tehdään kullekin muuttujien kombinaatiolle kymmenen kertaa. Tällöin saadaan luotettavampi arvio siitä, mikä kombinaatioista toimii parhaiten uudelle datalle ja vältetään ylisovittamista. Tässä työssä käytettiin kolminkertaista ristiinvalidointia.



Kuva 7: Rekursiivinen muuttujien eliminointi ristiinvalidoinnilla (RFECV)

Kuvassa 7 on kuvattu XGBoost-mallin osalta optimaalisen muuttujien määrän valinta. Nähdään, että aluksi mallin tulos paranee muuttujien määrän vähentyessä ja laskee melko jyrkästi kun muuttujien määrää vähennetään alle kymmeneen. Vähentämällä muuttujien määrää voidaan vähentää vähemmän tärkeiden muuttujien malliin tuomaa kohinaa ja ylisovittamista sekä muuttujien välistä multikollineaarisuutta. Muuttujien määrän valinta onkin tasapainoilua niiden tuoman lisäselitysvoiman sekä ylisovittamisen ja mahdollisen multikollineaarisuuden tuomien ongelmien välillä. Pienemmällä muuttujien määrällä mallin tulokset ovat myös läpinäkyvämpiä ja riippuvuussuhteita on helpompi arvioida ja selittää jälkikäteen. Isommalla muuttujien määrällä malli menee kohti “musta laatikko -mallia”, jossa on vaikeaa tai mahdotonta todeta, miksi malli on päätynyt kyseiseen tulokseen.

Kuvasta 7 nähdään, että parhaan ROC-AUC -tuloksen antava muuttujien määrä olisi 32. Tulos ei kuitenkaan olennaisesti heikkene, vaikka muuttujien määrä pudotettaisiin noin kahteenkymmeneen. Alustavan mallinnuksen perusteella osa muuttujista ei tuottanut selkeästi lisäarvoa lopputulokseen nähden. Ylisovittamisen välttämiseksi on myös parempi suosia yksinkertaisempaa mallia silloin, kun päästään suunnilleen samaan lopputulokseen. Tämän vuoksi mallinnukseen valittu muuttujien määrä on 19.

Ongelmana optimaalisen muuttujien määrän valinnassa on se, että mallin sovittaminen vaati arvojen valitsemista mallin hyperparametreille. On mahdollista, että jokin muuttujien määrä tai jotkin muuttujat toimivat paremmin toisilla hyperparametrien asetuksilla. Tällöin valitsemalla ensin optimaaliset muuttujat ja sen jälkeen optimoimalla hyperparametrit ei välttämättä päädytä optimaaliseen muuttujien ja hyperparametrien kombinaatioon. Kaikkien hyperparametrien ja muuttujien kombinaatioiden testaaminen ja ristiivalidointi tekisi sovittamisesta kuitenkin liian hidasta.

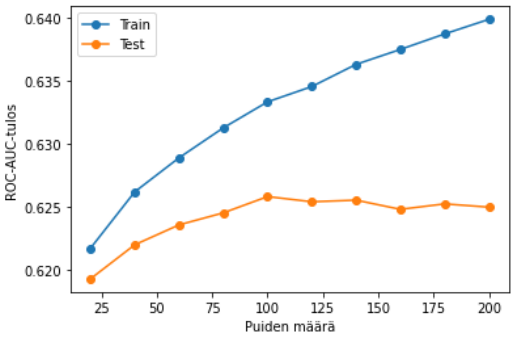
Tämän vuoksi käytettiin iteraatiivista lähestymistapaa, jossa pyrittiin löytämään riittävän hyvät hyperparametrit mallille ja sen jälkeen palattiin muuttujien valintaan. Lopullisten hyperparametrien valinnan jälkeen varmistettiin myös, että valitut hyperparametrit johtivat samaan optimaalisten muuttujien kombinaatioon ja näin pyrittiin varmistamaan, että optimaalisten muuttujien valinta pysyy samana myös hyperparamterien optimoinnin jälkeen.

Hyperparametrien optimoinnissa hyödynnettiin Grid Search -menetelmää kolminkertaisella ristiinvalidoinnilla. Siinä valitaan jokaiselle halutulle hyperparametrille testattavat arvot, jonka jälkeen malli sovitetaan ja ristiinvalidoidaan jokaiselle näiden parametrien kombinaatiolle. XGBoost-mallissa tärkeimpiä hyperparametreja ovat puiden määrä ja puun syvyys. Muita optimoinnissa käytettyjä hyperparametreja olivat learning rate, subsample, colsample bytree, min child weight ja luokkien paino. Neuroverkkojen osalta tärkeimpiä hyperparametreja ovat piilokerrosten ja neuronien määrät, läpikäyntien (epoch) määrä, erien (batch) koko, learning rate sekä optimointimenetelmä.

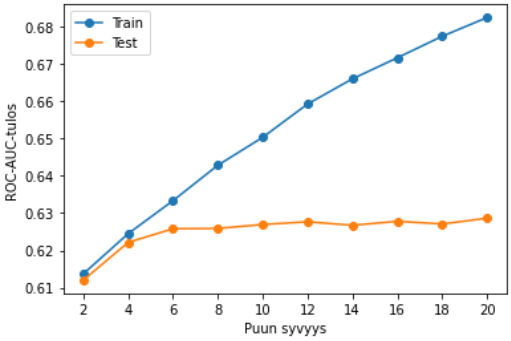
ROCAUC		Puun syvyys					
		4	6	8	12	16	20
Puiden määrä	20	0.6576	0.6635	0.6656	0.6675	0.6680	0.6673
	40	0.6621	0.6664	0.6676	0.6692	0.6690	0.6685
	80	0.6647	0.6682	0.6690	0.6694	0.6694	0.6688
	160	0.6667	0.6692	0.6693	0.6685	0.6683	0.6673
	320	0.6677	0.6693	0.6690	0.6674	0.6666	0.6649
	640	0.6681	0.6686	0.6677	0.6650	0.6642	0.6629

Taulukko 1: XGBoost-mallin parametrien optimointi

Taulukossa 1 on kuvattu XGBoost-mallin suoriutuminen eri puiden määrillä ja puun syvyyksillä. Taulukosta nähdään, että puiden määrän tai syvyyden kasvattaminen parantaa mallinnuksen tuloksia. Pienemmällä puiden määrällä vaaditaan suurempi puun syvyys, jotta päästään yhtä hyvään tulokseen ja päin vastoin. Riittävällä puiden määrällä puun syyvyden kasvattaminen ei tietyn rajan jälkeen enää paranna mallin tulosta. Puiden määrän kasvattaminen ei myöskään tietyn rajan jälkeen paranna tulosta enää merkittävästi.



(a) Opetus vs. testidata eri puiden määrillä (puun syvyys = 6)



(b) Opetus vs. testidata eri puun syvyyksillä (puiden määrä = 100)

Kuva 8: XGBoost - opetus vs. testidata

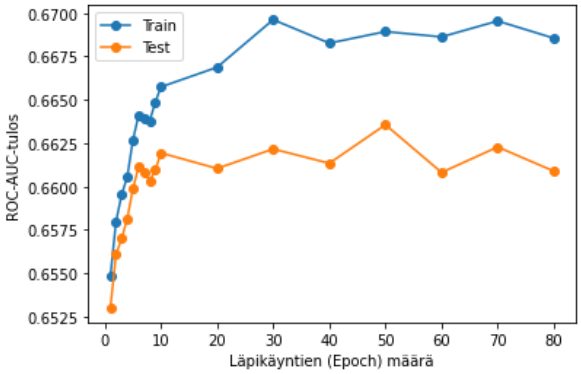
Parametri	**Arvo**
Erän (batch) koko	64
Learning rate	0.001
Piilokerrosten määrä	2
Neuronien määrä piilokerroksissa	(32,16)

Taulukko 2: Neuroverkkomalli - valitut parametrit

Kuvassa 8 on tarkasteltu mallin suoriutumista opetus- ja testidatassa, kun puiden määrää tai syvyyttä kasvatetaan. Nähdään, että molemmissa tapauksissa tietyn rajan jälkeen mallin tulos testida-

tassa ei enää merkittävästi parane tai jopa laskee, mutta opetusdatassa tulos jatkaa paranemistaan, kunnes malli ennustaa oikein käytännössä jokaisen opetusdatan havainnon. Tällöin malli on rankasti ylisovitettu eli se toimii erittäin hyvin opetusdatalle, mutta sillä ei ole samaa ennustevoimaa uudelle datalle. Tämän välttämiseksi hyperparametrit tulee valita siten, ettei turhaan kasvateta mallin kompleksisuutta, jollei se merkittävästi paranna mallin ennustekykyä uudelle datalle eli niin, että opetus- ja testidatan ero ei kasva liian suureksi. Tämän tarkastelun perusteella XBoost-mallin parametreiksi valittiin puun syvyydeksi 6 ja puiden määräksi 100.

Myös neuroverkkomallille toteutettiin hyperparametrien optimointi Grid Search menetelmää käyttäen. Optimoitavat parametrit olivat erän (batch) koko, learning rate, piilokerrosten määrä ja neuronien määrä kussakin kerroksessa. Läpikäyntien määrän osalta käytettiin aikaista pysäytystä (early stopping), jolloin mallin opetus pysäytetään, kun uudet läpikäynnit eivät enää paranna mallin tulosta. Grid Search menetelmällä valitut neuroverkkomallin parametrit on annettu taulukossa 2. Kuvassa 9 on näytetty, miten neuroverkkomallin tulos kehittyy läpikäyntien määrän kasvaessa. Nähdään, että alkuun läpikäyntien lisääminen parantaa mallin tulosta, mutta noin kymmenen läpikäynnin kohdalla tulos ei enää olennaisesti parane ja opetus- ja testidatan välinen ero kasvaa.



Kuva 9: Neuroverkko - opetus vs. testidata eri läpikäyntien (epoch) määrillä

4.4 Mallinnuksen tulokset

4.4.1 Selitettävä tekoäly ja Shapleyn arvot

Eräs koneoppimismallien ongelma verrattuna perinteisempiin malleihin on se, että niiden tulkittavuus on perinteisempiä malleja vaikeampaa. Tulosten tulkittavuus ja selitettävyys on kuitenkin tärkeää mallien hyödyntämisen kannalta. Päätöksenteossa on usein tärkeää tietää, miksi tiettyyn lopputulokseen päädytään. Yksi tekijä mallien selitettävyyden kannalta on se, miten kukin mallissa käytetyistä muuttujista vaikuttaa mallin lopputulokseen. Perinteisemmissä regressiomalleissa tämä on selkeämpää ja onnistuu tutkimalla kutakin muuttujaa vastaavia kertoimia. Koneoppimismalleissa taas muuttujien vaikutussuhteet mallin lopputulokseen ovat kompleksisempia, eikä vastaavien kertoimien määrittäminen ole yhtä suoraviivaista. Paljon viimeaikista tutkimusta onkin keskittynyt kehittämään koneoppimismalleihin soveltuvia vaikutussuhteiden arviointimenetelmiä. Yksi suosiota saavuttanu menetelmä on peliteoriaan perustuva Shapley-arvojen hyödyntäminen. [13]

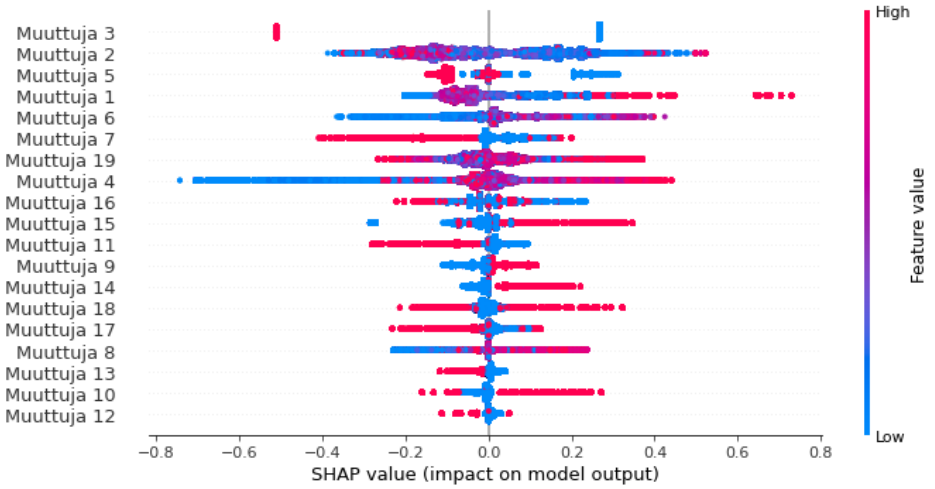
Shapley-arvojen käyttäminen pohjautuu peliteoriaan ja siinä mallin muuttujat vastaavat pelaajia, jotka “liittyvät peliin”. Yksi Shapley-arvojen ominaisuuksista one se, että niiden summa yhden havainnon kohdalla vastaa aina erotusta lähtötilanteen (ilman yhtään muuttujaa) ja lopputuloksen (kaikki muuttujat mukana) välillä. Tällöin on mahdollista havainnoida yksittäisen havainnon osalta, millä muuttujalla oli mikäkin vaikutus lopputulokseen. Hyödyntämällä keskiarvoja ja piirtämällä kuvioita kaikkien havaintojen osalta voidaan havainnoida kunkin muuttujan vaikutusta kaikkien havaintojen osalta.

Shapley-arvoilla on kuitenkin myös ongelmansa. Riippuen siitä, miten peiliteoreettinen viitekehys on implementoitu kuhunkin Shapley-arvoja hyödyntävään sovellukseen, saattaa tulosten välillä olla merkittäviä eroja. On esimerkiksi mahdollista, että jokin muuttuja vaikuttaa Shapley-arvojen perusteella merkittävältä mallin kannalta, vaikka se todellisuudessa olisikin täysin epärelevantti . Tämä

epävarmuus ei välttämättä ilmene, koska Shapley-arvoille ei useinkaan esitetä luottamusvälejä. Lisäksi Shapley-arvojen laskenta vaatii yleensä paljon laskentatehoa. Tästä huolimatta Shapley-arvot ovat yleisesti käytetty metodi koneoppimismallien selittämiseen ja niinpä niitä hyödynnetään tässäkin työssä. [13]

Alla on XGBoost- ja neuroverkkomallien pythonin SHAP-kirjaston avulla laaditut Shapley-kuvaajat. Kuvaajassa ylimpänä oleva muuttuja on mallin kannalta tärkein ja alimpana oleva muuttuja vähiten tärkeä. Vaaka-akselin arvot ovat Shapley-arvoja ja ne kuvaavat kuinka iso vaikutus kyseisellä muuttujalla oli yksittäisen havainnon lopputulokseen. Positiivinen Shapley-arvo tarkoittaa, että kyseinen muuttuja indikoi raukeamista kyseiselle havainnolle. Lopullinen ennuste vastaa Shapley-arvojen summaa lisättynä lähtötasoon (mallin odotusarvo ilman yhtään muuttujaa). Yksittäiset havainnot näkyvät kuvioissa pisteinä ja pisteen väri indikoi kyseisen muuttujan arvoa kyseiselle havainnolle. Punainen väri tarkoittaa, että muuttujan arvo oli korkea, esimerkiksi että kyseisen havainnon kohdalla ikä oli keskimääräistä korkeampi ja sininen arvo matalampaa muuttujan arvoa. Kategoristen (on / ei ole) muuttujien kohdalla punainen tarkoittaa, että kyseisellä havainnolla on kyseinen ominaisuus (muuttujan arvo = 1).

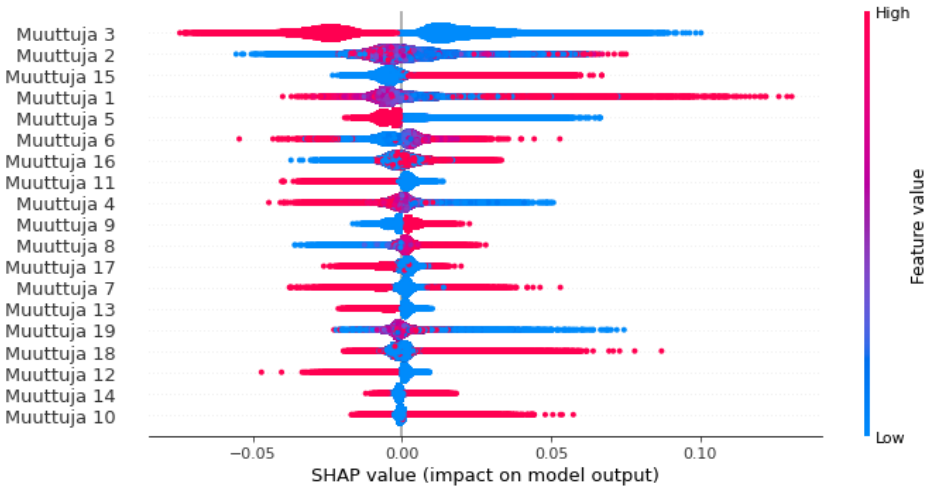
Tulosten tulkinta on suoraviivaisempaa, jos korkeat muuttujan arvot sijoittuvat yhdelle laidalle ja matalat arvot toiselle laidalle. Tällöin voidaan sanoa esimerkiksi, että korkealla iällä oli positiivinen (tai negatiivinen) vaikutus raukeavuuteen. Koneoppimismallien kompleksisuus näkyy siinä, että näin ei kuitenkaan aina ole. Varsinkin, mitä alemmas muuttujissa mennään, sitä useammin korkeita ja matalia muuttujan arvoja on nollan molemmilla puolilla. Tällöin riippuu siis muiden muuttujien arvoista, onko kyseisen muuttujan vaikutus raukeavuuteen positiivinen vai negatiivinen kyseisen havainnon kohdalla. Koneoppimismalleilla onkin mahdollista kuvata sellaisia ilmiöitä, joissa esimerkiksi iällä olisi eri suuntainen vaikutus raukeavuuteen eri asiakasryhmissä. Monesti tällaisia riippuvuussuhteita on kuitenkin vaikea erottaa ylisovittamisesta.



Kuva 10: SHAP-kuvaaja (XGBoost)

Kuvissa 10 ja 11 on kuvattu XGBoost-mallin ja neuroverkkomallin selittävien muuttujien vaikutusta mallin lopputulokseen SHAP-kuvaajan avulla. Nähdään, että muuttujien järjestys malleissa on pitkälti samankaltainen, mutta erojakin löytyy. Osa muuttujista vaikuttaa mallin lopputulokseen suoraviivaisemmin eli siten, että suuremmat muuttujan arvot lisäävät ennustettua raukeavuutta tai päin vastoin. Osa muuttujista vaikuttaa vähemmän suoraviivaisesti esimerkiksi Muuttujan 1 osalta vaikuttaisi, että XGBoost-mallissa kaikkein suurimmat muuttujan arvot lisäävät raukeavuuden todennäköisyyttä, keskisuuret pienentävät sitä hieman ja pienten arvojen vaikutus vaihtelee. Neuroverkkomallin osalta tulos on samankaltainen etenkin keskisuurten arvojen osalta, mutta ääriarvojen osalta vaikutus poikkeaa hieman XGBoost-mallista.

Tällainen mallien antama riippuvuussuhde voi kuvastaa, sitä että kyseisen muuttujan vaikutus raukeavuuteen on epälineaarinen ja riippuu myös muista muuttujista. Toisaalta pahimmillaan monimutkaiset riippuvuussuhteet voivat indokoida sitä, että malli on ylisovittava. Tätä on pyritty kontrolloimaan edellä kuvatuin menetelmin muun muassa ristiivalidoinneilla ja konservatiivisilla hyperpara-



Kuva 11: SHAP-kuvaaja (Neuroverkko)

metrien valinnoilla.

Yleisesti voidaan todeta, että mallien lopputulokseen eniten vaikuttavat muuttujat olivat samankaltaisia kuin aikaisemmissa tutkimuksissa havaitut, mutta erojakin löytyi osin johtuen myös siitä, että eri tutkimuksissa käytetyt muuttujat eroavat jonkin verran toisistaan. Tärkeämpiä mallin kannalta olivat mikrotason muuttujat, kuten asiakkaaseen ja tuotteeseen liittyvät muuttuja, kun taas makrotason muuttujat, eivät toimineet yhtä hyvin. Tämä johtuu todennäköisesti siitä, että datan aikaperiodi on suhteellisen lyhyt ja esimerkiksi työttömyysasteessa ei ajanjaksolla tapahtunut suuria muutoksia. Lisäksi datan ollessa vuosikohtaista oli toimivien makromuuttujien rakentaminen hankalaa.

4.4.2 Sekaannusmatriisi ja suoritusmittarit

Shapley-arvoilla on mahdollista kuvata ja selittää mallin logiikkaa ja riippuvuussuhteita sekä sitä, miten yksittäiseen ennustukseen on päädytty. Se ei kuitenkaan kerro vielä, miten hyvin malli toimii raukeavuuden ennustamisessa. Mallin selityskykyä on mahdollista kuvata useiden eri mittarien sekä esimerkiksi sekaannusmatriisin ja ROC-käyrän (Receiver operating characteristic) avulla.

Taulukossa 3 on kuvattuna sekaannusmatriisin logiikka. Sen perusteella on mahdollista muodostaa yksittäistä mittaria kattavampi kuva mallin toimivuudesta ja toisaalta monet muut mittarit muodostetaan siinä kuvattujen suureiden perusteella. Taulukossa 4 on annettu sekaannusmatriisit käytetyille malleille. Mallit on kalibroitu asettamalla positiivisen (raukeaa) ennusteen kynnysarvo niin, että väärien positiivisten ja väärien negatiivisten osuus on suunnilleen yhtä suuri. Tuloksista nähdään, että käytetyt koneoppimismallit suoriutuivat jonkin verran vertailuna käytettyä logistista regressiota paremmin, sillä oikeiden positiivisten ja oikeiden negatiivisten osuudet ovat kumpikin suurempia koneoppimismallien osalta. XGBoost-malli suoriutui neuroverkkomallia paremmin, mutta vain hieman.

		Ennustettu	
	Populaatio = (P+N)	**Positiivinen**	**Negatiivinen**
Todellinen	**Positiivinen (P)**	TP = Oikeat positiiviset (true positive)	FN = Väärät negatiiviset (false negative)
	Negatiivinen (N)	FP = Väärät positiiviset (false positive)	TN = Oikeat negatiiviset (true negative)

Taulukko 3: Sekaannusmatriisin muodostaminen

Logistinen regressio	
TP: 1.15%	FN: 6.73%
FP: 6.72%	TN: 85.39%

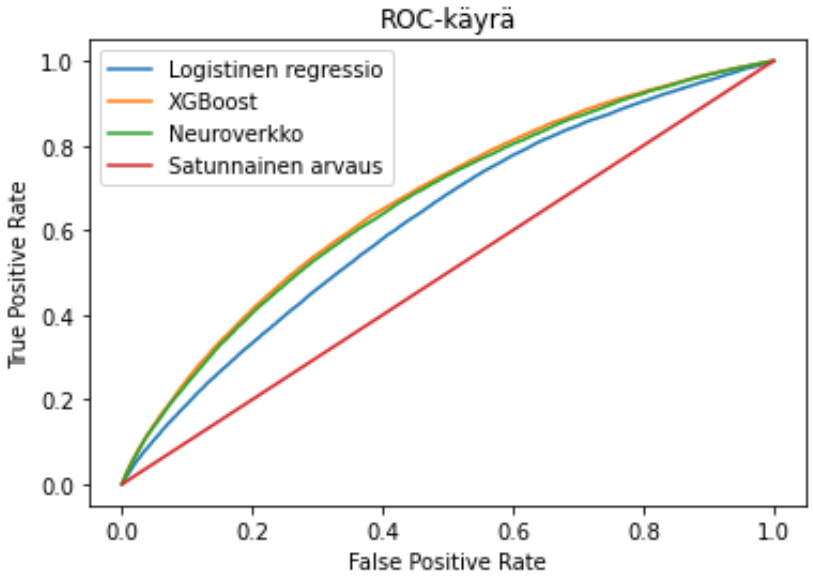
XGBoost	
TP: 1.45%	FN: 6.43%
FP: 6.39%	TN: 85.72%

Neuroverkko	
TP: 1.42%	FN: 6.46%
FP: 6.42%	TN: 85.69%

Taulukko 4: Sekaannusmatriisit eri malleille

Mallien suoriutumista on mahdollista kuvata myös ROC-käyrän avulla. Siinä on kuvattu mallin oikeiden positiivisten (TP) suhteellinen osuus kaikista positiivisista havainnoista (P) ja väärien positiivisten (FP) suhteellinen osuus kaikista negatiivisista havainnoista (N) mallin eri herkkyyden tasoilla. Ennustemalleissa on mahdollista säätää mallin herkkyyttä muuttamalla kynnysarvoa, jolla malli luokittelee havainnot raukeaviksi ja ei-raukeaviksi. Kynnysarvoa muuttaessa malli luokittelee herkemmin havainnot raukeviksi, mutta silloin käytännössä aina kasvaa myös väärin raukeaviksi luokiteltavien havaintojen osuus. Mitä vähemmän väärin luokiteltujen havaintojen osuus kasvaa mallin herkkyyttä lisättäessä, sitä paremmin malli pystyy kuvaamaan tarkasteltavaa ilmiötä.

Kuvassa 12 on kuvattu työssä käytettyjen mallien ROC-käyrät. Punainen käyrä kuvaa satunnaista arvausta vastaavaa ROC-käyrää. Satunnaisen arvauksen kohdalla väärien havaintojen määrä kasvaa samassa suhteessa oikeiden havaintojen kanssa. Täydellisessä mallissa taas käyrä kulkisi mahdollisimman pystysuoraan ylöspäin kohti kuvion vasenta yläkulmaa eli väärien havaintojen määrä ei kasvaisi herkkyyttä muutettaessa. Mitä suurempi käyrän alapuolella oleva pinta-ala on, sitä paremmin malli toimii. Kuviosta nähdään, että kaksi käytettyä koneoppimismallia suoriutuvat selvästi satunnaista arvausta paremmin. Ne toimivat myös paremmin kuin vertailussa käytetty logistinen regressiomalli. Tämän perusteella näyttäisi, että koneoppimismenetelmillä on siten mahdollista lisätä mallin selitysvoimaa perinteisempiin malleihin verrattuna.



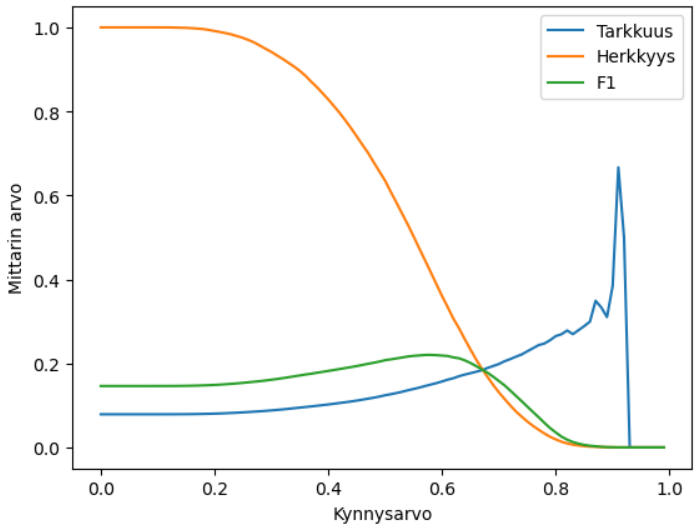
Kuva 12: ROC-käyrät eri malleille

Sekaannusmatriisin pohjalta on mahdollista muodostaa erilaisia mittareita kuvaamaan mallien toimivuutta. Virheettömyys (accuracy) kuvastaa kaikkien oikein ennustettujen havaintojen osuutta kaikista havainnoista $= (TP + TN)/(P + N)$. Tarkkuus (precision) taas kuvastaa oikeiden positiivisten havaintojen osuutta positiiviseksi ennustetuista havainnoista $= TP/(TP+FP)$. Se kertoo siten, kuinka todennäköisesti positiiviseksi (rauenneeksi) ennustettu havainto on todellisuudessa positiivinen (rauennut). Herkkyys (recall) taas kuvastaa oikeiden positiivisten havaintojen osuutta kaikista positiivisista havainnoista $= TP/P$. Se kertoo siten, kuinka suuren osuuden kaikista positiivisista (rauenneista) havainnoista malli pystyy tunnistamaan. Myös ROC-käyrän pohjalta voidaan laskea numeerinen mittari, ROC-AUC-pistearvo, joka lasketaan ROC-käyrän alle jäävänä pinta-alana. Tämä voi siten saada arvon 0 ja 1 välillä, missä arvo 1 vastaisi täydellistä mallia. Arvo 0.5 taas vastaisi puhdasta arvausta.

Mittari	Logistinen regressio	XGBoost	Neuroverkko
Virheettömyys (Accuracy)	86.55 %	87.17 %	87.11 %
Tarkkuus (Precision)	14.66 %	18.49 %	18.13 %
Herkkyys (Recall)	14.65 %	18.39 %	18.04 %
ROC-AUC	0.625	0.670	0.663

Taulukko 5: Mallien tulosten vertailu

Taulukossa 5 on annettu edellä kuvattujen mittareiden tulokset eri mallien osalta. Parhaiten toiminut XGBoost-malli pääsi noin 87 %:n virheettömyyteen. Koska datassa raukeamattomien osuus on niin suuri suhteessa rauenneisiin, olisi mallin virheettömyyttä mahdollista parantaa kasvattamalla negatiivisten ennusteiden osuutta. Parhaaseen virheettömyyteen päästäisiin ennustamalla kaikki havainnot negatiivisiksi. Tällainen malli ei olisi kuitenkaan kovin hyödyllinen. Vastaavasti voitaisiin optimoida mallin tarkkuutta kasvattamalla kynnysarvoa tai herkkyyttä pienentämällä kynnyarvoa. Tätä voidaan kuvata piirtämällä eri suoritusmittarit eri kynnysarvoilla. Kuvassa 13 on piirrettynä XGBoost-mallin tarkkuus, herkkyys ja F_1-arvo, joka kuvasta tarkkuuden ja herkkyyden harmonista keskiarvoa, eri kynnysarvoilla. Nähdään, että kynnysarvon kasvaessa mallin tarkkuus paranee (paitsi kynnysarvon 0.8 jälkeen, jolloin havaintojen määrä putoaa kohti nollaa) ja samalla herkkyys heikkenee. Nämä kaksi mittaria yhdistävä F_1-arvo saavuttaa maksiminsa noin kynnysarvon 0.6 kohdalla. Mallin sovelluskohteesta riippuu, onko tärkeämpää tunnistaa mahdollisimman suuri osa positiivisista havainnoista vai onko tärkeämpää välttää vääriä positiivisia havaintoja.



Kuva 13: XGBoost - suoritusmittarit eri kynnysarvoilla

Eri tutkimuksissa on annettu eri määritelmiä sille, mikä on riittävän hyvä ROC-AUC-pistearvo. Toisissa tutkimuksissa vasta yli 0.7 pistearvo lasketaan hyväksi, kun taas toisissa tutkimuksissa pienempikin pistearvo voi puoltaa mallin hyödyllisyyttä. Tulosten tulkinta riippuukin paljon mallin sovelluskohteesta. Esimerkiksi lääketieteellisissä sovelluksissa vaaditaan usein korkeampi pistearvo, jotta mallia voidaan pitää hyödyllisenä, kun taas toisissa sovelluskohteissa, kuten asiakaspysyvyyden ennustamisessa voisi riittää pienempikin pistearvo. Mallin hyödyllisyys riippuukin siitä, mikä on kustannus siitä, että malli on väärässä ja toisaalta oikeasta ennustuksesta saatava hyöty. [4]

Kaiken kaikkijaan tulosten perusteella koneoppimismallit vaikuttaisivat toimivan perinteisempiä menetelmiä paremmin raukeavuuden ennustamisessa ja käytetyistä malleista XGBoost toimi parhaiten. Tämä on linjassa aiempien tutkimusten kanssa, jossa myös boosting-pohjaisten mallien, on havaittu toimivan paremmin raukeamisen kuvaamisessa verrattuna neuroverkkoihin [3, 7]. Erot näiden kahden koneoppimismallin välillä eivät kuitenkaan olleet merkittäviä. Kuitenkin, kun huomioidaan vaadittava laskentateho ja mallin sovittamiseen käytettävä aika, jotka neuroverkkomallin osalta olivat suurempia, oli XGBoost-malli tarkastelluista menetelmistä paremmin raukeamisen ennustamiseen soveltuva.

XGBoostin etuna on myös jonkin verran neuroverkkoja parempi tulkittavuus. Päätöspuupohjaisten mallien avulla on neuroverkkoja helpommin mahdollista hahmottaa eri muuttujien tärkeys mallin kannalta ja vaikutukset mallin lopputulokseen.

Jäljelle jää vielä kysymys, onko koneoppimismallien käytöstä saatava hyöty suhteessa niiden sovittamiseen kuluvaan aikaan riittävän suuri raukeamisen ennustamisessa ja voitaisiinko mallin tuloksia parantaa optimoimalla mallia paremmin perustuen raukeamisen ennustamisesta saataviin hyötyihin. Kappaleessa 4.4.3 on pyritty optimoimaan XGBoost-mallia perustuen raukeamisen hallinnasta saatavaan taloudelliseen hyötyyn.

4.4.3 Regressiomalli raukeamisen hallinnan taloudelliselle hyödylle

Eräässä aiemmassa tutkimuksessa Loisel ym. käyttivät henkivakuutusten raukeamiseen sovellettujen koneoppimismallien arvioinnissa kehittämäänsä taloudellista mittaria. Siinä oletettiin, että kaikilla vakuutuksilla i on sama tuottoprosentti p suhteessa vakuutusmäärään F_i ja että niille vakuutuksille, joiden ennustetaan raukeavan, myönnetään alennus δ, jonka asiakas hyväksyy todennäköisyydellä γ. Vakuutetun kontaktoimisesta syntyy kiinteä kustannus c. Raukeamassa oleville ja raukeamattomille vakuutuksille on omat raukemisen todennäköisyysvektorinsa r. Lisäksi kassavirrat diskontataan korolla d. [11]

Tämän pohjalta tutkijat pystyivät muodostamaan regressiomallin vakuutetun kontaktoimisesta saatavalle tuotolle. Tällöin sovitettava muuttuja eli kontaktoimisesta saatava tuotto on muotoa:

$$R_i = \begin{cases} -PV(\delta, F_i, \hat{r}, i) - c, & \text{kun } y_i = 0, \\ \gamma PV(p - \delta, F_i, \hat{r}, i) - c, & \text{kun } y_i = 1, \end{cases}$$

missä PV eli tulevien kassavirtojen nykyarvo on muotoa

$$PV(\delta, F_i, r, i) = \sum_{t=0}^{T} \frac{pF_i r_t}{(1+d)^t}.$$

Tämän avulla voidaan koneoppimismalli sovittaa käyttäen tappiofunktiona keskimääräistä neliövirhettä

$$L(R_i, \hat{R}_i) = \frac{1}{N} \sum_{i=1}^{N} [R_i - \hat{R}_i]^2$$

ristiinvalidoinnissa. Lopuksi vakuutuksen ennustetaan raukeavan, kun estimoitu hyöty asiakkaan kontaktoinnista on positiivinen:

$$\hat{y_i} = \begin{cases} 1, & \text{kun } \hat{R}_i > 0, \\ 0, & \text{kun } \hat{R}_i < 0. \end{cases}$$

Tässä y_i tulee tulkita ennemminkin ennusteeksi siitä, onko raukeamisen hallinta kannattavaa, kuin varsinaiseksi ennusteeksi raukeamiselle. Lopulta koko portfolion osalta voidaan siten laskea raukeamisen hallinnasta saatava taloudellinen hyöty (NPV) kaavalla

$$NPV = \gamma PV(p - \delta, F_{(1,1)}, r_0, i) - PV(\delta, F_{(0,1)}, r_0, i) - c(N_{(0,1)} + N_{(1,1)}),$$

missä r_0 on raukeamisvektori raukeamattomille vakuutuksille, $F_{(j,k)}$ on vakuutusmäärä hvainnoille todellisella raukeamisella j ja raukeamisennusteella k, ja $(j,k) \in \{0,1\}^2$ ja vastaavasti $N_{(j,k)}$ kuvastaa kappalemääriä todellisella raukeamisella j ja raukeamisennusteella k. [11]

Tämän aiemman tutkimuksen pohjalta muodostettiin vastaava regressiomalli XGBoost-menetelmälle. Taloudellisen hyödyn mallin parametrit valittiin mahdollisimman hyvin aiempaa tutkimusta vastaaviksi. Näiden parametrien osalta testattiin kahta vaihtoehtoista mallia. Mallissa 1 asiakkaan kontaktoinnin kustannus ja annettava alennus ovat suurempia kuin Mallissa 2. Taulukossa 6 on

Mittari	Logistinen regressio	XGBoost (alkuperäinen malli)	XGBoost-reg (regressiomalli)
Taloudellinen hyöty (malli 1)	-18 716 €	22 734 €	82 377 €
Taloudellinen hyöty (malli 2)	85 380 €	216 877 €	407 395 €

Taulukko 6: Taloudellisen mittarin tulokset

annettu estimaatit raukeamisen hallinnan tuottamalle taloudelliselle hyödylle. Tulokset on esitetty aiemmissa kappaleissa kuvattujen logistisen regression ja XGBoost-mallien sekä yllä kuvatun XGBoost-mallille sovitetun taloudellista hyötyä optimoivan regressiomallin osalta. Tulokset on esitetty erikseen kahden eri taloudellisen hyödyn mallin parametrivalintojen osalta.

Nähdään, että alkuperäinen XGBoost-malli tuottaa logistista regressiota suuremman taloudellisen hyödyn. Tulos paranee, kun malli on optimoitu regressoimalla taloudellista hyötyä puhtaan klassifioinnin sijaan. Lopputulos saatavasta taloudellisesta hyödystä on herkkä taloudellisen hyödyn mallinnuksessa valituille parametreille.

Mallissa 1 raukeamisen hallinnasta saatava hyöty on pienempi suuremman annetun alennuksen ja suuremman kontaktoinnista syntyvän kustannuksen takia. Mallin 1 parametrivalinnoilla taloudellisen hyödyn perusteella optimoidun XGBoost-mallin (XGBoost-reg) tuottama kokonaisraukeavuus on jonkin verran todellista raukeavuutta pienempi, sillä väärän positiivisen ennusteen kustannus on suhteellisen suuri. Näillä parametrivalinnoilla logistinen regressio tuottaa negatiivisen taloudellisen hyödyn, sillä kustannus väärien asiakkaiden kontaktoinnista ja alennuksen antaminen sellaisille asiakkaille, jotka olisivat muutenkin jääneet ylittää onnistuneista asiakaspidoista saadun tuoton. Myös molempien XGBoost-mallien taloudellinen hyöty jää suhteellisen pieneksi.

Mallissa 2 annettu alennus ja kontaktoinnista syntyvä kustannus ovat pienempiä. Tällöin raukeamisen hallinta kannattaa herkemmin. Tässä tapauksessa kaikki mallit tuottavat positiivisen taloudellisen hyödyn ja XGBoost-reg tuottaa selvästi suurimman hyödyn. Ennustettu kokonaisraukeavuus on tässä tapauksessa kuitenkin selvästi todellista raukeavuutta suurempi eli "väärien hälytysten" määrä on selvästi kasvanut. Tämä on luonnolista sillä saatava potentiaalinen hyöty on suurempi ja toisaalta taas väärässä olemisen kustannus pienempi.

Lopulta onkin kiinni siitä, miten kustannustehokkaasti ja vaikuttavasti raukeamista pystytään hallinnoimaan ja toisaalta kuinka kannattava tuote on kyseessä eli minkä verran asiakkaan säilyttäminen hyödyttää yhtiötä. Voidaan kuitenkin todeta, että koneoppimismenetelmin raukeavuuden ennustamista on mahdollista parantaa ja taloudellisen hyödyn perusteella tehtävän optimoinnin avulla tulosta on mahdollista parantaa entisestään.

5 Johtopäätökset ja yhteenveto

Tässä työssä käsiteltiin koneoppimismenetelmiä ja niiden soveltamista henkivakuutusten raukeamisriskin ennustamiseen. Luvussa 2 käytiin läpi koneoppimisen teoriaa ja työssä käytettyjä koneoppimismalleja. Käsiteltäviksi malleiksi valittiin Gradient boosting (XGBoost) ja neuroverkkomallit. Luvussa 3 käytiin läpi henkivakuutusten raukeamisriskiä ja sen merkittävyyttä vakuutusyhtiölle. Käytiin läpi raukeamiseen vaikuttavia tekijöitä sekä aiempia tutkimuksia raukeamisriskin ennustamiseen liittyen.

Luvussa 4 esiteltiin käytännön sovellus, jossa valittuja koneoppimismenetelmiä sovellettiin raukeamisriskin ennustamiseen hyödyntäen suomalaisen henkivakuutusyhtiön dataa. Luvussa kuvattiin mallinnusprosessin kulku muun muassa muuttujien valinta ja hyperparametrien optimointi. Tämän jälkeen esiteltiin mallinnuksen tuloksia. Kuvattiin eri muuttujien vaikutus mallin lopputulokseen Shapley-arvoja hyödyntämällä. Mallien selitysvoimaa kuvattiin sekaannusmatriisin, ROC-käyrän ja erilaisten suoritusmittarien avulla. Lopuksi pyrittiin mittaamaan raukeavuuden hallinnasta saatua taloudellista hyötyä ja optimoimaan koneoppimismallia saadun taloudellisen hyödyn perusteella.

Raukeamisriski on yksi merkittävimmistä henkivakuutusyhtiön kohtaamista riskeistä. Tarkastelemalla eri yhtiöiden vakavaraisuutta ja taloudellista tilaa koskevaa raportointia nähdään, että raukeavuusriski muodostaa suurimman osan henkivakuutusriskin pääomavaateesta. Aiemmissa tutkimuksissa on testattu eri muuttujia raukeamisriskin ennustamisessa. Muuttujat voidaan jakaa makro- ja mikrotason muuttujiin.

Tässä työssä tehdyn mallinnuksen perusteella koneoppimismenetelmillä oli mahdollista parantaa

raukeavuuden ennustamista verrattuna perinteisempiin menetelmiin. Muuttujista paremmin toimivat mikrotason muuttujat. Mallien kannalta tärkeimmät muuttujat olivat pitkälti linjassa aiempaan tutkimukseen ja raukeamisen mallintamisessa yleisesti käytössä oleviin muuttujiin nähden. Käytetyistä koneoppimismalleista parhaiten toimi XGBoost-malli, joka suoriutui kuitenkin vain hieman neuroverkkomallia paremmin. Huomioiden mallien sovittamiseen käytettävä aika ja mallin selitettävyys, oli XGBoost-malli näistä kuitenkin paremmin raukeamisen ennustamiseen sopiva. Myös tämä oli linjassa aiempiin tutkimuksiin nähden.

Perustuen ROC-käyrään ja muihin suoritusmittareihin ei voitu vielä suoraan päätellä toimivatko käytetyt koneoppimismallit riittävän hyvin raukeamisen ennustamisessa. Tämän tarkasteluun muodostettiin taloudellisen hyödyn mittari, jonka avulla pyrittiin mittaamaan raukeamisen hallinnasta saatavaa hyötyä. Taloudellisen hyödyn mittari riippuu kuitenkin sen muodostamisessa tehtävistä oletuksista. Riippuen tehdyistä oletuksista ja käytetystä mallista tuotti raukeamisen hallinta joko hyvin pienen (tai jopa negatiivisen) taloudellisen hyödyn tai selvästi positiivisen taloudellisen hyödyn. Optimoimalla koneoppimismallia perustuen potentiaaliseen taloudelliseen hyötyyn oli tulosta mahdollista parantaa. Olettaen, että yhtiö pystyisi tarkemmin määrittämään raukeamisen hallinnasta johtuvan kustannuksen ja toisaalta asiakkaan säilyttämisestä saatavan potentiaalisen hyödyn olisi mittaria mahdollista tarkentaa. Tällöin koneoppimismallia olisi mahdollista kehittää juuri kyseiseen raukeamisriskin hallinnan sovellukseen sopivammaksi.

Koneoppimismenetelmien toimivuus riippuu suuresti myös käytetystä datasta ja saatavilla olevista muuttujista. Tuomalla malliin uusia ja parempia muuttujia sekä parantamalla datan laatua olisi raukeamisen ennustamista koneoppimismenetelmillä mahdollista kehittää entisestään. Lisäksi tekoäly ja koneoppimismenetelmät kehittyvät jatkuvasti, jolloin niistä saatava hyöty tulee todennäköisesti kasvamaan. Vaikuttaakin siltä, että tekoälyn ja koneoppimismenetelmien rooli vakuutusalalla tulee korostumaan tulevaisuudessa entisestään.

Viitteet

[1] Marco Aleandri. Modeling dynamic policyholder behavior through machine learning techniques. *Working paper*, 2017.

[2] Jorge Luis Andrade and José Luis Valencia. Modeling lapse rates using machine learning: A comparison between survival forests and cox proportional hazards techniques. *Anales del Instituto de Actuarios Españoles*, 2021.

[3] Diogo da Cunha Alcaide. Predicting lapse rate in life insurance: An exploration of machine learning techniques. *Master Program in Statistics and Information Management, Universidade Nova de Lisboa*, 2023.

[4] Anne A. H. de Hond, Ewout W. Steyerberg, and Ben van Calster. Interpreting area under the receiver operating characteristic curve. *The Lancet Digital Health*, 2022.

[5] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 1999.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2008.

[7] Mahlodi Kgare. Predicting lapse rate in life insurance using machine learning algorithms. *Master Program in Statistics, University of South Africa*, 2021.

[8] Dieter Kiesenbauer. Main determinants of lapse in the german life insurance industry. *North American Actuarial Journal*, 16(1):52–73, 2012.

[9] Euroopan Komissio. Komission delegoitu asetus (EU) 2015/35, 2015. `http://publications.europa.eu/resource/cellar/163318c6-1824-11ed-8fa0-01aa75ed71a1.0003.03/DOC_1`.

[10] Nordea Life. Mandatum life's solvency and financial condition report 2022, 2022. https://www.nordea.com/en/doc/solvency-and-financial-condition-report-2022-nordea-life-holding-ab.pdf.

[11] Stéphane Loisel, Pierrick Piette, and Cheng-Hsien Jason Tsai. Applying economic measures to lapse risk management with machine learning approaches. *arXiv:1906.05087*, 2019.

[12] Mandatum. Solvency and financial condition report 2022 nordea life holding ab, 2022. https://www.mandatum.fi/496115/contentassets/4aba1dc1a59d4f08b23e495250469453/mandatum-lifes-solvency-and-financial-condition-report-2022.pdf.

[13] Luke Merrick and Ankur Taly. The explanation game: Explaining machine learning models using shapley values. *Machine Learning and Knowledge Extraction*, 2020.

[14] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT press, 2018.

[15] David T. Russell, Stephen G. Fier, James M. Carson, and Randy E. Dumm. An empirical analysis of life insurance policy surrender activity. *Journal of Insurance Issues*, 36(1):35–57, 2013.