UPPSALA
UNIVERSITET

UPTEC STS 18034
Examensarbete 30 hp
Augusti 2018

Maskininlärning inom kommersiella fastigheter

Prediktion av framtida hyresvakanser

Brook Alemayehu
Fredrik Johnson

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Machine learning within commercial real estate: Prediction of future vacancies

Brook Alemayehu and Fredrik Johnson

The purpose of this thesis is to investigate the possibilities of predicting vacancies in the real estate market by using machine learning models in terms of classification. These models were mainly based on data from contracts between a Swedish real estate company and their tenants. Attributes such as annual renting cost and rental area for each contract were supplemented with additional data regarding financial and geographical information about the tenants. The data was stored in three different formats with the first having binary classes which aim is to predict if the tenant is moving out within a year or more. The format of the second and third version were both multi classification problems that aims to classify if the tenants might terminate their contract within a specific interval with the length of three and six months.

Based on the results from Microsoft Azure Machine Learning Studio, it is discovered that the multi classification problems perform rather poorly due to the classes being unbalanced. Regarding the performance of the binary model, a more satisfying result was obtained but not to the extend to say that the model can be used to determine a vacancy with high accuracy. It should rather be used as a risk analysis tool to detect if a tenant is showing tendencies that could result in a future vacancy. A major pitfall of this thesis was the lack of data and the financial information not being specific enough. The performance of the models will likely increase with a larger dataset and more accurate financial information.

Handledare: Jens Algerstam
Ämnesgranskare: Niklas Wahlström
Examinator: Elísabet Andrésdóttir
ISSN: 1650-8319, UPTEC STS 18034
Tryckt av: Uppsala

Populärvetenskaplig sammanfattning

Under den industriella revolutionen var tillgången på olja en starkt bidragande framgångsfaktor. På ett liknande sätt är idag tillgången på data ett verktyg för att skapa konkurrensfördelar i den pågående digitala revolutionen. Utvecklingen av bland annat smarta sensorer, självlärande datorer och molnbaserade lösningar har lagt grunden för ett enormt utbud av tillgänglig data. Trenden genomsyrar i princip alla branscher och fastighetsbranschen, som det här examensarbetet behandlar, är inget undantag. Tidigare forskning visar dock på att enbart samla in data inte nödvändigtvis är ett mål i sig, då den i regel behöver förädlas och tillämpas för att utgöra ett konkret värde.

Med hjälp av maskininlärning och tillgång till data innehållande hyreskontrakt har möjligheterna kring att prediktera framtida hyresvakanser i kommersiella fastigheter undersökts i detta examensarbete. Maskininlärning är uppbyggt kring beräkningsmetoder som använder sig av tidigare erfarenheter för att bland annat förbättra prestanda och skapa mer precisa prediktioner. Genom att skapa möjligheter för att förutse när ett hyreskontrakt kommer att sägas upp kan ett antal positiva effekter uppstå för ett kommersiellt fastighetsbolag. Den mest uppenbara effekten återfinns i en lägre vakansgrad då möjligheterna kring att matcha utbud och framtida efterfrågan förbättras. En ytterligare aspekt är att kommande renoveringar och fastighetsunderhåll underlättas då fastighetsägaren har en bättre uppfattning om antalet uthyrda objekt på längre sikt. På ett övergripande plan finns även potential för en förbättrad hyresmarknad och kundnöjdhet generellt.

Inom ramen för det här examensarbetet har maskininlärningsmodeller skapats med syftet att klassificera inom vilket tidsintervall ett hyreskontrakt förväntas sägas upp. Modellerna har tränats på data bestående av kontraktsutdrag, information om hyresgästernas finansiella status samt fastigheternas geografiska läge. Resultaten visar på svårigheter att utföra prediktioner med hög precision baserat på den tillgängliga datan och de modeller som utvärderats. Maskininlärning har utifrån denna studie visat sig vara ett besultsstöd som i dagsläget är svårt att implementera för prediktiva analyser inom fastighetsbranschen. Noterbart är dock att arbetet har aktualiserat flera intressanta aspekter inför framtida forskning inom området. Däribland vikten av en hög datakvalite

i form av att fastighetsägare samlar in relevant data som lagras på ett tillfredsställande sätt.

Förord

Detta examensarbete har genomförts som det avslutande momentet inom civilingenjörsutbildningen System i teknik och samhälle (STS) vid Uppsala universitet. Arbetet har utförts tillsammans med konsultföretaget Business Vision Consulting AB samt med stöd av institutionen för informationsteknologi vid Uppsala universitet.

Vi vill härmed rikta ett stort tack till de personer som på olika sätt har varit till stor hjälp under utformningen av denna rapport. Vår handledare på Business Vision, Jens Algerstam, har genomgående stöttat oss från idéutformning till praktiska råd och vägledning. Vår ämnesgranskare vid institutionen för informationsteknologi, Niklas Wahlström, har via sin djupa kunskap inom maskininlärning varit direkt avgörande för detta examensarbete. Vid sidan av dessa har även upplysningstjänsten Syna bidragit med data på ett generöst sätt. Avslutningsvis vill vi även passa på att tacka samtliga anställda på Business Vision som har bidragit med konstruktiv feedback och intresserat sig för vårt arbete.

Fredrik Johnson och Brook Alemayehu
Stockholm 2018-06-08

Lexikon

Lexikonet syftar till att underlätta för läsaren genom att belysa centrala begrepp som har använts i denna rapport. Förklaringen av del begrepp är anpassade till denna rapport och dess definition kan därmed skilja sig från andra definitioner av samma eller liknande begrepp.

- **Attribut**: Information avseende en specifik hyresgäst eller hyreskontrakt. T.ex. ett kontrakts löptid, lokaltyp och geografiskt läge.
- **Datapunkt**: Samling av attribut som avser en specifik instans i datasetet med olika attribut.
- **Dataset**: Samling av flera olika datapunkter gällande olika hyresgäster.
- **Träningsset (träningsdata)**: Samling av flera olika datapunkter gällande olika hyresgäster som en maskininlärningsmodell tränas mot.
- **Rådata:** Avser träningsdata och testdata innan den har förbehandlats. Med förbehandling menas att datan bland annat rensas och kompletteras för att uppnå en högre datakvalitet.
- **Testdata**: Samling av flera olika datapunkter gällande olika hyresgäster som en maskininlärningsmodell testas mot för att utvärdera dess prestanda.
- **Indata**: Den information som modellen får tillgång till via test- eller träningsset.
- **Utdata**: Det svar som datorn förväntas ge efter att ha bearbetat indatan. T.ex. en klasstillhörighet
- **Prediktion**: Prediktioner är klasserna som maskininlärningsmodellen ger datapunkterna i testdatan.
- **Maskininlärningsmodell (modell)**: Syftar till ett systematiskt försök att beskriva ett verklighetsbaserat fenomen. I det här fallet innebär detta prediktioner av framtida händelser.
- **Hyresvakans**: Lokal eller fastighet som saknar hyresgäst.
- **Binär**: Matematisk term som betyder att något kan ha två möjliga utfall.
- **Rekursivt:** En matematisk funktion som definieras utifrån referenser till sig själv.
- **Iteration:** Innebär att en funktion uppnår ett önskat resultat genom att upprepa beräkningar eller processer.

Innehållsförteckning

1. Inledning

Under slutet av 1700-talet var tillgången till olja en starkt bidragande faktor till den efterföljande industriella revolutionen. På ett liknande sätt är idag tillgången till data ett verktyg för att skapa konkurrensfördelar i den pågående digitala revolutionen (Venkatachalam, 2017). Utvecklingen av bland annat smarta sensorer, självlärande datorer och molnbaserade lösningar har bidragit till att skapa ett enormt utbud av tillgängliga datamängder och typer. Venkatachalam (2017) menar dock på att enbart samla in data inte är ett mål i sig, då den behöver förädlas och tillämpas för att utgöra ett konkret värde. Resonemanget får medhåll från författaren Hills (2016) som refererar till självkörande bilar, förbättrade reklamerbjudande och musik komponerad av datorer som exempel på produkter och tjänster som har sitt ursprung i stora datamängder som sedan har förädlats.

Venkatachalam (2017) skriver vidare att artificiell intelligens (AI) och maskininlärning är lämpliga metoder för att skapa just denna övergång mellan rådata och faktiska tillämpningar likt de tre exemplen som Hills skriver om. Trots att dessa begrepp nämns i många sammanhang kan den exakta innebörden och definitionen variera. En definition av AI är en dators förmåga att fylla samma intelligenta funktion som den mänskliga hjärnan (Rossini, 2000). Maskininlärning syftar i sin tur till att lära datorer att lära dem själva (Hills, 2016).

Både AI och maskininlärning har fått stor uppmärksamhet inom flertalet branscher, däribland fastighetsbranschen där den nya tekniken spås påverka sektorn på flera sätt. Till de mest nämnvärda användningsområdena hör bland annat prognostisering av framtida fastighetsunderhåll och riskminimering vid kreditgivning (Taylor, 2017). Till stor del bygger tekniken på ett proaktivt arbetssätt, vilket stämmer överens med hur många aktörer inom fastighetsbranschen arbetar redan idag. Fastighetsbolagen skapar prognoser för kostnader och intäkter parallellt med att investerare försöker förutse byggnaders framtida värde (Rossini, 2000).

1.1 Problemformulering

Inför genomförandet av det här examensarbetet har tidigare forskning som behandlar fastigheter i kombination med maskininlärning studerats. I många fall har rapporternas fokus legat på prediktering av framtida prisnivåer, vilket endast återspeglar en dimension av den svenska marknaden för kommersiella fastigheter. Lundström (2008) beskriver fastighetsmarknaden som resultatet av samspelet mellan fyra delmarknader i form av fastighet-, bygg-, kapital-, och hyresmarknaden. Den sistnämnda delmarknaden är den mest centrala och har således störst inverkan på hur framgångsrikt ett fastighetsbolag är (Lind & Lundström, 2009).

Mot bakgrunden att kommersiella fastighetsbolags affärsidé bygger på att hyra ut fastigheter och lokaler är vakanser, det vill säga att ett specifikt objekt saknar hyresgäst, ett återkommande problem bland fastighetsägare. I vissa sammanhang används begreppet naturliga vakanser, vilket kan jämföras med ett buffertlager där fastighetsägaren reserverar en viss mängd yta som ska kunna användas till att möta framtida efterfrågan som inte går att förutse (Grenadier, 1993). I andra fall beror en vakans på att efterfrågan generellt sett har sjunkit. Enligt Remøy och Van der Voordt (2007) styrs efterfrågan på ett specifikt objekt i regel av dess geografiska läge och skick samt det rådande läget på fastighetsmarknaden som helhet. Ett ytterligare exempel ges av att en befintlig hyresgäst väljer att säga upp kontraktet, som i sin tur kan grundas i flera olika faktorer.

Sammantaget kan vakanser uppstå till följd av många olika anledningar och en modell som kan förutse hyresvakanser skulle således kunna vara ett användbart hjälpmedel för fastighetsägare. Det är även intressant att studera om en viss typ av fastigheter uppvisar ett avvikande mönster jämfört med urvalsgruppen som helhet. Exempelvis om en viss typ av lokal eller område har en högre vakansgrad än andra samt vad de underliggande orsakerna i sådana fall grundas i.

1.2 Syfte

Syftet med detta examensarbete är att undersöka möjligheterna kring att prediktera hyresvakanser i kommersiella fastigheter med hjälp av maskininlärning. Utifrån tillgänglig datan bestående av hyresavtal mellan ett svenskt fastighetsbolag och dess hyresgäster samt geografisk och finansiell information aveende hyresgästerna är målsättningen att klassificera inom vilket tidsintervall ett hyreskontrakt eventuellt kommer att sägas upp. Vidare är ett delmoment även att utvärdera maskininlärning som beslutsstöd genom att studera tillförlitligheten och potentiella problemområden i prediktionerna. Mot bakgrunden av ovan nämnda syfte har följande delsyften formulerats för att genomföra studien:

- Skapa prediktiva modeller i Microsoft Azure Machine Learning Studio för att undersöka möjligheterna kring att förutse hyresvakanser i kommersiella fastigheter
- Utvärdera dessa modeller i termer av tillförlitlighet och potentiella problemområden inom fastighetsbranschen
- Identifiera eventuella korrelationer i datasetet som kan kopplas till hyresvakanser

1.3 Rapportens disposition

Efter det inledande stycket (kapitel 1) innefattande rapportens introduktion och syfte följer det andra avsnittet som avser att förse läsaren med nödvändig bakgrundsinformation kring området maskininlärning. Detta innefattar dels en introduktion till ämnet samt viktiga aspekter att ta hänsyn till vid predikativa analyser inom maskininlärning (kapitel 2). Vidare följer en detaljerad presentation av det data som har inkluderats i studien (kapitel 3) och det tillvägagångssätt som har använts vid den inledande datainsamlingen till konstruktionsarbetet av modellerna (kapitel 4). Därefter följer resultatet från den modellering som har genomförts (kapitel 5). Studien avslutas med en diskussion (kapitel 6) och slutsats (kapitel 7) som återkopplar till det inledande syftet och det genomförda arbetet.

2. Bakgrund

I det här avsnittet introduceras inledningsvis forskningsfältet AI och mer specifikt dess underområde maskininlärning. Därefter ges en mer detaljerad beskrivning av delområdet supervised learning som studien ämnar att fördjupa sig inom. Därefter följer en genomgång av de algoritmer som har studerats samt vilka typer av utvärderingsmått som har använts för att bedöma kvalitén på de aktuella maskininlärningsmodellerna.

2.1 Introduktion till forskningsfältet

AI var tidigare främst förknippat med spelindustrin där man utvecklade spel som kunde erbjuda spelaren en verklighetsbaserad spelpartner eller motståndare, och därmed benämndes som intelligent trots att det var datorstyrt (Charles et al. 2008, s.9). Numera är tillämpningsområdena i princip obegränsade och återfinns inom bland annat bildigenkänning, prediktiva analyser och självkörande fordon. Enligt Nationalencyklopedin (2018) kan AI definieras utifrån termerna inlärning, tänkande och intuition. Det vill säga att hantera lärande, anpassning till olika situationer samt utnyttja tidigare erfarenheter på ett effektivt sätt (Nationalencyklopedin, 2018). Ur ett användarperspektiv är implementering av AI applicerbart inom områden där mänsklig intelligens efterfrågas (Russel, 1995). Lohr (2012) skriver vidare att maskininlärning, som är ett delområde inom AI, är uppbyggt kring intelligenta algoritmer som tränas utifrån tidigare data. Med andra ord kan AI och maskininlärning vara användbart inom en kontext som kombinerar komplexa uppgifter med tillgång till stora datamängder. Inom maskininlärning medför en större mängd data att algoritmen kan lära sig mer (Lohr, 2012).

2.2 Maskininlärning

Maskininlärning är ett begrepp som blivit allt mer omtalat de senaste åren. Men vad är egentligen maskininlärning? Detta är en fråga som har fått ett allt tydligare svar till följd av att flera pionjärer hjälpt till att utveckla industrin under de senaste sex årtiondena (Bell, 2014). I en rapport skriven av matematikern Alan Turing (1950) ställer sig författaren frågan *"Can machines think?"* I ett försök till att besvara sin egen fråga beskriver Turing det så kallade *Imitationsspelet*, där två deltagare bestående av en dator och en människa

tävlar mot varandra. Syftet med spelet är att respektive deltagare ska övertala en domare att just de är en människa och inte en dator. Denna typ av experiment tillämpas än idag genom bland annat tävlingar där datorer försöker övertala en domare att de chattar med en människa (Bell, 2014). Den amerikanske pionjären inom AI och maskininlärning, Arthur Samuel (1959), definierar maskininlärning som ett tillvägagångssätt för att skapa möjligheten för datorer att lära sig utan att bli specifikt programmerade för hur de ska agera vid varje unik situation. Samuel har främst krediterats för sitt arbete inom spelindustrin och för sitt arbetet inom maskininlärning under hans anställning på IBM (Bell, 2014). Som jämförelseobjekt till Samuels definition, kan Tom M.Mitchell, styrelsemedlem för maskininlärning på Carnegie Mellon University, definition användas:

"A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T, as measured by P, improves with the experience E" (Mitchell, 1997).

Samuel och Mitchell, har i sällskap med många andra, lett utvecklingen av maskininlärning till dess nuvarande status. Det går idag att övergripande beskriva maskininlärning som beräkningsmetoder vilka använder sig av tidigare erfarenheter för att förbättra prestanda eller skapa mer precisa prediktioner. Erfarenheten som beräkningsmetoden använder refererar till den tidigare informationen som finns tillgänglig. Denna information består i regel av elektronisk data som är samlad och förberedd för att analyseras. Oavsett i vilken form datan finns i så är den absolut viktigaste aspekten dess storlek och kvalitet, då detta är avgörande för hur lyckosam maskininlärningsmodellen kommer att vara i sin prediktion (Bell, 2014).

Enligt Mohri (2014) kan maskininlärningsmodeller kategoriseras efter vilka metoder eller algoritmer som de använder. Två av dessa kategorier ges av så kallad övervakad inlärning (eng. supervised learning) och oövervakad inlärning (eng. unsupervised learning) som illustreras i figur 1 nedan. Gällande den förstnämnda kategorin är det övergripande målet att utifrån insamlad data prediktera värden på utdata (Hastie et al., 2009: s.9-11). Strategin bygger på att en algoritm använder sig av ett känt dataset (träningsdata) bestående av in- och utdata. Exempelvis kan en klassificeringsalgoritm (övervakad inlärning) lära sig att identifiera djur efter att den har tränats på ett dataset innehållande bilder på djur som har märkts upp med korrekt art och ett antal andra kännetecken för

djuret i fråga. Oövervakad inlärning syftar i sin tur till metoder som ska finna mönster utan att de förses med ett givet svar, likt klassetiketterna i övervakad inlärning.

Syftet med träningsdatan är att skapa en modell som ska kunna prediktera utdata för ny och tidigare osedda indata. För att kunna validera modellens precision används ett ytterligare dataset som benämns som testdata (Mathworks, 2018). Med hjälp av detta dataset är det således möjligt att utvärdera hur väl modellens prediktioner överensstämmer med verkligheten som ges av informationen i testdatasetet. Nedan följer en sammanfattning av de mest centrala delområdena inom maskininlärning, där fokus i denna rapport kommer att riktas mot supervised learning:

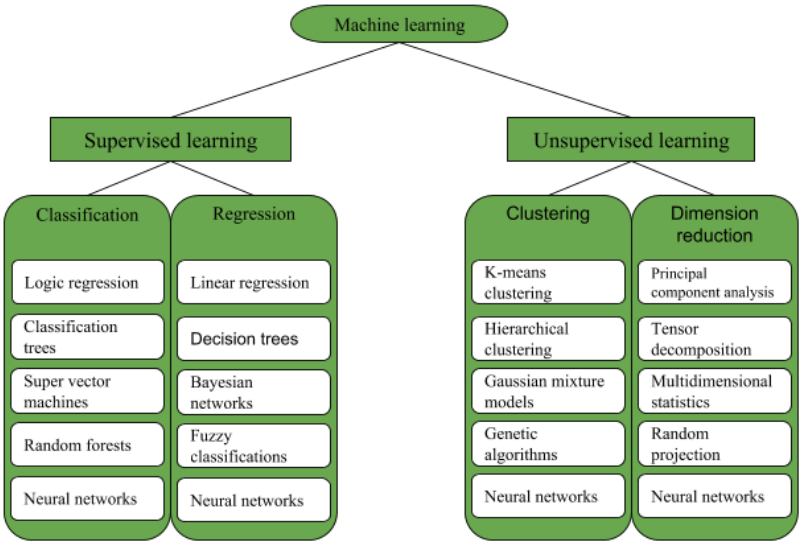
Supervised learning

- *Klassificering*: Denna kategori av maskininlärningsmetoder har målet att prediktera vilken klass olika datapunkter tillhör. Ett exempel på detta skulle kunna vara textanalys där modellen ska avgöra om det är ett positivt laddat meddelande eller negativt laddad. Detta skulle även kunna vara bildklassificering där modellen ska avgöra vilken kategori bilden tillhör i form av till exempel landskap, djur eller porträtt (Mohri, 2014).
- *Regression:* Till skillnad från klassificering predikterar en regressionsmodell ett kontinuerligt värde för den aktuella indatan. Dessa prediktioner kan utgöras av aktievärden eller förändringar i andra ekonomiska variabler. Prestandan i en regressionsmodell skiljer sig från klassificering då den mäts i förhållande till ett reellt värde. Inom klassificering är det predikterade värdet antingen rätt eller fel då förses med en klassetikett som inte kan mätas i förhållande till ett referensvärde på samma sätt (Mohri, 2014).

Unsupervised learning

- *Klustring:* Delar in datan in i likartade regioner. Detta används oftast när målet är att analysera väldigt stora dataset. När sociala nätverk ska analyseras så används oftast klustring då algoritmen försöker identifiera olika grupper inom stora grupper människor (Mohri, 2014).

- *Dimensionell reduktion:* Reducerar dimensionaliteten av datan till en lägre dimension medan den behåller vissa egenskaper av den ursprungliga representationen. Ett vanligt exempel av detta är förbehandlingen av digitala bilder (Mohri, 2014).



Figur 1: Illustration av vilka klasser som tillhör respektive kategori samt vilka algoritmer som kan användas (Louridas & Ebert, 2016).

2.3 Designprocessen

Maskininlärningsfältet innefattar tusentals varianter av algoritmer som i sin tur kan tillämpas inom olika områden. Gemensamt för samtliga algoritmer inom maskininlärning är dock att de är uppbyggda kring samma typ av struktur som kan delas in tre delområden - representation, evaluering och optimering (Domingos, 2012). Vidare menar Domingos (2012) att det är möjligt att applicera olika tillvägagångssätt inom respektive område vid arbetet att utveckla en prediktiv modell. Det är dock inte möjligt att skapa en kombination som löser alla typer av problem utan olika kombinationer är användbara i olika situationer (Shawkat & Smith, 2004). Mitchell (1998, s.5-10) har utvecklat en generell designprocess som kan ligga till grund för ett mer djupgående arbete kring att utforma de tre delområdena som nämnts ovan. Nedan följer en beskrivning av Mitchells designprocess:

Val av algoritm

Enligt Mitchell (1998) syftar det inledande steget till att välja rätt algoritm för modellen. Detta är ett centralt moment då ett felaktigt val kan påverka inlärningsprocessen i en negativ riktning. I många fall skiljer sig den teoretiska utgångspunkten från praktiska tillämpningar. Enligt teorin är inlärningsprocessens pålitlighet som högst när mängden träningsdata är ekvivalent med mängden testdata. Inom praktiska tillämpningar är det dock vanligt förekommande att mängden träningsdata som används under inlärningsprocessen skiljer sig från testdatan som modellen slutligen kommer att valideras mot (Mitchell, 1998: s.6).

Val av målfunktion

Det andra steget i designprocessen syftar till att definiera vilken typ av kunskap som systemet ska tränas till att uppnå. I detta arbete är målfunktionen $F(x)$ när en hyresvakans kommer uppstå, där x representerar de olika datapunkter. Utformningen av målfunktionen kan variera från fall till fall då den följer av vilken typ av problem som ska behandlas. Det vill säga att en målfunktion kan vara allt från att prediktera en hyresvakans till att en modell ska kunna identifiera en bild av en hund när det är målfunktionen. Vid praktiska genomföranden kan det vara problematiskt att träna en modell att prediktera en målfunktion på ett tillfredsställande sätt då den begränsas av bland annat beräkningskapaciteten (Mitchell, 1998: s.7). Ett vanligt tillvägagångssätt är därför att göra approximeringar av målfunktionen då inlärningsprocessen ska resultera i en beskrivning av den ideala målfunktionen som både är effektiv och möjlig att implementera (Mitchell, 1998: s.7).

Val av representation för målfunktion

Utöver att en målfunktion ska formuleras, måste den även representeras på ett sådant sätt som en dator kan hantera. Detta kan göras genom att formulera en hypotes, vilket är en specifik funktion vars syfte är att efterlikna målfunktionen i så stor utsträckning som möjligt (Raschka, 2015). I detta arbete så representerar de olika klasserna hypoteserna. Enligt Blockeel (2011) tillämpar många maskininlärningsmodeller en slags sökprocess där ett antal datapunkter i testdatan appliceras på alla möjliga hypoteser, förutsatt att de skulle kunna passa testdatan. Utifrån detta synliggör sedan algoritmerna de hypoteser som bäst beskriver datan. Slutligen menar Mitchell (1997) att en avvägning mellan mängden träningsdata och hur väl representationen motsvarar målfunktionen måste göras. Om

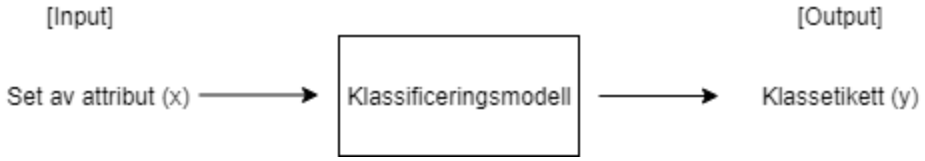
approximeringen är väldigt lik målfunktionen krävs exempelvis en större mängd träningsdata.

Evaluering och optimering av systemet

Det avslutande steget syftar inledningsvis till att särskilja bra och dåliga modeller från varandra. Detta görs med hjälp av olika tillvägagångssätt för att utvärdera modellerna (Mitchell, 1997: s.9). Exempelvis genom att studera korrekthet, precision eller känslighet (dessa begrepp beskrivs mer utförligt i avsnitt 2.6). Därefter identifieras den bäst presterande modellen utifrån ovannämnda utvärderingsmått.

2.4 Klassificering

Utifrån tidigare nämnda figur 1 går det att utläsa hur övervakad inlärning i sin tur kan delas in i undergrupperingar innehållande ett antal olika algoritmer. Enligt Browniee (2016) är klassificering (eng. classification) en av de vanligaste metoderna inom övervakad inlärning. Domingos (2012) skriver vidare att klassificering kan definieras som ett system vars syfte är att ange vilken klass en viss typ av indata tillhör. För indatan kan värdena antingen vara diskreta eller kontinuerliga och för utdatan kan de endast vara diskreta (Domingos, 2012).



Figur 2. Översikt av klassificeringsprocessen

Ett klassificeringsproblem utgår från en viss datamängd $(x_1, y_1), \ldots, (x_n, y_n)$ för att konstruera en modell. Enligt Tan (2005) är det möjligt att beskriva klassificering genom att varje datapunkt x_i ur dess mängd X är kategoriserad med etiketten y_i från målmängden Y. I korthet är syftet med klassificeringsmodeller och dess algoritmer att definiera en funktion som innehar förmågan att ange rätt etikett till nya och tidigare okända datapunkter ur mängden X (Cunningham, et al. 2008).

Learned-Miller (2014) exemplifierar ett klassificeringsproblem genom att visa på praktiska användningsområden inom sjukvården. I exemplet beskrivs ett träningsdataset inom klassificering som n stycken ordnade par i form av $(x_1, y_1), \ldots, (x_n, y_n)$, där x_i

representerar mätningar från en specifik datapunkt och y_i i sin tur representerar en etikett för samma datapunkt. Om x_i motsvaras av patientinformation innefattande attributen vikt, längd och blodtryck etc, skulle y_i kunna beskriva etiketterna *hälsosam* och *icke-hälsosam*. Klassificering kan således bidra till att göra datadrivna gissningar kring framtida patienters hälsa baserat på informationen i träningsdatan (Learned-Miller, 2014).

2.4.1 Riskfaktorer vid klassificering

Vid övervakad inlärning finns ett antal riskfaktorer som kan påverka en modells utfall och tillförlitlighet när den tränas på befintlig data. Det mest centrala vid klassificeringsproblem ligger i risken att modellen förses med obalanserad data eller att den överanpassas till den data som den tränas på och därmed får svårigheter att analysera ny data. Vidare kan även bristande datakvalitet och storlek på datasetet påverka modellen negativt då den inte ges tillräckligt goda förutsättningar i inlärningsprocessen.

Obalanserad data

Problematiken som uppstår tillföljd av så kallad obalanserad data (eng. Imbalanced data) är relativt vanligt förekommande inom maskininlärning (Awad & Khanna, 2015: s.52). I korthet innebär detta att en klass har markant mindre antal datapunkter jämfört med den andra klassen, vilket således innebär att hela datasetet blir obalanserat (Awad & Khanna, 2015: s.52). Den mindre klassen benämns som minoritetsklassen (eng. minority class) och den större som majoritetsklassen (eng. majority class). Vid modellering med obalanserade dataset där det huvudsakliga målet är att skapa en så bra prediktion för den mindre klassen som möjligt är det viktigt att använda sig av rätt typ av utväderingsmått. Istället för att fokusera på modellens totala precision bör fokus riktas mot korstabellen, vilken beskrivs närmare under avsnittet modellutvärdering (Awad & Khanna, 2015: s.53).

Om målet istället är att transformera det obalanserade datasetet till ett balanserat finns ett antal metoder att tillgå. Utöver att fundamentala åtgärder som till exempel att försöka samla in mer data, utvärdera olika algoritmer eller på annat sätt ändra i modellen finns även mer sofistikerade lösningar. Bland annat är det möjligt att skapa syntetiska datapunkter, vilket innebär att en algoritm skapar ny data utifrån minoritetsklassen. Metoden ska inte förväxlas med att enbart kopiera existerande datapunkter då den bygger på statistiska beräkningar som skapar nya unika datapunkter.

Överträning

Trots att modellutvärderingen indikerar ett positivt resultat kan det faktiska utfallet vara det motsatta till följd av överträning. Överträning eller överanpassning (*eng. Overfitting)* inom maskininlärning uppstår när en modell anpassas i för stor utsträckning till dess träningsdata. Modellen riskerar att bli kontraproduktiv då den istället för att tillämpa generella regler och mönster som grund för prediktionen, lär sig att känna igen brus och andra egenheter i den tillgängliga träningsdatan (Dieterich, 1995). I förlängningen innebär detta att en algoritm som inte har tränats till att arbeta med generella mönster riskerar att få svårigheter med att behandla ny data och på så sätt skapar en felaktig bild av verkligheten (Engelbrecht, 2007).

Inom maskininlärning har överträning blivit ett centralt problem och kan i många fall vara svårt att upptäcka (Domingos, 2012). Genom att bryta ned generaliserbara problem i bias och varians kan en större förståelse för överträning skapas. Bias är maskininlärning modellens tendens att konsekvent lära sig samma fel. Varians är i sin tur tendensen att lära sig slumpmässiga saker oberoende av den faktiska signalen (Domingos, 2012). I figur 1 nedan illustreras korrelationen mellan bias och varians, samt hur ändringar i dessa parametrar påverkar modellens prestanda.

Datakvalitet

Vid datadrivna analyser är förbehandling av data ett centralt moment. Tidigare forskning visar på att förbehandling av data utgör cirka 80 procent av det totala arbetet inom dataintensiva applikation (Zhang, et al., 2003). I många fall är den tillgängliga datan osammanhängande, ofullständig och fylld av brus som kan påverka analysarbetet negativt. Att förbehandla sådan data tills dess att en tillräckligt god kvalitet uppnås leder oftast till en reducerad datamängd jämfört med ursprungsdatan. Zhang (2003) skriver vidare att mer högkvalitativ data skapar bättre förutsättningar för att även analyserna ska hålla en hög kvalitet. Resonemanget får även medhåll från Pyle (1999) som skriver att bristfällig kvalitet på den ingående datan kommer att resultera i motsvarande kvalitet på det utgående resultatet.

Datasetets storlek

Utöver ett datasets kvalitet har dess storlek också visat sig ha stor betydelse (Mukherjee et al., 2003). Med andra ord tenderar en klassificerare att uppvisa ett bättre resultat när storleken på datasetet växer. I praktiken är dock den tillgängliga mängden data begränsad och en relevant frågeställning är därför hur stor mängd data som krävs för ett specifik problem. För att hantera ett för litet dataset finns ett antal olika tillvägagångssätt. Ett alternativ är definiera den minsta storleken som erfordras för att avvisa en nollhypotes och därmed kunna uppnå så kallad statistisk säkerhet (Maxwell et al, 2008). Ett annat alternativ är istället att utöka mängden data genom att skapa dubbletter av den befintliga datan.

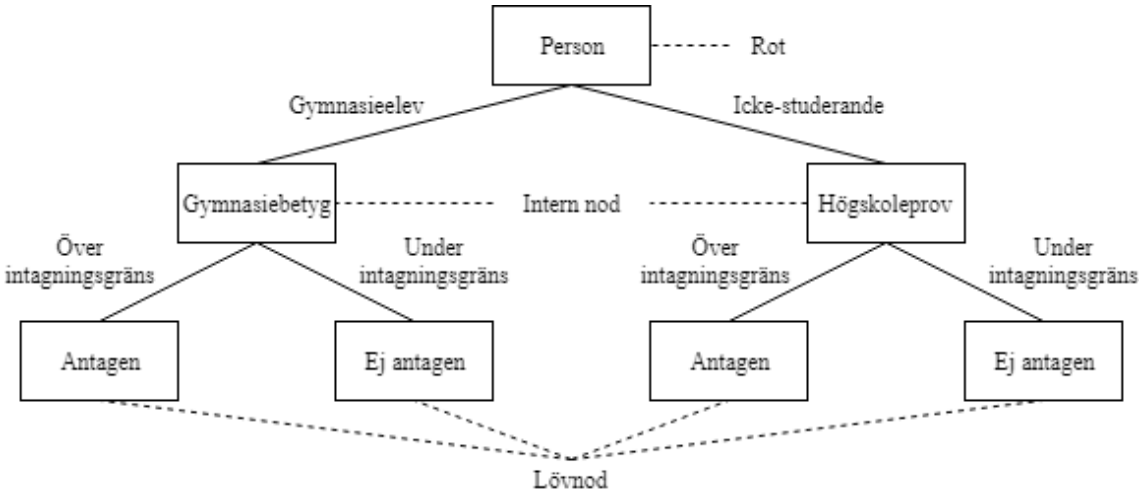
2.5 Algoritmer

Inom maskininlärning och dess underområde övervakad inlärning finns ett brett utbud av algoritmer. Valet av algoritm styrs av problemformuleringen och vilken data som finns tillgänglig. Detta medför att det generellt sett finns algoritmer som lämpar sig för binära frågeställning där det endast efterfrågas en klassificering mellan två klasser. Samtidigt som det även finns algoritmer som är menade att hantera en klassificeringsprocess där fler än två klasser ska behandlas. Enligt *No Free Lunch Theorem* finns det ingen algoritm som presterar bättre än andra algoritmer för alla möjliga typer av dataset (Shawkat & Smith, 2004). Detta medför att de flesta klassificeringsproblem kräver att ett antal algoritmer utvärderas för att bestämma vilken som är mest lämpad att använda. Utifrån detta har algoritmerna som presenteras nedan valts ut till följd av att de representera ett urval av de vanligaste algoritmerna inom klassificering enligt Shawkat och Smith (2004).

2.5.1 Beslutsträd

Beslutsträd (eng. decision tree) är en maskininlärningsteknik som tillämpas inom flera olika områden. Från att ursprungligen ha utvecklats i termer av informationsutvinning och mönsterigenkänning, finns det numera tillämpningsområden inom flertalet industrier. Bland annat för att ställa medicinska diagnoser, beräkna försäkringspremier och bedöma låntagarens kreditvärdighet (Coadou, 2013: s.1). Inom klassificering används i regel en funktion för att tilldela en ny instans en klass baserat på majoriteten av de observationerna som har använts för att träna modellen (James et al., 2013).

Ur ett matematiskt perspektiv är ett beslutsträd ekvivalent med ett rotat binärt träd, det vill säga ett träd som enbart består av två klasser (Coadou, 2013: s.2). Beslutsträdet utgår från den så kallade rotnoden (eng. root node) och varje nod kan i sin tur delas i två stycken nya noder framtill dess att ett stoppkriterium uppnås.



Figur 3. Illustration av ett beslutsträd för unversitetsantagning

I figur 3 går det att utläsa hur sorteringen inleds från trädets översta del i form av dess rot (eng. root). Utifrån modellens uppsatta regler, exempelvis om person i fråga är student eller icke-studerande, växer trädet fram via olika förgreningar tills dess att ett stoppkriterium uppnås. Beslutsträdets förgreningar benämns som interna noder (eng. Internal nodes) och de noder som befinner sig längst ner i trädet kallas för löv (eng. Leaves). Shotton (2013) beskriver vidare logiken bakom ett beslutsträd genom att poängtera ett antal centrala aspekter. Vid en grafisk representation av ett träd, likt den i figur 4, följer kopplingarna mellan trädets noder en specifik riktning, det vill säga att $X \rightarrow Z$ är inte detsamma som $Z \rightarrow X$. Vidare finns det endast en väg till respektive nod, i praktiken innebär detta att varje barnnod (eng. Child node) inte kan ha mer än en föräldranod (eng. Parent node) som är placerad ovanför.

Torgo (2000) skriver att ett sätt för att konstruera ett beslutsträd är att rekursivt dela in träningssetet i mindre delset genom att definiera regler som underlättar beslutsprocessen. Till dessa regler hör ett tillvägagångssätt för att välja vilket attribut som algoritmen ska splita på, exempelvis det attribut som skapar noder med högst homogenitet. Vidare måste algoritmen kunna avgöra när en nod är en lövnöd. Till exempel om samtliga datapunkter i en nod har samma klasstillhörighet. Den avslutande regeln syftar till att ange

klassetiketten för varje lövnod. Ett sätt är att ange den klass som majoriteten av de datapunkter som har använts för att träna modellen inom den aktuella regionen. Förutsatt att X_t representerar träningsdatan där t utgör en nod och $y = y_1, y_2, \dots, y_n$ motsvarar klassetiketterna för ett problem med k stycken klasser, kan en rekursiv algoritm förklaras utifrån två steg (Torgo, 2000):

- Om alla datapunkter i X_t tillhör klass k är t en lövnod med tillhörande klassetikett y_t
- Om X_t består av datapunkter som tillhör mer än en klass används ett attributtest för att kunna dela in datan i mindre subset. Barnnoder genereras utifrån testets utfall och instanserna i fördelas X_t sedan ut till dessa noder.

Nackdelen med att basera modellen på enskilda beslutsträd återfinns i att de i många fall har svårt att återspegla verklighetsbaserade problem som har en mer komplex struktur. Problematiken ligger huvudsakligen i att praktiska exempel kräver ett stort antal beslutskriter och att det därför är svårt att konstruera generella modeller (Lawsona et al., 2017). Ett sätt för att förbättra en klassificerare är med hjälp av enså kallad ensabelmetod, det vill säga en metod där resultaten från flera beslutsträd viktas samman. Bland tillvägagångs sätten återfinns Boosting och Decision Forest, vilka beskrivs nedan.

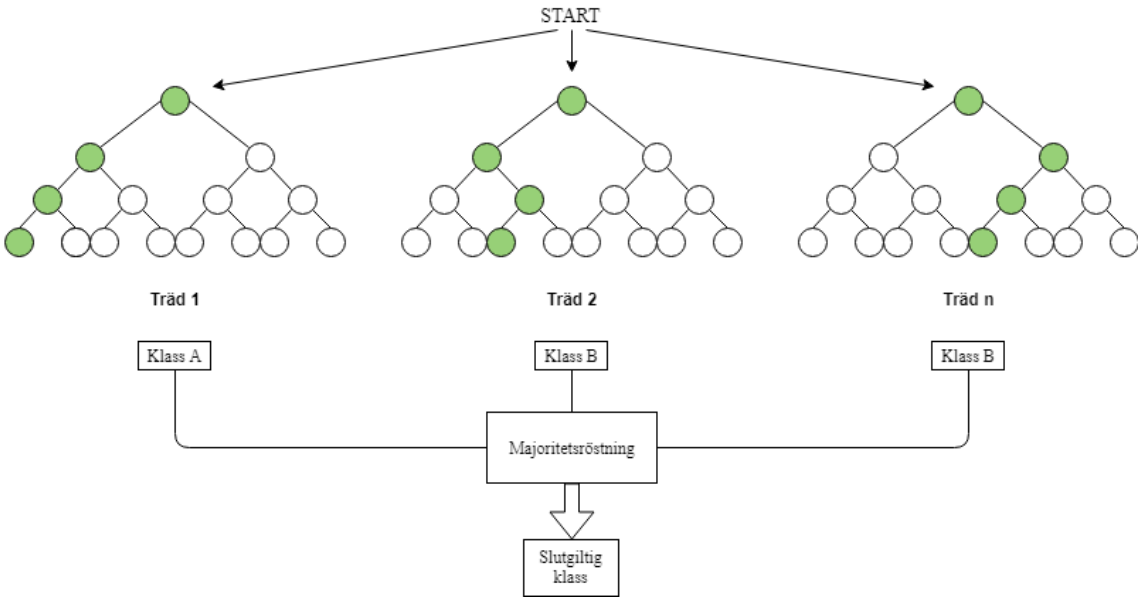
Boosting

Enligt Coadou (2013) är så kallad boosting ett effektivt sätt att förbättra en klassificerare. Detta är något som inte enbart gäller för beslutsträd utan har även visat sig vara effektivt för bland annat neurala nätverk. I korthet innebär metoden att ett andra träd korrigerar för misstagen i det första trädet, därefter skapas ett tredje träd som i sin tur korrigerar för misstagen i det andra trädet och så vidare (Microsoft Azure, 2018). Prediktionen baseras sedan på det sammanvägda resultatet från samtliga träd. En nackdel som bör belysas med modellen är dess krav på hög minneskapacitet och därför kan ha svårt att hantera väldigt stora dataset.

Decision Forest

Som namnet antyder bygger dessa algoritmer på information från flera olika beslutsträd. Det är möjligt att basera algoritmerna på träd från både regressions- och klassificeringsproblem, vilket gör dem användbara vid olika typer av frågeställningar

(Suthaharan, 2016). En gemensam nämnare med ovan nämnda boosting är att båda tillvägagångssätten bygger en klassificerarer utifrån ett stort antal av mindre klassificerare. Boosting innebär att varje klassificerare tränas sekvensiellt genom att en ny klassificerare tränas för att förbättra sin föregångare. I fallet för Decision Forest tränas respektive klassificerare istället oberoende av övriga klassificerare. En nackdel är dock dess känslighet för överanpassning, där olika klassificerare kan överanpassa datan på olika sätt.



Figur 4. Illustration av algoritmen Decision Forest

I Figur 4 illustreras en förenklad bild över Decision Forests beslutsgång. Noterbart är hur problematiken för ett enskilt beslutsträd kan kringgås då det är möjligt att hantera ett större antal beslutskriterium. Vidare tillämpas röstning (eng. Voting) för att hitta den mest populära klassen (Microsoft Azure ML Studio, 2018). Träd som har en hög precision kommer även att viktas tyngre i modellens slutgiltiga beslut. Nämnvärt är även att algoritmen som återfinns i Microsoft Azure ML Studio använder hela datasetet för varje träd, vilket skiljer sig något från det vanligaste tillvägagångssättet inom Decision Forest där respektive träd i många fall endast använder slumpvist utvalda delar ur datasetet (Microsoft Azure ML Studio, 2018).

Multiclass Decision Forest

Det finns två olika metoder för att använda decision forest när det förekommer flera klasser. Den ena metoden är en-mot-andra metoden (eng. One-against-others) vilket bygger på att reducera en K-klass klassificeringsproblem till ett binärt problem. En ny klassbenämning är definierad för k i det binära problemet enligt följande (Sun et al. 2010):

$$Y_k = \begin{Bmatrix} 1\ om\ datapunkten\ \text{är}\ i\ klass\ k \\ -1\ annars \end{Bmatrix} \qquad (1)$$

Till exempel i fallet det finns tre (klass *A, B och C)* olika klasser så omvandlas detta till tre olika binära klassificeringsproblem. I första problemet så blir klass $A = 1$ medan klass $B\ och\ C = -1$. Det andra problemet tilldelar klass $B = 1,\ A\ och\ C = -1$. Slutligen så blir det sista problemet så att klass $C = 1$ medan $A\ och\ B = -1$. Det vill säga för k antalet klasser skapas K antal klassificerare. För att göra prediktioner på ny data så appliceras det till alla binära klassificerare och prediktionen returneras. En datapunkt från testdatan undersöks alltså på alla klassificerare. Slutligen är klassen av den nya datan predikterad enligt följande (Sun et al. 2010):

$$F(x) = \arg\max_k \{Y_k, k = 1, \dots, K\} \qquad (2)$$

Den andra metoden för att prediktera K antal klasser med decision forest är att göra det parvis. Även denna metod reducerar klassificeringsproblemet till ett binärt problem, men genom $K(K-1)/2$. Om problemet från början har fyra olika klasser så kommer denna metod dela upp problemet i sex olika klassificeringsproblem $\frac{4(4-1)}{2} = 6$. De samtliga klasserna paras ihop med alla möjliga kombinationer och beräknas som ett binärt problem. Anta vidare att det nya binära problemet har klasserna k_1 och k_2 så definieras de vidare enligt (Sun et al. 2010):

$$Y^{(k_1,k_2)} = \begin{Bmatrix} 1\ om\ datapunkten\ \text{är}\ i\ klass\ k_1 \\ -1\ om\ datapunkten\ \text{är}\ i\ klass\ k_2 \end{Bmatrix} \qquad (3)$$

Den binära klassificieraren svarar med $Y(k_1, k_2)$ genom att endast använda sig utav klass k_1 eller k_2. Skillnaden från den andra metoden är alltså att denna metod endast undersöker två klasser i taget. I fallet som den tidigare modellen där klasserna *A, B* och *C* användes skulle denna metod resultera i att det första klassificieringsproblemet innehåller endast

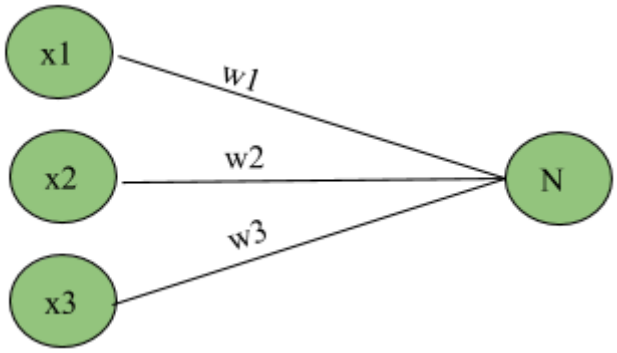
klass *A* och *B.* Klass *A* tilldelas den nya klassen 1 samtidigt som klass *B* tilldelas klassen -1. Det andra problemet använder sig utav klass *A* och *C*, men där *A=1* och C=*-1*.Sista klassificeringsproblemet innehåller klass *B, C* där *B=1* och *C= -1*. Slutligen får datapunkten från testdatan den klass som vinner flest parvisa jämförelser. I det fall om det uppstår flera klasser vilka har vunnit lika många gånger så slumpas den slutgiltiga klassen från en av vinnarna (Sun et al 2010).

2.5.2 Neurala nätverk

Multiclass Neural Network

Artificiella neurala nätverk (ANN) är en beräkningsmodell inspirerad av biologi. Det vill säga att modellen efterliknar hur hjärnan skickar information mellan olika neuroner. Modellen processar element, så kallade neuroner, som viktas utefter sin relevans innan de skickas vidare i nätverket (Shanmuganathan 2016). Deboeck och Kohonene (1998) beskriver neurala nätverk som en sammansättning av olika matematiska metoder som kan användas för att förutse händelser, klustring, klassificering och signalbehandling.

Neuronerna skickar information mellan varandra med hjälp av en aktiveringsfunktion. Viss data kan ha en högre relevans för neuronerna och därav bör neuronerna prioritera denna information. Detta gör nätverket genom att vikta datan innan den når neuronerna. En neuron *N* med dess data *x* och vikter *w* kan ses i figur 5 nedan. Neuronerna summerar alla vikter och data som kommer in innan den skickar ifrån sig en utsignal som antingen går vidare till en ytterligare neuron/neuroner eller så representerar utsignalen det slutgiltiga svaret om det inte finns något ytterligare lager. Antalet lager och neuroner väljs efter storleken på träningsdatan och dess kvalitet då för många gömda lager och neuroner kan leda till överträning medan för få kan leda till att det önskade målet inte uppfylls (Priddy & keller, 2005).



Figur 5: Figuren visar hur data kopplas till en neuron med vikter mellan.

2.6 Modellutvärdering

Inom forskningsfälten för informationsutvinning och maskininlärning finns ett antal metoder som kan användas vid modellutvärdering. Modellerna som innefattas i den här rapporten behandlar klassificeringsproblem och kan därmed utvärderas med hjälp av korstabeller (eng. Confusion matrix) som består av en matris av storleken $k \times k$ för k antal klasser (Ting, 2010).

		Predikterad klass
	Sant positiv	Falskt negativ
Verklig klass	Falskt positiv	Sant negativ

Figur 6. Korstabell som visar de potentiella utfallen för två stycken klasser.

Utifrån figur 6 går det att utläsa att *sant positiv* (eng. true positive, TP) och *sant negativ* (eng. true negative, TN) representerar antalet observationer som har predikteras korrekt. På samma sätt representerar *falskt positiv* (eng. false positive, FP) och *falskt negativ* (eng. false negative, FN) antalet observationer som har predikteras felaktigt. Således är summan av TP, TN, FP och FN det totala antalet observationer. Detta gäller endast binära klasser, när det kommer till modeller som använder sig utav fler än 2 klasser så visas andelen korrekt predikterade i matrisen och sedan andelen felaktiga prediktioner i de andra cellerna. Ett optimalt resultat skulle därmed ges av en matris innehållande enbart nollor bortsett från diagonalen mellan $(1, 1)$ och (k, k), vilket skulle innebära att modellen predikterat rätt utfall för samtliga observationer (Ting, 2010). Ur ett

modellerings perspektiv inom klassificering är måtten som beskrivits ovan grunden för en mer utförlig analys där korrekthet (eng. accuracy), precision (eng. precision) och känslighet (eng. recall) beräknas.

Korrekthet är det mest intuitiva måttet då det anger andelen korrekt predikterade observationer dividerat med det totala antalet observationer (Sammut et al., 2011). Att enbart studera korrektheten kan dock vara vilseledande om inte datasetet är symmetriskt där värdena för falskt positiva och falskt negativt i princip är ekvivalenta.

$$Korrekthet = \frac{TP+TN}{TP+TN+FP+FN} \qquad (4)$$

Precision kan vidare definieras som andelen positiva observationer som är korrekt predikterade dividerat med det totala antalet positivt predikterade observationerna. Följaktligen korrelerar en hög precision med en låg andel observationer som indikerar falskt positiv.

$$Precision = \frac{TP}{TP+FP} \qquad (5)$$

Slutligen kan känslighet beskrivas som de positivt korrekt predikterade observationerna dividerat med samtliga positiva värden. Känslighet är med andra ord en indikation på hur stor andel av de positiva värdena som klassificerades korrekt.

$$Känslighet = \frac{TP}{TP+FN} \qquad (6)$$

Vid frågeställningar som innefattar mer än två klasser kan det vara nödvändigt att använda så kallade mikro- och makro-medelvärden för precision och känslighet (Yang, 1997). Utifrån ekvation (4)-(6) går det att utläsa hur dessa medelvärden är en vidareutveckling av de mått som beskrivits ovan. Enligt Yang (1997) baseras ett makro-medelvärde på prestandan i respektive klass utan att ta hänsyn till hur frekvent samma klass förekommer. I fallet för ett mikro-medelvärde tillämpas istället summan av samtliga sant positiva, falskt positiva och falskt negativa. Detta är möjligt då modellerna som behandlar fler klasser definerar om problemet till K-antalet binära klassificeringsproblem som i sin tur kan delas upp i en positiv respektive negativa klass. Jämförelsevis med makro-medelvärden ligger tyngdpunkten hos mikro-medelvärden vid prestandan per exempel. Detta medför att prestandan hos vanligt förekommande klasser är av större vikt än de mer sällsynta (Yang, 1997). I ekvationerna representerar k antalet klasser.

Precision

$$Makro - medelvärde = \frac{1}{k}\sum \frac{TP_i}{TP_i+FN_i} \quad (7)$$

$$Mikro - medelvärde = \frac{\sum TP_i}{\sum TP_i+\sum FN_i} \quad (8)$$

Känslighet

$$Makro - medelvärde = \frac{1}{k}\sum \frac{TP_i}{TP_i+FP_i} \quad (9)$$

$$Mikro - medelvärde = \frac{\sum TP_i}{\sum TP_i+\sum FP_i} \quad (10)$$

F-score är ännu ett utvärderingsmått som tillämpas inom klassificering vid sidan av precision och känslighet (Yang, 1997). I ett F-score värde kombineras precisionen och känsligheten i form av ett viktat medelvärde. Genom att beräkna en modells F-score är det möjligt att utvärdera två stycken aspekter. Dels exaktheten i modellen samt dess robusthet, det vill säga om den missar att klassificera en specifik typ eller del av data.

$$F - score = \frac{2\times precision\times känslighet}{precision+känslighet} \quad (11)$$

2.6.1 Permutation Feature Importance

Permutation Feature Importance (PFI) är en algoritm som används i syfte att utvärdera relevansen hos de olika attributen för maskininlärningsmodellen. Med andra ord hur stor inverkan ett specifikt attribut har på modellens slutgiltiga resultat. Exempelvis är det möjligt att ta bort attribut och bibehålla samma resultat för modellen som helhet. Genom att studera hur mycket varje attribut påverkar det slutgiltiga resultatet kan algoritmen poängsätta de valda attributens relevans. Det vill säga om attributet inte har någon effekt på resultatet så kommer PFI ge den en poäng och därifrån kan slutsatsen att attributet är redundant dras (Microsoft Azure ML Studio, 2018). För att ta reda på relevansen av varje attribut så beräknar först algoritmen ut värdet för varje prediktion med attributen som vanligt. Sedan väljer algoritmen ett attribut (t.ex. lokaltyp) för att slumpmässigt blanda och byta plats på datapunkterna men endast för det valda attributet. Resultatet jämförs sedan med det första resultatet som togs fram med det ursprungliga datasetet. Detta sker iterativt över alla kolumner för en i taget (Microsoft Azure ML Studio, 2018). Formeln definieras som:

$$PFI = P_b - P_s \qquad (12)$$

I ekvation (12) refererar P_b till korrektheten som erhålls med hjälp av det ursprungliga attributet medan P_s är korrektheten som erhålls från ett slumpmässigt valt attribut (Bleik , 2015). Det finns inget definerat intervall för vad PFI värdet kan anta utan desto högre värde det antar desto mer relevant är attributet.

3. Data

I detta avsnitt ges en redogörelse för den data som har samlats in. Syftet är att ge läsaren en grundläggande förståelse kring vilka datakällor som har används inför det efterföljande metodavsnittet där bland annat förbehandling och datakonsolidering beskrivs närmare.

Den data som har använts i modelleringen består av kontraktsinformation från 2847 stycken hyresgäster. Antalet hyreskontrakt uppgår i sin tur till 5626 till följd av att en del hyresgäster hyr mer än en lokal eller fastighet. Utöver kontraktsinformationen har finansiella nyckeltal för hyresgästerna och data kring snitthyror för olika områden också inkluderats. Nedan följer en närmare presentation av de datatkällor som har innefattats i studien.

Data från hyresavtal

Datan som kommer ifrån hyresavtalen består av vad som kallas för frysningar. Det vill säga att varje månad så fryser hyresföretaget alla kontrakt och sparar all information om kontrakten för den månaden. Detta har gjorts varje månad sedan databasen med kontrakt skapades. Detta innebär att samma kontrakt förekommer flera gånger men de olika attributen kan skilja sig åt då kontrakten förändras med tiden. Detta har resulterat i att datan från hyresavtalen innehåller cirka 170 000 datapunkter. Nedanför följer en tabell med alla attribut som följer från en frysning av ett kontrakt.

Tabell 1. *Översikt för de attribut som innefattas i datasetet avseende hyreskontrakt.*

ATTRIBUT	FÖRKLARING
ORGANISATIONSNUMMER	Organisationsnummer till företaget som är hyresgäst
PERIODKEY	Detta är datumet för månaden som denna frysning skedde och hur dessa attribut såg ut för kontrakten då.
AVTAL_ID	Ett unikt identifikationsnummer för kontraktet
AVTAL_KONTRAKT_FROM	Datumet då hyresgästen kontrakt börja gälla.
AVTAL_FORLANGT_TOM	Datumet som kontraktet gäller till.
AVTAL_AVISERING_TOM	Datumet då hyresgästen sa upp kontraktet.
AVTAL_UPPSAGNING_ORSAK	Anledning varför hyresgästen säger upp kontraktet
ARVS_TOTAL	Den årliga hyran som hyresgästen betalar.
ARSV_RABATT	Eventuell rabatt som hyresgästen kan ha erhållit.
LOKALTYP	Vad det är för typ av lokal som kontraktet gäller dvs lager, kontor etc.

Finansiell bolagsinformation

Den finansiella datan har hämtats för de företag som finns i datasetet innehållande hyresinformation. Denna information fanns dock endast för de senaste fem aktiva åren för företagen. Det vill säga om ett företag slutade vara aktiv 2016 så finns denna information endast för perioden 2011-2015. Den specifika informationen som hämtades

var vinst, omsättning, antalet anställda samt branschtillhörighet. Noterbart är att denna information baserades på siffror från koncernnivå, vilket i vissa fall kan bli missvisande då det kan vara ett mindre dotterbolag som är den aktuella hyresgästen. Nedan följer de attribut som hämtades för de senaste fem aktiva åren:

Tabell 2. Översikt för de attribut som innefattas i datasetet avseende finansiell bolagsinformation.

ATTRIBUT	FÖRKLARING
ORGANISATIONSNUMMER	Bolaget som informationen gäller för.
ÅR	Vilket år som datan avser.
VINST	Hur mycket företaget i fråga gjorde i vinst eller förlust det aktuella året. Anges i procentform.
OMSÄTTNING	Hur mycket företaget hade i omsättning (i svenska kronor)
ANTAL ANSTÄLLDA	Hur många som är anställda på företaget.
SNI	Vilken bransch företaget är aktivt inom.

Geografisk data

Då datan innehållande hyresavtalen inte anger vilken fastighet eller område som avses så har geografisk data hämtats för att urskilja de olika fastigheterna. För varje avtal så har specifik adress och område extraherats. Utöver detta har även information gällande snitthyran för respektive område inhämtats.

Tabell 3. Översikt för de attribut som innefattas i datasetet avseende geografisk data.

ATTRIBUT	FÖRKLARING
AVTAL_ID	Vilket avtal som följande information hänvisar till.
POSTADRESS	Vilken postnummer som lokalen eller fastigheten har.
ADRESS	Vilken adress lokalen finns på.
OMRÅDE	Vilken stadsdel som lokalen ligger i.

4. Metod

I metodavsnittet återfinns studiens avgränsningar samt en presentation av de programvaror som har använts. Vidare beskrivs tillvägagångssättet för hur data har insamlats och förbehandlas. Därefter följer en beskrivning av arbete kring att konsolidera olika datakällor samt hur maskininlärningsprocessen har utförts. Avslutningsvis presenteras ett reflekterande avsnitt i form av metodkritik.

4.1 Avgränsningar

I samband med att detta examensarbete genomfördes har flertalet avgränsningar gjorts för att begränsa dess omfång. Inledningsvis var det nödvändigt att begränsa studien på ett övergripande plan för att möta tidsaspekten om 20 veckor samt för att på ett så tydligt sätt som möjligt kunna uppfylla arbetets syfte. Utifrån detta har sedan ytterligare avgränsningar arbetats fram, vilka presenteras nedan.

Generella avgränsningar

Utgångspunkten för studien är att använda data avseende kontraktsinformation mellan ett kommersiellt fastighetsbolag och dess hyresgäster. Av naturliga skäl är därför en initial avgränsning att endast studera data som innehåller element från dessa hyreskontrakt. Vidare fanns en ursprunglig idé om att avgränsa studien till att endast innefatta kontrakt som avsåg kontorslokaler. Detta hade dock bidragit till en markant reducering av det totala datasetet, vilket la grunden för beslutet att innefatta alla typer av lokaler.

Attributavgränsningar

Valen av attribut i termer av antal och vilka typer som ansetts vara relevanta för modelleringen har baserats på flertalet faktorer. Inledningsvis filtrerades en del attribut bort till följd av att de saknades ett för stort antal datapunkter inom attributet i fråga. Genom att filtrera bort dessa typer av attribut kunde kvalitén på de datapunkter som slutligen användes i modelleringen bibehållas på en högre nivå. I Microsoft ML Studio har verktyget PFI använts för att finna den optimala uppsättningen av attribut. I korthet bygger PFI (utförligare beskrivet i 2.6.1 Permutation Feature Importance) på att de tillgängliga attributens relevans rankas i förhållande till vald modell och dataset. I föregående kapitel 3 Data presenteras de attribut som slutligen användes i maskininlärningsmodelleringen.

Modellavgränsingar

De maskininlärningsalgoritmer som återfinns i studien har begränsats till ett urval av de mest frekvent förekommande algoritmerna inom klassificeringsproblem. Modellavgränsningen innebär därför att algoritmerna Boosted decision tree, Decision forest, Multiclass decision forest samt Multiclass neural network har studerats. Vid en mer djupgående studie hade denna avgränsning varit möjlig att utöka i flera dimensioner. Dels hade flera algoritmer inom klassificering kunnat inkluderats. Vidare skulle det även utifrån frågeställning varit intressant att utöka studien till att undersöka regressionsmodeller då de i vissa fall kan ge ett mer precist svar jämfört med algoritmerna inom klassificering.

4.2 Val av programvara

Under arbetets gång har huvudsakligen tre stycken programvaror används, vilka beskrivs mer utförligt nedan. Valet av just dessa programvaror grundar sig främst i att Business Vision, som arbetet har genomförts tillsammans med, använder sig Microsofts produkter och därmed kunde vara behjälpliga vid eventuella frågor och problem. I korthet har Excel och SQL Server Management Studio används för att sammanställa och behandla data, för att sedan överföras till Azure Machine Learning Studio där modelleringen har utförts.

4.2.1 Microsoft Excel

Microsoft Excel är en del av officepaketet och används till att utföra beräkningar och lättare datahanteringsuppgifter. Inom ramen för den här studien har Excel fyllt en viktig funktion i flera avseenden. Data från olika källor har lagrats och förbehandlats var för sig i olika Excelfiler. Vidare har har Excel även använts för att beräkna finansiella nyckeltal baserat på ursprungsdatan samt fungerat som lagringsplats för såväl rådata som färdiga dataset.

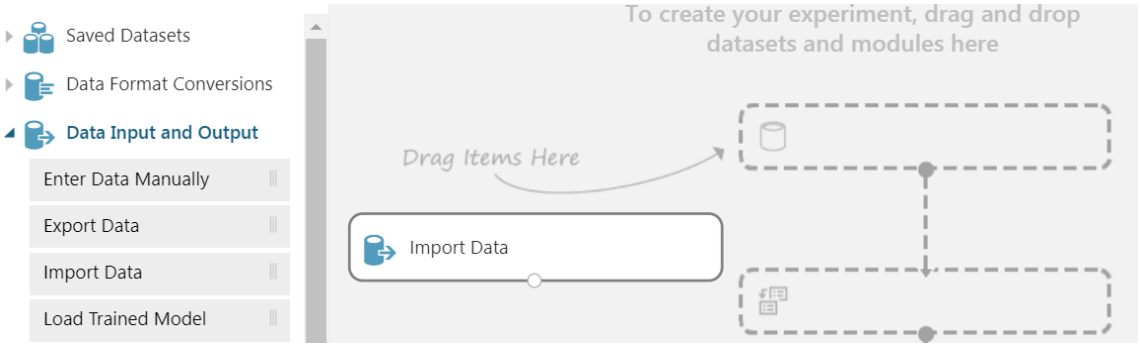
4.2.2 Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) är ett verktyg som gör det möjligt att ansluta, konfigurera, hantera och utveckla alla komponenter i en SQL *(Structured Query Language)* server. SSMS har bland annat använts till att filtrera data och sammanfoga Excelfiler. När arbetet i SSMS var avklarat så sparades datan återigen ner till en Excelfil

då det formatet visade sig vara fördelaktigt vid exporteringen till Microsoft Machine Learning Studio jämfört med att ansluta en SQL-databas.

4.2.3 Microsoft Azure Machine Learning Studio

Azure Machine Learning Studio är en del av Microsoft Azure som samlar samtliga av Microsofts molnbaserade tjänster på samma plattform. Azure Machine Learning Studio *(Azure ML Studio)* erbjuder in sin tur användaren en miljö för att utföra prediktiva analyser utifrån sin data. Detta sker med hjälp av ett så kallat *drag and drop* verktyg som möjliggör byggandet, test och distribuering av prediktiva analysmodeller (Microsoft, 2018).



Figur 7. Illustration av Microsoft Azure ML Studios drag and drop verktyg

4.3 Extrahering av data

4.3.1 Data från hyresavtal

Data avseende kontraktsinformation mellan hyresvärden och dess hyresgäster finns sparade i en Excelfil. Hela datasetet laddades upp i en SQL-databas där relevanta parametrar filtrerades ut med hjälp av SQL-queries. Den återstående datan sparades slutligen ner i en CSV-fil, vilket är ett filformat som är kompatibelt med Microsoft Azure ML Studio.

4.3.2 Finansiell bolagsdata

Data om hyresgästernas finansiella information kommer huvudsakligen från en samarbetspartner till Business Vision. För de hyresgäster där informationen var ofullständig genomfördes en manuell komplettering med hjälp av tjänsten Allabolag.se som tillhandahåller finansiell information om svenska företag. Den kompletterande informationen fördes slutligen in i samma Excelfil som den övriga finansiella informationen.

4.3.3 Geografisk data

Data gällande hyresgästernas geografiska position erhölls i en separat Excelfil från hyresvärdens databas. Denna data kompletterades sedan med ytterligare information avseende snitthyror för respektive område med hjälp av tjänsten *yta.se:s* sökfunktion.

4.4 Förbehandling av data

Till följd av att datan som har inkluderats i studien var av varierande kvalitet och form har den i viss utsträckning förbehandlats inför modelleringen i Azure ML Studio. Detta arbete utfördes i Microsoft Excel och SQL Management Studio.

4.4.1 Data från hyresavtal

Data gällande hyresavtalen förbehandlades både i Microsoft Excel och SQL Management Studio. Arbetet i Excel resulterade i att ytterligare kolumner skapades för att förtydliga datan. Den första kolumnen som lades till var hur många månader som hyresgästerna redan hade hyrt respektive lokal. Detta gjordes eftersom att varje kontrakt har en frysning varje månad varav de första frysningar skapades 2008. Det finns dock kontrakt med äldre historik och som tecknades redan 1997. Detta innebar att antalet månader från att kontraktet skapades till varje frysning beräknades och lades till med hjälp avgjordes i Excel med hjälp av villkorsstyrda beräkningar. Den andra kolumnen som skapades var antalet månader som det var kvar innan hyresgästen sa upp kontraktet. Tack vare att det redan fanns en kolumn som specificera när kontrakten sades upp så blev det möjligt att

beräkna vid varje frysning hur många månader det var kvar innan hyresgästen skulle säga upp kontraktet.

Från kolumnen med antalet månader kvar till uppsägning kunde olika grupper skapas i ytterligare en kolumn. Denna kolumn sammanställer alla frysningar inom olika grupper med ett sex-månaders intervall. Vilket innebär att alla kontrakt som sägs upp inom sex månader hamnar i gruppen *6*, alla kontrakt som sägs upp mellan 6 till 12 månader hamnar i gruppen *12* osv upp till 43 månader. Den sista gruppen *43* innehåller även de kontrakt om 43 månader eller mer. Det skapades även en kolumn med tre månaders intervall vilket hade klasser upp till 24 månader där resterande frysningar hamna i klassen *25*. Den sista kolumnen som lades med klasser var en kolumn med binära klasser. Där grupperades alla kontrakt som avslutades inom ett år i klass *1* och resterande i klass *2*. Även detta gjordes i Excel med hjälp av IF-satser.

Tabell 4. Klassindelning för tre-månaders intervall

KLASS	**3**	**6**	**9**	**12**	**15**	**18**	**21**	**24**	**25**
ÅTERSTÅENDE KONTRAKTSTID (MÅNADER)	*1-3*	*4-6*	*7-9*	*10-12*	*13-15*	*16-18*	*19-21*	*22-24*	*25≤*

Tabell 5. Klassindelning för sex-månaders intervall

KLASS	***6***	***12***	***18***	***24***	***30***	***36***	***42***	***43***
KONTRAKTSTID (MÅNADER)	*1-6*	*7-12*	*13-18*	*19-24*	*25-30*	*31-36*	*37-42*	*43≤*

Tabell 6. Klassindelning för under respektive över ett år

KLASS	***1***	***2***
ÅTERSTÅENDE KONTRAKTSTID (ÅR)	$1 \geq år$	$1 \leq år$

Anledningen till dessa olika tillvägagångssätt för att dela upp klasserna är för att få olika grader av precision av prediktionen. Det vill säga grupperna som har tre-månaders intervall är väldigt specifik och om denna klass skulle predikteras rätt innebär det att kontraktet kommer sägas upp väldigt snart. Resultaten från sex-månaders intervallet är däremot inte lika specifik då en rätt prediktion betyder att kontraktet kan sägas upp när som helst inom 6 månader. Fördelen med detta intervall är dock att fördelningen mellan klasserna i datasetet blir bättre. Detta gäller även den sistnämnda grupperingen som ska prediktera om ett kontrakt sägs upp inom 1 år eller mer. Intervallet är väldigt stort men spridningen blir betydligt bättre samt detta är den informationen som fastighetsföretagen är mest intresserad av.

Kolumnerna *hyra/kvm* och *hyreshöjning* lades även till. Hyreshöjningen beräknades genom att studera om hyran ändrades mellan olika frysningar för samma kontrakt. Om detta var fallet så beräknades den procentuella hyreshöjningen ut. Hyran per kvadratmeter beräknades med hjälp av kolumnerna som fanns i ursprungsdatan, vilket visade ytan som varje hyresgäst hade hyrt samt den totala årshyran. För att underlätta arbetet i SQL skapades även en ny kolumn som endast innehöll året för varje frysning. Detta gjordes då varje frysning är gjord månadsvis och kolumnen med denna information innehåller hela datumet. Det som dock är relevant för att senare kombinera det med finansiell data är endast året. Slutligen togs kolumnen som innehöll anledning till varför kontrakten sades upp bort då denna varken var komplett eller tydlig.

4.4.2 Finansiell bolagsdata

Den finansiella datan innehöll bland annat information om omsättning, vinstmarginal och antal anställda för respektive bolag, vilket illustrerades över en femårsperiod. Utifrån parametrarna beräknades sedan tillväxten för dessa nyckeltal i Excel. Genom att införa

ett tillväxtmått i procent underlättades jämförelser mellan olika företag då exempelvis antalet anställda mellan två företag kan skilja sig markant, samtidigt som tillväxten i procent från år till år är mer homogen. Avslutningsvis filtrerades majoriteten av de icke aktiva bolagen bort då det saknades fullständig historik.

4.4.3 Geografisk data

I den geografiska datan utfördes ingen förbehandling utöver vad som har beskrivits i avsnittet extrahering av data. Motiveringen till beslutet återfinns i att datan redan fanns sparad i en Excelfil som innehöll nödvändiga attribut för att sammanfoga den med de andra datakällorna.

4.5 Datakonsolidering

För att sammanställa all data som har hämtats har datan för hyresavtalen, finansiella datan och geografiska datan laddats upp i Microsoft SMSS. Denna åtgärd vidtogs för att på ett smidigt sätt sammanställa all data till en gemensam tabell där informationen från de olika dataseten matchas ihop med rätt hyresavtal. Den första sammansättningen var mellan hyresavtalen och den finansiella datan. I och med att ett flertal kontrakt kunde vara kontrakt som gällde i slutet av 1990 början av 2000-talet så fanns det även företag som det inte gick att hitta finansiell data till. Detta resulterade i att alla företag som det inte fanns data till filtrerades bort när den finansiella datan sammanställdes med hyreskontrakten. Den finansiella informationen var endast dokumenterade för en femårsperiod så detta gjorde även att alla frysningar som inte var inom denna period filtrerades bort.

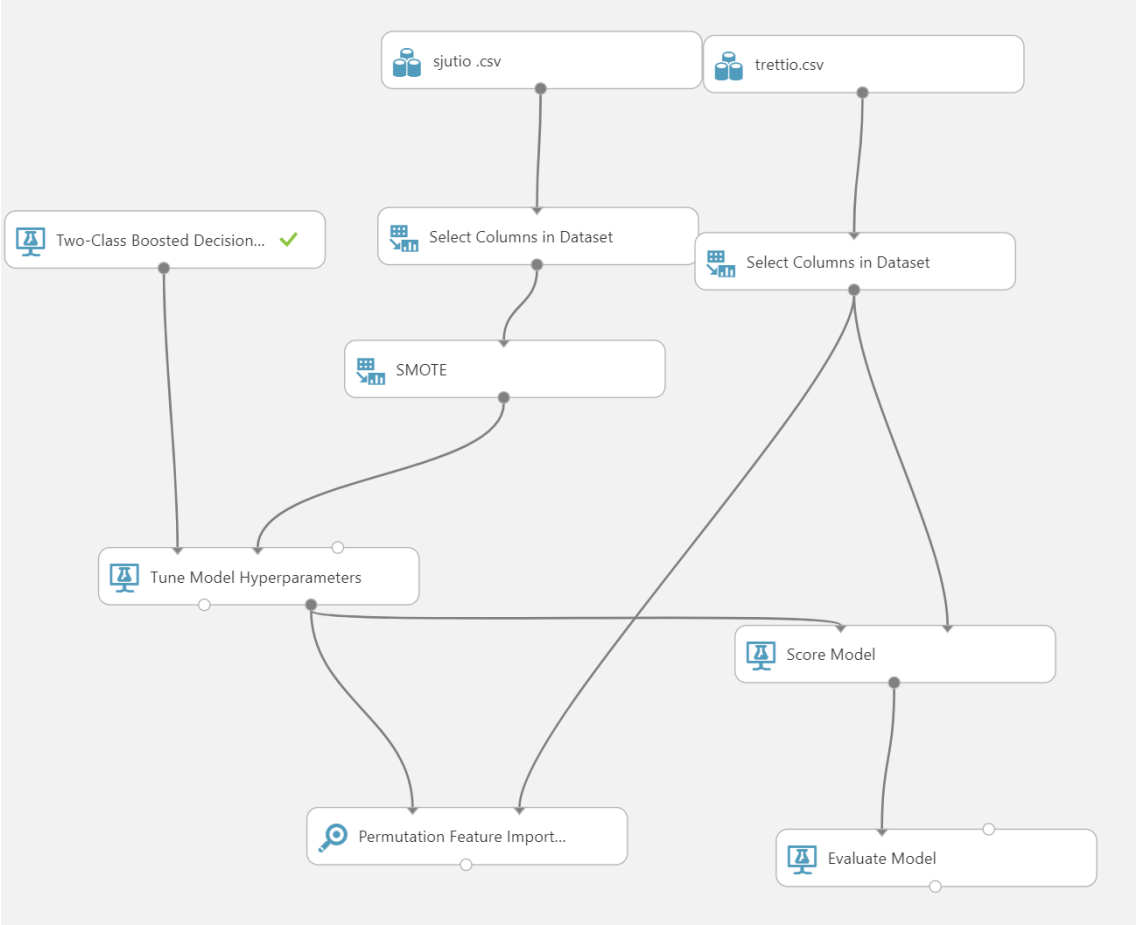
Då den finansiella informationen angavs på årsbasis så matchades den ihop med varje frysning för motsvarande år. Slutligen sammanfogades datan även med den geografiska datan, vilket medförde att all information återfanns på samma plats. Informationen kopierades därefter till en Excelfil innehållande cirka 9000 datapunkter.

I det avslutande steget delades filen sedan till två olika filer, där den ena innehöll cirka 70 procent av datan vilket skulle användas som träningsdata och den andra innehöll de resterande 30 procenten som representerar testsdatan. I Azure ML Studio finns moduler som kan dela upp ett dataset i tränings- respektive testdata. Det tillvägagångssättet medförde dock att ett specifikt hyreskontrakt kunde placeras i både tränings- och testdatan

till följd av sättet som datan hade lagrats. Därför utfördes uppdelningen manuellt genom att två stycken separata filer skapades i Excel. För att göra det möjligt att ladda upp dessa filer till Azuremiljön transformades filerna till filformatet CSV.

4.6 Modellering

Efter förbehandlingen och konsolideringen av datasetet var nästa steg att utföra modelleringen i Azure ML Studio. I figur 8 nedan illustreras en av totalt åtta stycken modeller som har studerats. Denna figur tjänar som exempel för samtliga modeller då uppbyggnaden är identisk bortsett från vilken algoritm som används samt modulen SMOTE, vars innebörd beskrivs utförligare längre fram i detta avsnitt. I appendix återfinns en översikt för samtliga modeller. Nedan följer en närmare beskrivning av vilken funktion som respektive modul fyller bortsett från de olika algoritmerna som beskrivits tidigare.



Figur 8. *Översikt för en av maskininlärningsmodellerna som skapats i Azure ML Studio.*

Select Columns

Den första modulen som används är *select columns*, denna modul gör det möjligt att utesluta olika parametrar (kolumner) om de anses ge en negativ effekt på modellen. Detta gjordes med hjälp av PFI som undersökte alla parametrar för att se vilka som inte var nödvändiga. Detta innebär dock att den första körningen måste göras med alla parametrar för att sedan med hjälp av informationen som ges av PFI modulen gå tillbaka till *select columns* och utesluta dessa attribut. Detta gjordes både för tränings- och testdatan då modellen måste ha identiska attribut kategorier för att fungera (Microsoft Azure ML Studio, 2018).

SMOTE

Nästa steg för datan är SMOTE (Synthetic Minority Oversampling Technique) modulen för att få den obalanserade datan att bli mer balanserad. SMOTE-modulen skapar nya datapunkter av den underrepresenterade klassen genom att kombinera parametrar från olika datapunkter som är inom samma klass. Kolumnen *inom 1 år eller mer* valdes i denna modul då klassen *1* som representerade att ett kontrakt skulle sägas upp inom ett år var underrepresenterad. Denna klass innehöll endast cirka 2000 datapunkter medan klass *2* innehöll cirka 4000. Med hjälp av SMOTE kunde klass *1* öka med 2000 datapunkter så de båda klasserna innehöll 4000 datapunkter var. SMOTE modulen enbart på träningsdatan så alla modeller som har tränats med binära klasser ska kunna jämföras (Microsoft Azure ML Studio, 2018).

Tune Hyperparameter

Kedjan som innehåller träningsdatan kopplades efter SMOTE modulen vidare in i *Tune hyperparamete*r modulen samt till denna modul kopplades även träningsalgoritmen. Denna modul användes för att få fram de bästa parametrarna för träningsalgoritmen för datasetet. Modulen tränar algoritmen i flera iterationer där den testar olika parametrar för att sedan välja den iterationen som tränade modellen bäst. I modulen var även kolumnen som datan ska klassificera tvungen att specificeras samt så valdes inställningen att modulen skulle iterera över alla parametrar (Entire grid) då den är förinställd på att bara hoppa runt och testa olika (Microsoft Azure ML Studio, 2018).

Permutation Feature Importance

Modulen beräknar relevansen för varje attribut i datasetet och skapar på så sätt ett underlag för vilka attribut som bör användas för en given modell och träningsdataset. För att kunna använda denna modul så tränas modellen först en gång med alla attribut. Efter den första träningen är gjord så kan attributen relevans studeras med hjälp av PFI och attribut kan därefter uteslutas från modellen med hjälp av *select columns* modulen. En mer detaljerad beskrivning finns att läsa i avsnitt 2.6.1 Permutation Feature Importance (Microsoft Azure ML Studio, 2018).

Score Model

I denna modul möts den tränade modellen och testsdatan. Efter träningsalgoritmen har tränats på den träningsdatan och tagit fram en model så utvärderas den på testsdatan. *Score model* modulen försöker med hjälp av tränade modellen att prediktera vilken klass de olika datapunkterna tillhör. Modulen redovisar även med vilken sannolikhet den anser att gissningen stämmer (Microsoft Azure ML Studio, 2018).

Evaluate Model

Modulens syfte är att skapa underlag för att utvärdera hur väl modellen presterar. Resultaten som modulen genererar är bland annat Recall, Precision och Accuracy, vilka har beskrivits närmare under avsnittet Modellutvärdering i bakgrundskapitlet. Modulen genererar dessa mått genom att analysera prediktionerna som har gjorts i *Score model* och jämför sedan dessa med den faktiska klassen som datapunkter tillhör (Microsoft Azure ML Studio, 2018)

4.7 Metodkritik

Antalet ögonblicksbilder, det vill säga antalet datapunkter, var cirka 170 000 stycken initialt. Trots att ambitionen att bibehålla ett så stort dataset som möjligt har dess storlek reducerats till följd av olika faktorer. Bland annat på grund av datakvaliten som försämrats av att en del av attributen inte fanns representerade i alla datapunkter. En annan aspekt var att datasetet var obalanserat (detta åtgärdades dock med hjälp av SMOTE), vilket innebär att vissa typer av data var överrepresenterade. Med ett sådant dataset finns

risken att modellen ska bli övertränad och därmed tappa sin funktion vid modellering av tidigare osedd data.

Den rådata som extraherades inledningsvis baserades på antaganden om vilken typ av data som skulle kunna vara intressant med studiens syfte och frågeställning i åtanke. De utvalda attributen är inte nödvändigtvis de parametrar som på bäst sätt kan återspegla varför en hyresvakans uppstår. Däremot kan de sammantaget ses som en rimlig utgångspunkt då det saknas liknande studier att förhålla sig till. Vid ett alternativt tillvägagångssätt hade all tillgänglig data gällande hyreskontrakt kunnat extraheras för att på så sätt skapa ett så brett underlag som möjligt. Därefter hade PFI-modulen varit ett lämpligt verktyg för att sortera ut relevanta attribut, istället för att tillämpa den först efter att en första utgallring genomförts. Samtidigt kan ett alltför stort dataset påverka exekveringstider och hanterbarheten i en negativ riktning.

Det slutgiltiga datasetet bestod av cirka 9000 datapunkter, vilket är en markant minskning jämfört med de 170 000 som fanns tillgängliga inledningsvis. Trots detta bör reduceringen ses i förhållande till att denna data håller en högre kvalitet i många avseenden. Bland annat innehåller den flera relevanta attribut i form av bland annat finansiella nyckeltal och geografisk information. Vidare har ofullständiga attribut sorterats bort och obalansen har justerats.

5. Resultat

Detta avsnitt innehåller de resultat som har genererats från de undersökta maskininlärningsmodellerna i Microsoft Azure ML Studio. Inledningsvis presenteras en översikt för de olika modellerna. Därefter följer resultaten för de binära modellerna i form av Boosted Decision Tree och Decision Forest. Slutligen återfinns resultaten för Multiclass Neural Network och Multiclass Decision Forest.

Tabell 7. Modellöversikt innefattande algoritmtyp och klassindelning

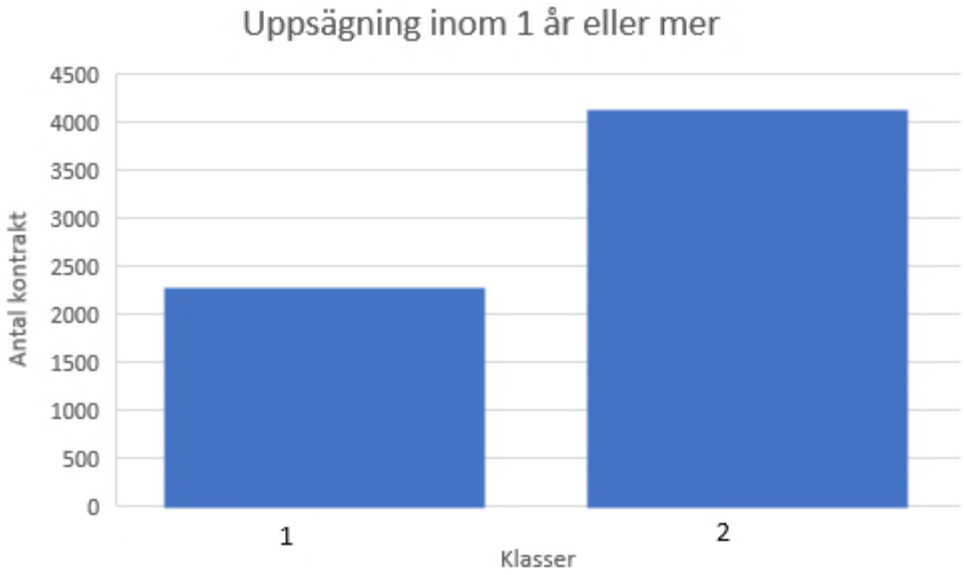
Modell	**Algoritm**	**Klasser i datasetet**	**Tabellreferens**
1	*Boosted Decision Tree*	*Över/under ett år*	*6*
2	*Decision Forst*	*Över/under ett år*	*6*
3	*Boosted Decision Tree*	*Över/under ett år med SMOTE*	*6*
4	*Decision Forest*	*Över/under ett år med SMOTE*	*6*
5	*Multiclass Neural Network*	*Tre-månaders interval*	*4*
6	*Multiclass Neural Network*	*Sex-månaders intervall*	*5*
7	*Multiclass Decision Forest*	*Tre-månaders intervall*	*4*
8	*Multiclass Decision Forest*	*Sex-månaders intervall*	*5*

5.1 Binära modeller

Modellering utfört innan balansering med SMOTE

Den binära modelleringen har som tidigare nämnt genomförts på klasserna *1* och *2*. *Klass 1* innefattar alla kontrakt som sägs upp inom ett år och resterande kontrakt hamnar i klass *2*. Nedan i figur 9 visas fördelningen mellan de två klasserna där det finns strax under 2500 kontrakt tillhörande klass *1* och cirka 4000 kontrakt som tillhör klass *2*. Det vill säga att fördelningen är obalanserad enligt resonemanget i avsnitt 2.4.1.

För att kunna presentera samtliga utvärderingsmått (kapitel 2.6) har båda klasserna använts som den positiva klassen vid enskilda uträkningar för respektive klass. Noterbart är dock att klass 2 har valts som den positiva klassen i de korstabeller som presenteras.



Figur 9. Fördelningen för träningsdata på kontrakt med de två klasserna 1 och 2 före balansering.

I tabellerna åtta och nio nedan presenteras utfallet från modelleringen med algoritmen *Boosted Decision Tree* (modell 1) och *Decision Forest* (modell 2) för det obalanserade datasetet. Resultaten från de båda algoritmerna kommer från två separata körningar i den maskininlärningsmodell som har presenterats tidigare i figur 7. I fallet för *Boosted Decision Tree* har modellen angett en korrekt klassetikett vid totalt 2007 (*sant positiv + sant negativ)* tillfällen och en felaktig vid 938 (*falskt positiv + falskt negativ)* stycken. Modellens känslighet uppgår till 51% för *klass1* respekive 75% för *klass 2* enligt ekvation (6). Vidare uppmättes precisionen till 46% för *klass 1* samt 79% för *klass 2* enligt ekvation (5).

Tabell 8. Korstabell för Boosted Decision Tree utan SMOTE

	Klassificerad som över ett år	***Klassificerad som under ett år***
Faktisk klass över ett år (klass 2)	*Sant positiv* *1572*	*Falskt negativ* *529*
Faktisk klass under ett år (klass 1)	*Falskt positiv* *409*	*Sant negativ* *435*

För *Decision Forest* visar resultatet att en korrekt klassificering har utförts vid totalt 2225 tillfällen och en felaktig vid 720 tillfällen efter att balanseringen med SMOTE-algoritmen hade utförts. Det totala antalet klassificeringar som utfördes var 2945 stycken vid båda modelleringarna. Modellens känslighet var 46% för *klass 1* samt 87% för *klass 2*. Precisionen motsvarades av 60% respektive 80% för de bägge klasserna.

Tabell 9. Korstabell för Decision Forest utan SMOTE

	Klassificerad som över ett år	***Klassificerad som under ett år***
Faktisk klass över ett år (klass 2)	*Sant positiv* *1827*	*Falskt negativ* *265*
Faktisk klass under ett år (klass 1)	*Falskt positiv* *455*	*Sant negativ* *398*

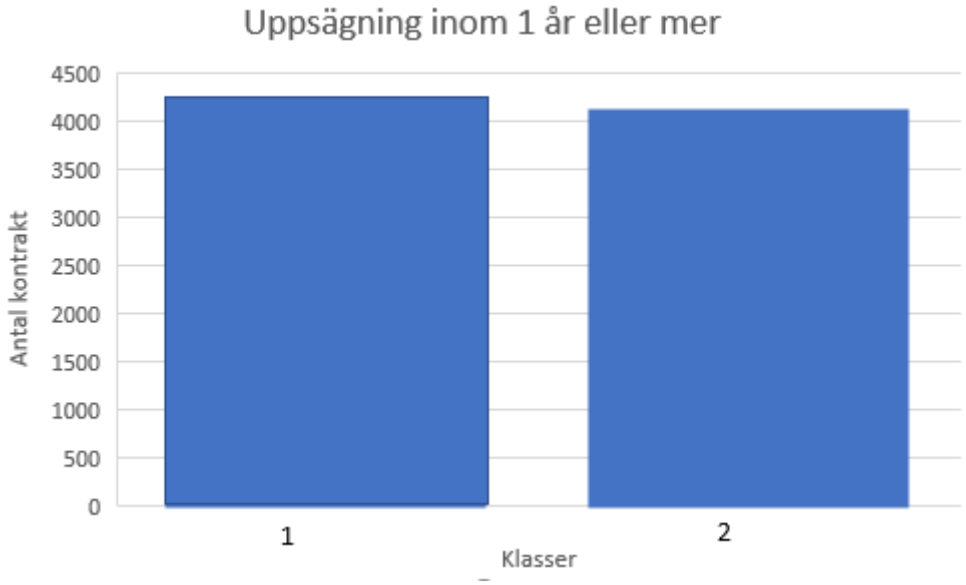
För att utvärdera algoritmernas prestanda används de kvalitetsmått som har presenterats tidigare i metodavsnittet. Överlag uppvisas en likvärdig prestanda, främst gällande korrekthet och precision. Gällande känsligheten återfinns ett högre värde för *Boosted Decision Tree* samtidigt som *Decision Forest* presterar bättre när det kommer till F-score.

Tabell 10. Översikt för kvalitetetsmåtten hos de binära modellerna för det obalanserade datasetet

Kvalitetsmått	**Boosted Decision Tree**		**Decision Forest**	
	Klass 1	*Klass 2*	*Klass 1*	*Klass 2*
Korrekthet	0,68	0,68	0,75	0,75
Precision	0,46	0,79	0,60	0,80
Känslighet	0,51	0,75	0,46	0,87
F-score	0,48	0,77	0,52	0,83

Modellering utfört efter balansering med SMOTE

För att balansera upp träningsdatan har syntetiska datapunkter skapats med hjälp av *SMOTE* vilket har resulterat i att klass *1* har strax över 4000 kontrakt och klass *2* har kvar cirka 4000 kontrakt. Detta illustreras i figur 10 nedanför.



Figur 10. Fördelningen för träningsdata på kontrakt för klasserna 1 och 2 efter balansering med SMOTE.

I tabell 11 och 12 redovisas resultaten för samma algoritmer som används för det obalanserade datasetet, denna gång så har klasserna dock balanserats med hjälp av *SMOTE.* Resultaten visar att modellerna presterar på liknande sätt gentemot varandra. Den stora skillnaden jämfört med modelleringen med det obalanserade datasetet är att modellerna som har tränats med det balanserade datasetet har presterat bättre för *klass 1* för samtliga utvärderingsmått bortsett från precisionen i modell 2. Modellen som har tränats med *Boosted Decision Tree* (modell 3) samt med den balanserade datan har en känslighet om 53% för *klass 1* och 80% för *klass 2,* vilket är en förbättring i båda fallen jämfört med det obalanserade datasetet (tabell 10). Resultaten har även förbättrats gällande precisionen, vilket går att utläsa vid en jämförelse mellan tabellen 10 och tabell 13 för det balanserade datasetet.

Tabell 11. Korstabell för Boosted Decision Tree efter balansering med SMOTE

	Klassificerad som över ett år	***Klassificerad som under ett år***
Faktisk klass över ett år	*Sant positiv* *1676*	*Falskt negativ* *425*
Faktisk klass under ett år	*Falskt positiv* *402*	*Sant negativ* *451*

Modellen som presenteras i tabell 12 som har tränat med algoritmen *Decision forest* (modell 4) har en känslighet om 56% för klass *1* och 82% för *klass 2*. Detta är en förbättring jämfört med det obalanserade datasetet för klass *1* då den modellen prediktera rätt för 46% av de som tillhörde klass *1*.

Tabell 12. Korstabell för Decision Forest med SMOTE

	Klassificerad som över ett år	***Klassificerad som under ett år***
Faktisk klass över ett år	*Sant positiv* *1712*	*Falskt negativ* *383*
Faktisk klass under ett år	*Falskt positiv* *362*	*Sant negativ* *491*

I tabell 13 visas en sammanställning över kvalitetsmåtten för de två algoritmerna. Överlag presterar modellerna på ett likvärdigt sätt, med en viss fördel för respektive algoritm beroende på vilket utvärderingsmått som studeras.

Tabell 13. Översikt för kvalitetétsmåtten hos de binära modellerna efter SMOTE

Kvalitetsmått	**Boosted Decision Tree**		**Decision Forest**	
	Klass 1	*Klass 2*	*Klass 1*	*Klass 2*
Korrekthet	0,72	0,72	0,75	0,75
Precision	0,51	0,81	0,56	0,82
Känslighet	0,53	0,80	0,58	0,82
F-score	0,52	0,80	0,57	0,82

Attribututvärdering med PFI

Som tidigare har beskrivits är PFI en metod för att utvärdera de valda attributens relevansen för modellens resultat. Till följd av att ett stort antal attribut har inkluderats i modelleringen presenteras endast de fem attribut som har högst relevans för respektive modell nedan. För *Boosted Decision Tree* är det således attributet *Avtal_Forlangt_Tom* som har högst relevans och för *Decision Forest* ges den högsta relevansen i form av attributet *Periodkey*. Det förstnämnda attributet avser datumet som kontraktet gäller till och det andra syftar till datumet när en så kallad frysning sker (se förklaringar i tabell 1).

Tabell 14. *PFI-värden för de fem attribut med högst relevans för Boosted Decision Tree*

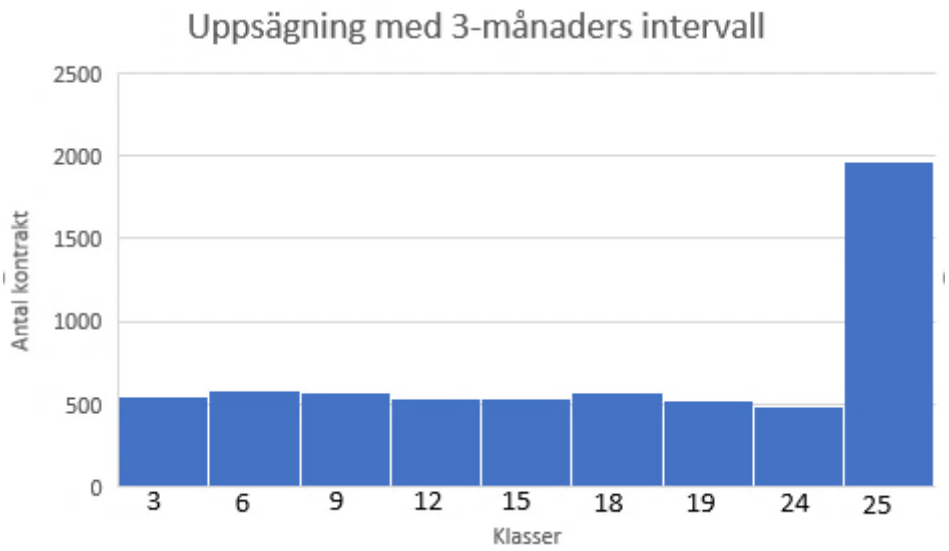
Attribut	**PFI-värde före balansering**	**Attribut**	**PFI-värde efter balansering**
Avtal_Forlangt_Tom	0,090	*Avtal_Forlangt_Tom*	0,097
Lokaltyp	0,056	*PeriodKey*	0,092
Periodkey	0,033	*Lokaltyp*	0,030
Avtal_Kontrakt_from	0,030	*Year*	0,017
hyrda_manader	0,014	*Avtal_Kontrakt_From*	0,016

Tabell 15. PFI-värden för de fem attribut med högst relevans för Decision Forest

Attribut	PFI-värde före balansering	Attribut	PFI-värde efter balansering
PeriodKey	0,132	*PeriodKey*	0,092
Avtal_Forlangt_Tom	0,093	*Avtal_Forlangt_Tom*	0,009
Lokaltyp	0,048	*Lokaltyp*	0,022
Hyrda_manader	0,034	*year*	0,017
Sni_a_swe_name	0,024	*Ö/U snitthyra*	0,001

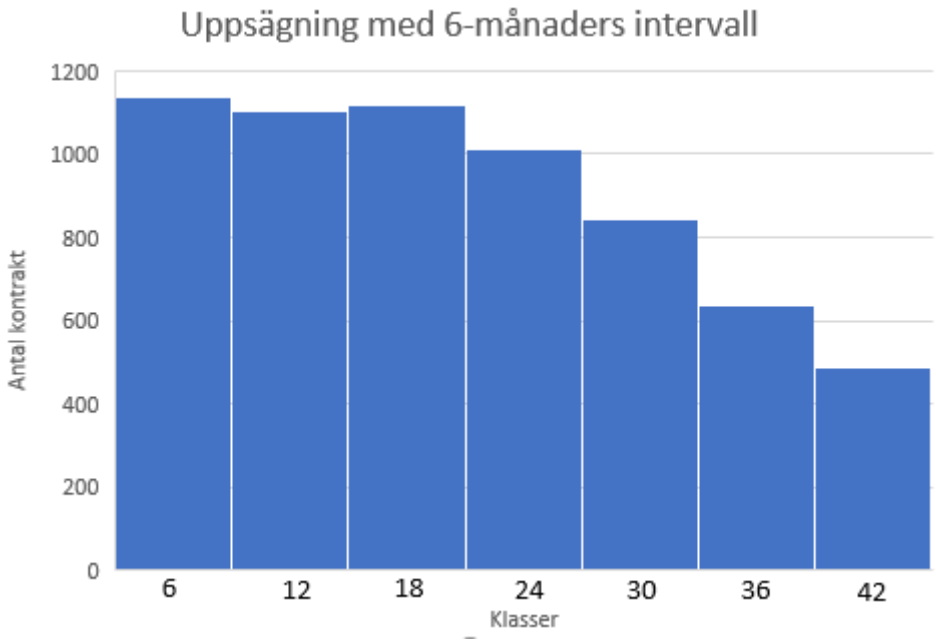
5.2 Multiklass modeller

Modelleringen med de så kallade multiklass-algoritmerna har i enlighet med de binära algoritmerna utförts i modellen som har presenterats i figur 8. I övrigt följde en något annorlunda struktur jämfört med de binära modellerna. Dessa modeller testades på två olika versioner av datasetet där skillnaden låg i uppbyggnaden av klasserna. I den första versionen delades alla kontrakt in i grupper av tre månaders intervall upp till två år där resterande kontrakt hamnade i klass *25*. Denna fördelning illustreras i figur 11 nedan där indelningen på x-axeln ska läsas som 3 (0-3 månader), 6 (4-6 månader) och så vidare:



Figur 11. Fördelningen för träningsdata på kontrakt med 3 månaders intervall för tidsperioden 2 år. Klass 3 innebär 0-3 månader, klass 6 innebär 4-6 månader etc.

Den andra versionen av detta dataset innehöll samma kontrakt men där kontrakten hade grupperats inom samma klass med ett sex månaders intervall upp till 3.5 år. Fördelningen är överlag bättre för denna version. Detta visas i figur 12.



Figur 12. Fördelningen för träningsdata på kontrakt med 6 månaders intervall för tidsperioden 3,5 år. Klass 6 innebär 0-6 månader, klass 12 innebär 7-12 månader etc.

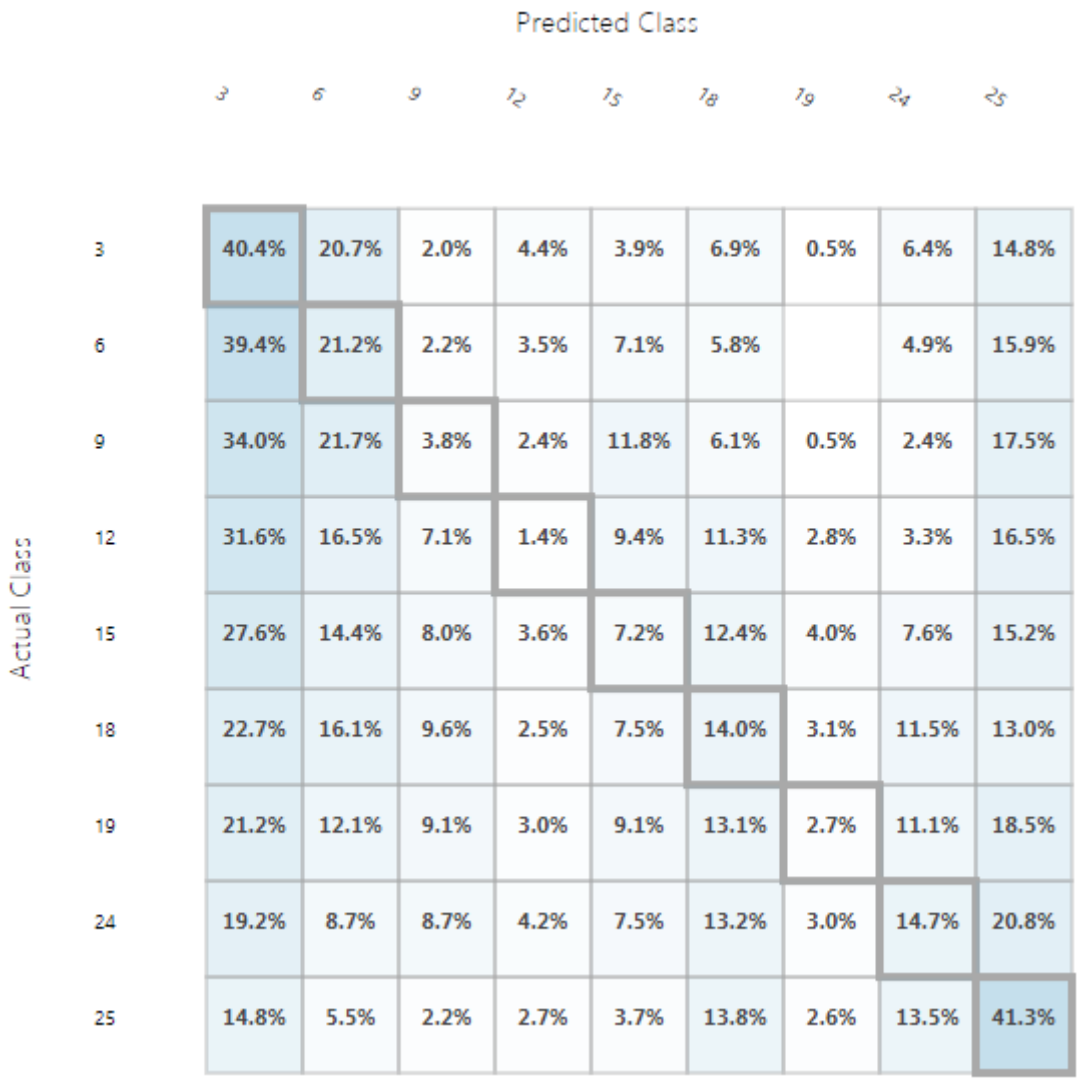
Multiclass Neural Network

I tabell 16 återfinns resultatet för algoritmen *Multiclass Neural Network* vid modelleringar innehållande tre (modell 5) respektive sex-månaders intervall (modell 6). Utifrån tabellen går det att utläsa hur algoritmen presterar bättre enligt samtliga utvärderingsmått, bortsett från den genomsnittliga korrektheten, för sex-månaders intervallet. I tabellen förekommer termerna övergripande samt genomsnittlig precision. Den övergripande precisionen avser antalet korrekt utförda prediktioner dividerat med det totala antalet prediktioner som utförts. Genomsnittlig precision innebär i sin tur summan av korrektheten i respektive klass dividerat med antalet klasser.

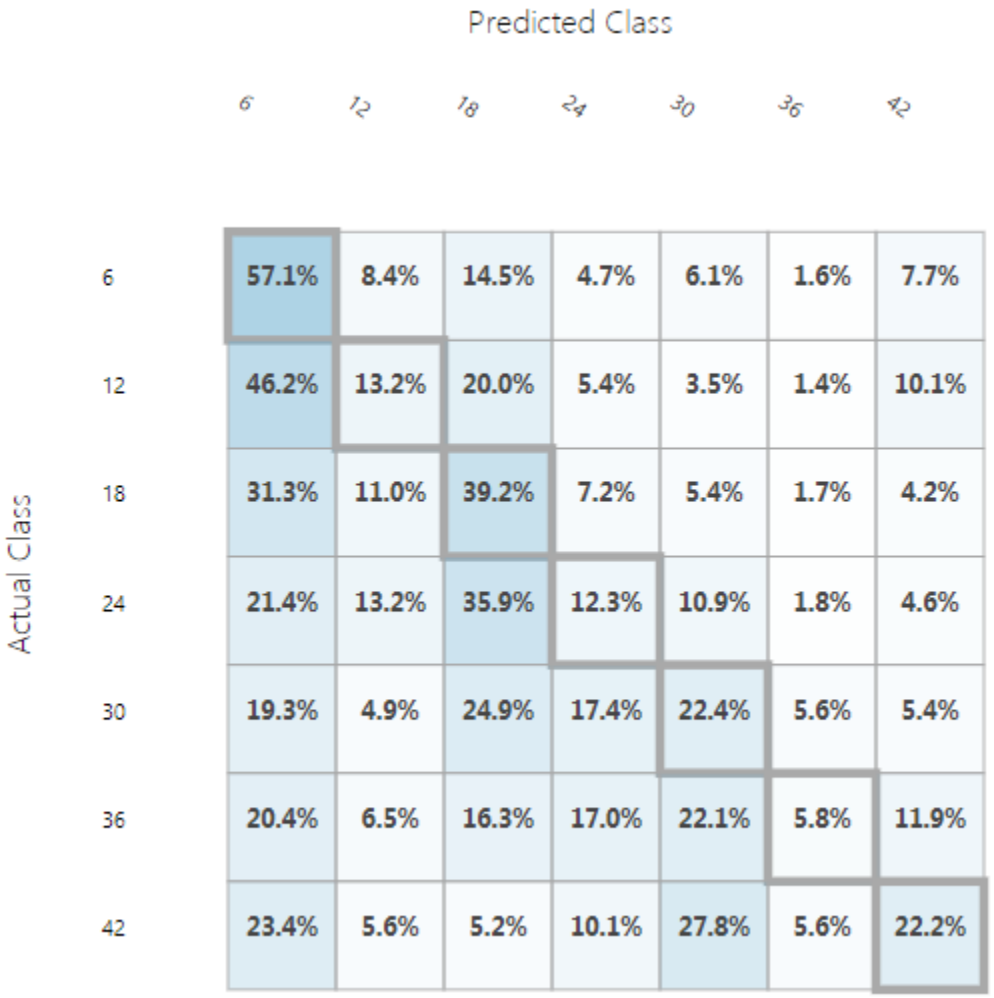
Tabell 16. Sammanställning av kvalitetsmåtten för Multiclass Neural Network

Multiclass Neural Network	**Tre-månaders intervall**	**Sex-månaders intervall**
Övergripande korrekthet	0,22	0,26
Genomsnittlig korrekthet	0,83	0,78
Makro-medelvärde precision	0,22	0,26
Mikro-medelvärde precision	0,15	0,24
Makro-medelvärde känslighet	0,22	0,26
Mikro-medelvärde känslighet	0,16	0,25

Figur 13 och 14 illustrerar korstabellerna för tre- respektive sex-månaders intervallet. Istället för att presentera antalet datapunkter för varje klassektikett som i fallet för de binära algoritmerna ges här istället en procentsats. På y-axeln av dessa tabeller så visas det faktiska värde medan x-axeln visar vad modellen har predikterat för värde. De korrekt predikterade klasserna ges av diagonalen från (3 månader, 3 månader) till (25 månader, 25 månader) där cellerna har markerats med en grå ram. En cell som saknar siffror motsvarar noll procent. Överlag är precisionen (ekvation (7) och (8)) låg för tre-månaders intervallet som kan ses i tabell 16 ovan.



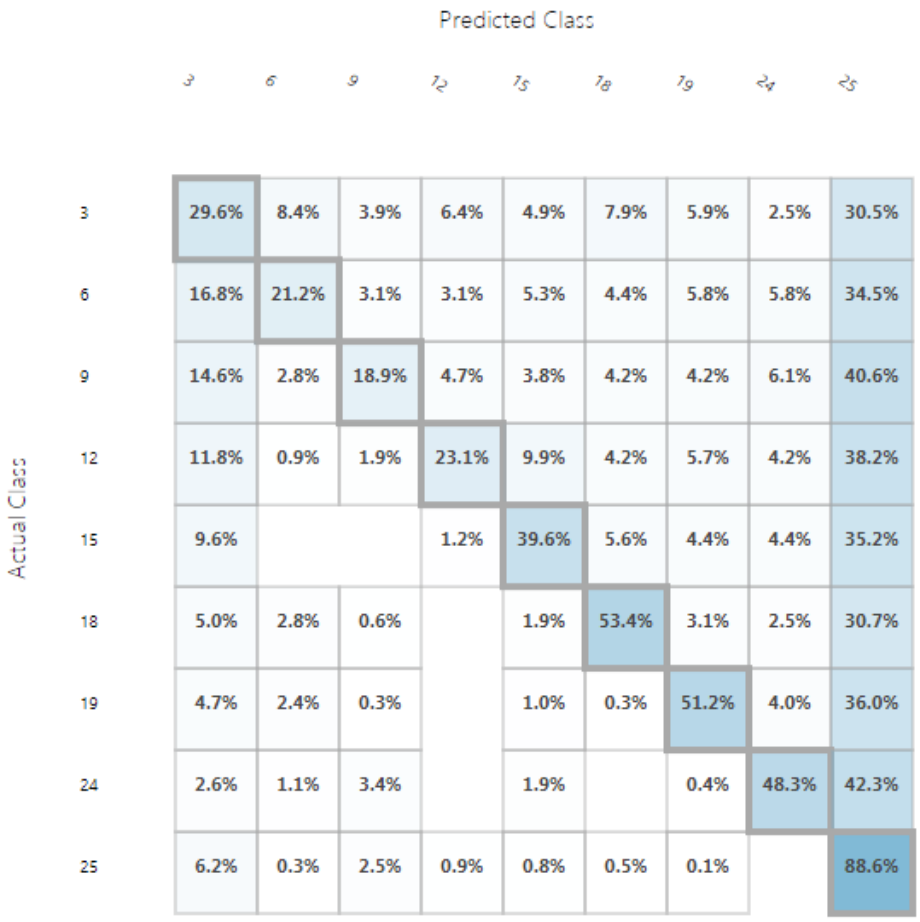
Figur 13. Tre-månaders intervallet för Multiclass Neural Network där diagonalen representerar andelen korrekta prediktioner.



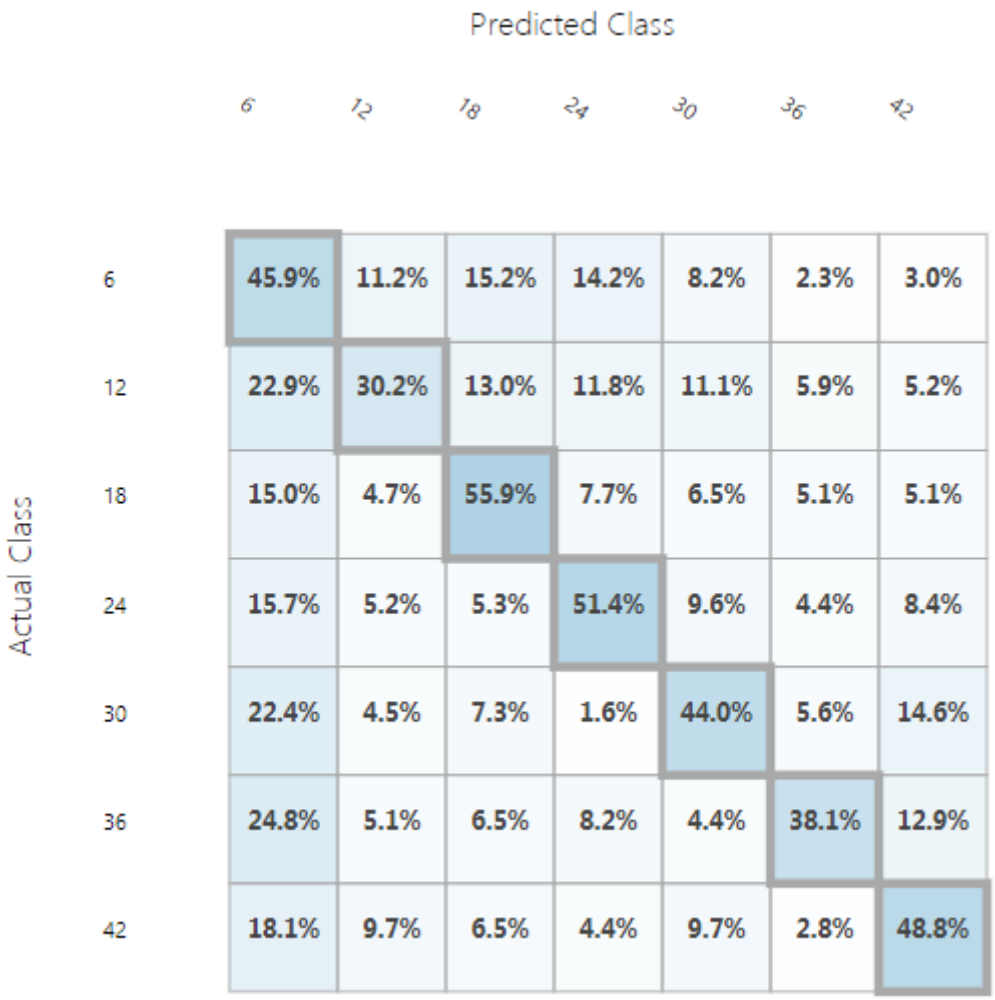
Figur 14. *Sex-månaders intervall för Multiclass Neural Network där diagonalen representerar andelen korrekta prediktioner.*

Multiclass Decision Forest

Figurerna 15 och 16 illustrerar samma sak som 13 och 14 men dessa modeller är tränade med algoritmen *Multiclass Decision Forest*. Överlag presterar denna modell bättre både för tre och sex-månaders intervall jämfört med *Multiclass Neural Network.* Modellen är tränad med sex-månaders intervallet presterar även bättre då den överlag lyckas bättre att prediktera rätt klass.



Figur 15. Tre-månaders intervall för Multiclass Decision Forest där diagonalen representerar andelen korrekta prediktioner och de tomma cellerna innebär noll procent.



Figur 16. Sex-månaders intervall för Multiclass Decision Forest där diagonalen representerar andelen korrekta prediktioner och de tomma cellerna innebär noll procent.

I tabell 17 visas en sammanställning av de utvärderingsmått som genereras från Microsoft Azure ML Studio. Överlag presterar *Multiclass Decision Forest* relativt likvärdigt för intervallen tre (modell 7) och sex månader (modell 8). Trots att det är en marginell skillnad är resultatet något bättre för tre-månaders intervallet vilket kan indikera att just detta intervall är relevant att fördjupa sig inom vid vidare forskning.

Tabell 17. Sammanställning av kvalitetsmåtten för Multiclass Decision Forest

Multiclass Decision Forest	**Tre-månaders intervall**	**Sex-månaders intervall**
Övergripande korrekthet	0,54	0,46
Genomsnittlig korrekthet	0,90	0,85
Makro-medelvärde precision	0,54	0,46
Mikro-medelvärde precision	0,54	0,46
Makro-medelvärde känslighet	0,54	0,46
Mikro-medelvärde känslighet	0,41	0,45

I tabell 18 och 19 presenteras PFI-värdena för *Multiclass Neural Network* och *Multiclass Decision Forest.* Till vänster ses de värden som representerar tre-månaders intervallet och till höger intervallet för sex månader.

Tabell 18. PFI-värden för de fem attribut med högst relevans för Multiclass Neural Network

Attribut	**PFI-värde för tre-månaders**	**Attribut**	**PFI-värde för sex-månaders**
Avtal_Forlangt_Tom	0,074	*PeriodKey*	0,156
Lokaltyp	0,074	*Sni_a_swe_name*	0,119
Hyrda_manader	0,064	*Hyrda_manader*	0,084
PeriodKey	0,049	*Year*	0,075
sni_a_swe_name	0,044	*Avtal_Forlangt_Tom*	0,058

Tabell 19. *PFI-värden för de fem attribut med högst relevans för Multiclass Decision Forest på sex månaders intervall*

Attribut	**PFI-värde för tre-månaders**	**Attribut**	**PFI-värde för sex-månaders**
Avtal_Forlangt_Tom	0,094	*PeriodKey*	0,153
Tillväxt_anställda	0,049	*Avtal_Forlangt_Tom*	0,124
Hyra_Manader	0,034	*Hyra_Manader*	0,027
PeriodKey	0,025	*Tillväxt_anställda*	0,019
Avtal_ID	0,020	*Tillväxt_vinst*	0,014

6. Diskussion

I det här avsnittet diskuteras modellernas utfall och prestanda utifrån vad som har presenterats i kapitel fem. Avslutningsvis följer en diskussion kring dess användbarhet som beslutsstöd inom fastighetsbranschen.

6.1 Binär klassificering

Modellering utfört innan balansering med SMOTE

De undersökta modellerna (modell 1-4, tabell 7) har samtliga dess för- och nackdelar gentemot de två klasserna *under* respektive *över* ett år. Inledningsvis tränades modellerna på den obalanserade datan, där kvalitetsmåtten som presenterats i tabell 10 visar kvalitetsmåtten för de två olika algoritmer som användes för modell 1 och 2 (tabell 7). Modellerna presterar väldigt jämnt med ett marginellt övertag för modell 2 som tränades med *Decision forest*. Kvalitetsmåtten för de två modellerna kan dock vara missvisande då de obalanserade modellerna i form av modell 1 och 2 tenderar att favorisera klass 2 (över ett år). Till följd av detta har modellerna i fråga lyckats få en *korrekthet* över 50% vilket betyder att de har predikterat rätt för majoriteten av testdatan, samtidigt som de även har utfört många felaktiga prediktioner där data som tillhör klassen 1 har klassificerats som klass 2.

Klass 1 (under ett år) har blivit direkt lidande av att modellerna favoriserar *klass 2* och modell 1 (tabell 7) har endast uppnått en känslighet om 51%. Ett liknande scenario kan ses för modell 2, där algoritmen *Decision Forest* har resulterat i en känslighet om 46% av för *klass 1*. Detta skulle kunna tolkas som en sorts överträning som enligt Dieterich (1995) kännetecknas av att en modell har anpassats i för stor utsträckning till dess träningsdata. I det aktuella fallet är det således möjligt att modellerna har hittat ett mönster i att majoriteten av träningsdatan tillhör klass 2 och därav antar att detta är applicerbart generellt sett.

Modellering utförd efter balansering med SMOTE

För att åtgärda problematiken kring underrepresentationen av *klass 1* genomfördes en balansering av datasetet som resulterade i en mer jämn fördelning som kan ses i figur 10. Modell 3 (tabell 7) innehållande algoritmen *Boosted Decision Tree* var den första modellen som tränades mot det balanserade datasetet. Resultatet visar på en *känslighet* av

53% för alla datapunkter i testdatan som tillhörde klass *1*. Detta resultat är en marginell förbättring då modell 1 som tränades med samma algoritm men med den obalanserade datan lyckades få en *känslighet* på 51% av testdatan som tillhör klass *1*. De övriga måtten har även endast lyckats få en liten förbättring. Resultaten pekar mot att klass 1 inte urskiljer sig tillräckligt mycket från klass *2* för att modellen ska kunna urskilja dem. Klass *2* uppvisar en förbättring då modellen har en känslighet på 80% av testdatan, vilket ska jämföras med den tidigare känsligheten (modell 1) som är 75%.

Modell 4 (tabell 7) var den andra modellen som tränades på det balanserade datasetet och använde sig utav *Decision Forest*-algoritmen. Denna modell presterar bättre för klass *1* för alla kvalitetsmått förutom precision (56%) jämfört med modell 2 som fick en precision på 60%. Detta kan bero på att modell 2 favoriserar klass *2* på samma sätt som modell 1 gör. Detta medför att modell 2 har predikterat betydligt fler datapunkter till klass *2* överlag, medan modell 4 har predikterat fler till klass *1*. Tyvärr så verkar det inte som modellen har hittat något tydligt mönster som urskiljer denna klass. *Klass 2* presterar något sämre för denna modell då den bland annat har en *känslighet* på 82%, samtidigt som *känsigheten* för det obalanserade datasetet resulterade i 87%. Teoretiskt så borde det balanserade datasetet (modell 4) prestera då den favoriserar klass *2* mindre. Problemet som har uppstått är dock att i och med att modellen inte lyckats hitta ett övertygande mönster för klass *1* så har dess *känslighet* blivit drabbad. Kvalitetsmåtten presterar överlag bättre för modell 4 än modell 2 även fast skillnaden är väldigt liten.

Enligt vad som har beskrivits i avsnittet 2.4.1 gällande obalanserad data är det dock inte tillräckligt att studera de aktuella kvalitetsmåtten när det kommer till ett ojämnt fördelat dataset (Awad & Khanna, 2015: s.52). Utifrån Awad och Khannas (2015) resonemang bör fokus istället riktats mot resultaten i korstabellerna då de övriga utvärderingsmåtten riskerar att ge en missvisande bild. Exempelvis kan en klass ha blivit favoriserad som i fallet med modell 1 och 2. Modell 3 och 4 tenderar även att favorisera klass *2* men har inte gjort detta i utsträckning som modell 1 och 2, därav är det intressant utvärdera dessa två (modell 3 och 4) ytterligare. Då det rör sig om modeller som i båda fallen tränats mot ett balanserat dataset är det återigen relevant att studera respektive utvärderingsmått istället för korstabellerna. Skillnaden är marginell, dock har modell 4 med algoritmen *Decision Forest* något bättre värden. Denna modell har presterat bättre för samtliga kvalitetsmått för båda klasserna. Även utifrån dessa resultat är det därmed inte

möjligt att dra slutsatser kring vilken av dessa två modeller som är den bästa, utan endast att de två modellerna som tränats på det balanserade datasetet presterar bättre jämfört med fallet för det obalanserade datasetet.

6.2 Multiklass-baserad klassificering

Utöver den binära klassificeringen skapades även modeller för att hantera multiklass-baserad klassificering där fler än två utfall var möjligt. Dataseten som användes liknade de som används vid tidigare klassificering med den huvudsakliga skillnaden att de innehöll flera klasser enligt vad som har beskrivits i kapitel fyra och sex. Vid en jämförelse mellan dataseten innehållande tre respektive sex-månaders intervall framkommer skillnader i fördelningen av datapunkter. Gällande det förstnämnda datasetet är fördelningen relativt jämn mellan de olika klassetiketterna bortsett från *Klass 25* som representerar samtliga kontrakt över 24 månader. Denna klass innehåller cirka 2400 datapunkter jämfört med övriga klasser som innehåller omkring 600 datapunkter. Likt vad som diskuterades i föregående del om binär klassificering skulle detta kunna peka på att även detta dataset till viss del är obalanserat enligt Awad & Khanna (2015) synsätt. Detta kan vara ett resultat av ett för litet dataset, vilket kan påverka det slutgiltiga resultatet negativt (Mukherjee et al., 2003). I fallet för datasetet innehållande sex-månaders intervallet var spridningen bättre än det tidigare datasetet men är inte helt balanserat då klass *42* är underrepresenterad. Dock innebär längre tidsintervall ett större tidsspektrum som ett kontrakt skulle kunna sägas upp inom.

Multiclass Neural Network

Inledningsvis skapades Modell 5 (tre-månaders) och 6 (sex-månaders) som innefattade algoritmen *Multiclass Neural Network)* (tabell 7). Skillnaden mellan de två modellernas kvalitetsmått (tabell 16) är marginell med ett litet övertag för modell 6. Denna modell presterar bättre för alla kvalitetsmått bortsett från den genomsnittlig korrektheten där det skiljer 5 procentenheter till modell 5:s fördel. Vidare har det nämnts att modell 5:s dataset är obalanserat på grund av *Klass 25:s* dominans och fokus bör således riktas mot korstabellerna istället för kvalitetsmåtten (Awad & Khanna, 2015: s.53). I det här fallet är det dock relevant att studera kvalitetsmåtten då det istället för att vara missledande i en positiv riktning visar på att modellen underpresterar i form av låga värden generellt samt vid jämförelse med modell 6. Skillnad är dock marginell till modell 6 fördel, vilket kan förklaras med att inte heller denna modell är fullständigt balanserad.

Utifrån vad som kan utläsas i korstabellen (figur 13) har modell 5 tydliga problem när det kommer till att prediktera en korrekt klassetikett. Klasserna *3* och *25* utmärker sig som de klasser som modellen lyckades prediktera bäst, men visar även sig att det är dessa två klasser som modellen predikterar mest frekvent oavsett vad det verkliga svaret är. Detta betyder att dessa resultat blir missvisande då den även predikterar fel flest gånger för dessa klasser. För *klass 25* överensstämmer även detta med resonemanget i kapitel 2.4.1, till följd av att denna klass är kraftigt överrepresenterad vilket kan ses i figur 11. Majoriteten av de övriga klasserna predikteras väldigt få gånger och ligger generellt omkring 10%.

När det kommer till modell 6 så presterar den bättre gällande korstabellen (figur 14) jämfört med modell 5. Denna modell är som tidigare nämnt tränad med datasetet som innehåller sex-månaders intervallet. Noterbart är hur klasserna 6 och 18 predikteras mest frekvent, vilket sammanfaller med att dessa klasser är högst representerade i träningssetet (se fördelning i figur 12). Det finns dock ingen klass som modellen lyckas prediktera rätt för majoriteten bortsett för klass *6* då dess svar kan bli missvisande då denna har predikteras med en klar majoritet oavsett om det verkliga svaret är denna klass eller inte.

Multiclass Decision Forest

Modellerna 7 (tre-månaders intervall) och 8 (sex-månaders intervall) som tränades på med algoritmen *Multiclass Decision Forest* (tabell 7) har tydligt presterat bättre än de tidigare modellerna som tränades med samma dataset. När det kommer till kvalitetsmåtten (tabell 17) har modell 7 ett övertag och presterar bättre på alla mått förutom *Mikro-medelvärde känslighet* där modell 8 uppvisar ett marginellt bättre resultat om 4 procentenheter. Gällande den *genomsnittliga korrektheten* uppvisas höga resultat för bägge modellerna, vilket avviker från de övriga kvalitetsmåtten. Detta är sannolikt ett tecken på att kvalitetsmåtten har påverkats av den obalans som existerade i dataseten. Till skillnad från modell 5, som prestera sämre än modell 6, har modell 7 som sagt presterat bättre än modell 8. Både modell 5 och 7 har tränats med samma dataset, det verkar dock som modell 7 har varit känsligare för obalansen som existerar i datasetet och som följd genererat missvisande kvalitetsmått. Överlag ses en förbättring i kvalitetsmåtten, förbättringen är dock inte tillräckligt bra för att dra slutsatsen att modellerna är fullt pålitliga.

Korstabellen (figur 15) för modell 7 visar tydligt att kvalitetsmåtten för denna modell inte är helt pålitliga, speciellt gällande den *Genomsnittliga korrektheten.* Modellen har lyckats prediktera korrekt klassetikett för majoriteten av klasserna i 20-50% av gångerna bortsett för klass *25* där 88% har uppnåtts. Dessvärre visar resultatet på ett återkommande problem där modellen majoriteten av prediktionerna väljer klass 25, något som bidrar till att ett stort antal felaktiga prediktioner. Detta visar på vad tidigare diskuterades gällande hur modellen har påverkats av den obalanserade datan då klass *25* är överrepresenterade i datasetet. Bortsett från detta har modellen genererat relativt goda resultat i termer av klass *18, 19* och *24* då modellen har predikterat rätt majoriteten av gångerna som den har predikterat att datapunkterna ska tillhöra någon av dessa klasser.

Modell 8 visar sig vara den modell som överlag uppvisar de bästa resultaten bland multiklass modellerna. Denna modell har lyckats prediktera en korrekt klassetikett i 40-50% av försöken för samtliga klasser vilket går att utläsa i dess korstabell (figur 16). Modellen har haft fel flest gånger när den har predikterat att datapunkten ska tillhöra klass *6*, detta är förståeligt då denna klass är högst representerad i träningsetet. Samtidigt har modellen problem med att prediktera en korrekt etikett i de fall där den verkliga klassen är klass 6. Detta kan tyda på att det inte finns något som urskiljer denna klass tillräckligt tydligt från de andra klasserna. Ett oväntat resultat är majoriteten av gångerna som modellen har predikterat för klass *42* så har modellen haft rätt. Anledningen varför detta är oväntat är på grund av att denna klass är en av de mest underrepresenterade i träningssetet. Vidare utmärker sig klass 18 som den klass där modellen predikterar en korrekt etikett flest antal gånger. Relevant i sammanhanget är att klass 18 tillhör de klasser som är högst representerade i träningssetet, dock verkar den inte ha favoriseras i en större utsträckning än övriga klasser. Detta gör resultaten mer pålitlig än de andra klasserna i de tidigare modellerna som har varit överrepresenterade.

Det är tydligt att algoritmen *Multiclass Decision Forest* presterar bättre än *Multiclass Neural Network.* Det går också att se att dataseten med sex-månaders intervall ger mer pålitliga svar än dataseten med tre-månaders intervall till följd av att den obalanserade datan påverkar modellen i hög grad och bidrar till missvisande kvalitetsmått. Utifrån resultaten är det värt att belysa huruvida någon av dessa modeller är tillräckligt pålitliga för att tjäna som beslutsstöd vid prediktering av hyresvakanser. Rimligtvis bör det i sådana fall vara modell 8, men modellen skulle snarare fungera som en indikator på att

en hyresgäst ligger i riskzonen för att säga upp ett kontrakt än när specifikt det kommer att ske.

6.3 Uppkomna problemområden

Missvisande finansiell information

Ett problem som har diskuterats genomgående i detta avsnitt är den obalanserade datan som huvudsakligen har försvårat träningen av modellerna. Vid sidan av detta har datan avseenden den finansiella informationen haft en relativt liten påverkan till följd av flertalet faktorer. Dels var det endast möjligt att erhålla finansiell data för en fem-års period, vilket i sin tur bidrog till att antalet datapunkter minskade drastiskt. Vidare innefattades företag som inte längre var aktiva och således exkluderades dessa företag då det saknades uppdaterad information. En ytterligare aspekt var att den finansiella datan avsåg hela företaget eller koncernen som helhet. Detta kan vara vilseledande då det inte explicit avspeglar hyresgästen i frågas ekonomiska situation. Exempelvis om ett företags ekonomiavdelning hyr en lokal och det finansiella siffrorna avser en global koncern. I ett försök för att motverka detta har dock gjorts genom att omvandla resultaten till tillväxt för att på så sätt skapa en mer rättvis bild över företagets situation från år till år.

Inte heller antalet anställda har bidragit med en önskvärd effekt. Detta beror på att siffrorna återigen avser koncernnivå och därmed blir missvisande gällande hur många personer som vistas i en lokal eller fastighet, vilket var det ursprungliga syftet med attributet. Även detta har motverkats genom att ta fram tillväxten av antalet anställda i procent för att få en bild om hur mycket företaget växer eller minskar i termer av antalet anställda. Detta har bidragit till en bättre bild över detta men är fortfarande inte helt korrekt och ger därav inte den önskade effekten.

Ett generellt problem som uppkom under arbetets gång var att den aktuella datan inte sparats korrekt. Detta visade sig genom att delar av kontrakten inte var kompletta i termer av att delar av attributen saknades. Följden blev att dessa kontrakt uteslöts från datasetet. Kombinationen av detta och problematiken med den finansiella informationen som tidigare nämnts har datasetet minskat från 170 000 datapunkter till cirka 10 000. Hade modellerna haft möjlighet att använda sig utav alla 170 000 datapunkter hade resultaten med sannolikt blivit bättre då det funnits ett bättre underlag för att identifiera potentiella

mönster. Samtidigt hade ett större dataset potentiellt kunnat öka risken för överträning och problem med att hantera nya dataset där en generaliserbar modell erfordras.

På samma tema som aspekterna som diskuterats ovan bidrog attributet *bransch* med delvis missvisande information. I likhet med den finansiella informationen hänvisar branschetiketten till koncernen som helhet. Mot den bakgrunden kan hyresgäst verksam inom ekonomi felaktigt kategoriseras som exempelvis byggverksamhet då den tillhör en större koncern inom bygg.

Permutation Feature Importances relevans

Utifrån resultaten som erhållits från PFI-modulen har det framkommit att ingen av attributen visar sig ha övervägande effekt på uppsägning av kontrakten. Däremot pekar resultaten mot att attributen i kombination med varandra ger en godkänd effekt trots att de var för sig har låga PFI-värden. En möjlig orsak till detta skulle kunna vara den missvisande datan som diskuterats tidigare. Då datan inte återspeglar en korrekt bild av den specifika hyresgästen är det möjligt att attribut som sannolikt skulle haft en stor effekt inte når upp till sin fulla potential. Exempelvis *antalet anställda,* då brist på yta för de anställda kan vara en stor anledning till uppsägningen av ett kontrakt.

Detta gäller för samtliga modeller som har presenterats i resultaten genom tabell 8, 9, 11 och 12. De attribut som näst intill alltid var bland de fem bästa attributen gällande PFI-värdet var *lokaltyp, hyrda_manader, periodkey* och *avtal_forlangt_tom*. Dessa attribut urskiljde sig marginellt men återfanns i regel bland de attribut som presterade bäst. Möjligtvis återfinns förklaringen i att dessa attribut är de som visar den mest specifika information om respektive hyresgäst. Ett ytterligare attribut som delvis återfanns bland de mest relevanta var *tillväxt_antal_anställda.* Detta styrker hypotesen gällande att det specifika antalet anställda för respektive hyresgäst skulle vara ett attribut med hög relevans. Detta då attributet är omvandlat till tillväxt i procent för att visa en mer generell bild om hur företaget växer i antalet anställda till skillnad från attributet *antal anställda* som kan vara missvisande då det gäller för hela koncernen och inte bara respektive lokal.

7. Slutsats

Utifrån det arbete som genomförts är det möjligt att dra flertalet slutsatser mot bakgrunden av studiens syfte och de frågeställningar som presenterades inledningsvis. Dels pekar detta arbete mot att det har varit möjligt att skapa prediktiva modeller för hyresvakanser men att resultaten utifrån prediktionerna inte är tillräckligt tillfredsställande. Följaktligen är tillförlitligheten hos modellerna för låg och skulle vara bristfälliga som enskilt beslutsstöd i arbetet med att förutse hyresvakanser. I dagsläget skulle modellerna kunna användas som en slags riskanalys för att se om en hyresgäst eventuellt påvisar tendenser för en uppsägning. Detta har sannolikt att göra med att datakvalitetén inte levt upp till önskad nivå och därav visas en delvis felaktig bild av hyresgästernas situation. Med andra ord skulle modellerna kunna förbättras väsentligt med mer specifik data samt ett större dataset.

Baserat på de 8 modeller som har behandlats är det modell 4 som har utmärkt sig som den som modell som presterar absolut bäst när både korstabeller och kvalitetsmått studeras. Denna modell använde sig av det balanserade datasetet och algoritmen *Decision Forest.* Återigen är det värt att nämna att tillförlitligheten inte är tillräckligt hög men att den skulle kunna tjäna som utgångspunkt för en slags riskanalys. Vidare saknades utmärkande faktorer bland de attribut som behandlats. Som tidigare nämnt, var detta sannolikt en effekt av den missvisande datan som används.

Vid vidare forskning är rekommendationen att mer specifik data används i den grad det är möjligt. Ett intressant attribut som inte var möjlig att innefatta i denna studie är så kallad *trafik i lokal.* Med detta menas att i och med att många företag har börjat implementera så kallade smarta sensorer skulle det vara möjligt att samla in hur många personer som faktiskt vistas i en lokal eller fastighet. Detta hade varit ett tydligare sätt för att studera om ett företag är påväg att växa ur en lokal eller inte.

Referenser

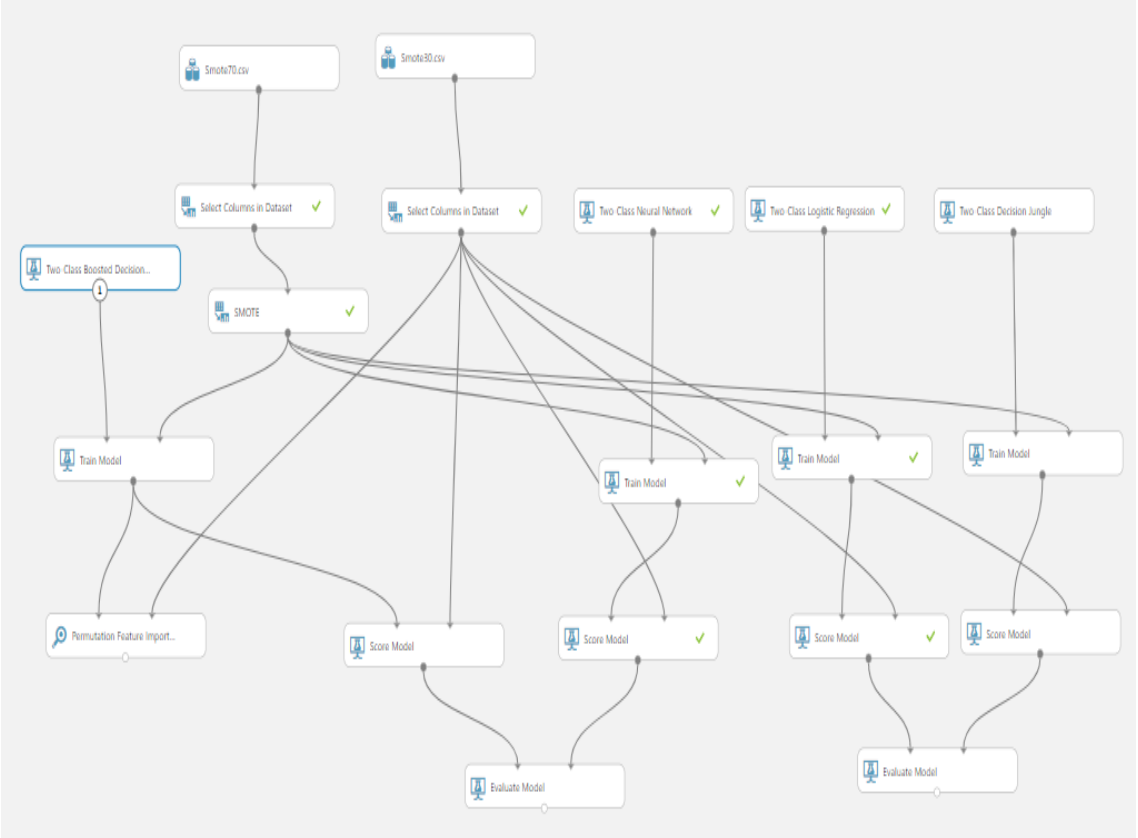
- Awad, M., Khanna, R. (2015), *Efficient Learning Machines - Theories, Concepts, and Applications for Engineers and System Designer*s. Apress, Berkeley, CA. s.54
- Azure Machine Learning Studio. (2018) https://docs.Microsoft.com/en-us/azure/machine-learning/studio-module-reference/ (Hämtad 2018-03-18)
- Bell, J 2014, *Machine Learning: Hands-On for Developers and Technical Professionals*, Wiley, Somerset. Available from: ProQuest Ebook Central. [29 January 2018].
- Bejrum, H & Lundström, S. (1989). *Mest sannolik hyra för lokaler - principer och metoder*. Sveriges fastighetsägarförbund. Stockholm
- Bleik, S. (2015). *Permutation Feature Importance. I: Cortana Intelligence and Machine Learning.* url: https://blogs.technet.Microsoft.com/ machinelearning/2015/04/14/permutation-feature-importance/ (hämtad 2018-03-15).
- Blockeel, H., (2011), *Hypothesis space*. Encyclopedia of Machine Learning, vol. 1, s.511-513
- Boyd Enders F. (2018) *Coefficient of determination*. https://www.britannica.com/science/coefficient-of-determination (Hämtad 2018-02-28)
- Charles, D., Fyfe, C., Livingstone, D., McGlinchey, S., 2008. *Biologically inspired artificial intelligence for computer games*. Medical Information Science Reference, Hershey, PA.
- Coadou, Y. (2013), *Boosted Decision Trees and Applications*. EDP Science. s.1-11
- Dietterich, T. (1995), *Overfitting and undercomputing in machine learning*. ACM Computing Surveys (CSUR), vol. 27, nr. 3, s.326-327.
- Domingos, P., (2012), *A few useful things to know about machine learning*. Communications of the ACM, vol. 55, nr. 10, s.78–87.
- Engelbrecht, A.P. (2007). *Computational Intelligence: An Introduction*. I: John Wiley & Sons, s.1–96.

- Grenadier, S. R. (1993). *Local and National Determinants of Office Vacancies*. Journal of urban economics, s.57-71
- *Gepsoft*. https://www.gepsoft.com/gxpt4kb/Chapter10/Section1/SS06.htm (Hämtad 2018-02-27)
- Hastie, T., Tibshirani, R., Friedman, J.H. (2009), *The elements of statistical learning: data mining, inference, and prediction*, andra upplagan, Springer, New York, NY.
- Hills J. (2016) *The Effects of Machine Learning in Real Estate* https://www.rexsoftware.com/effects-machine-learning-real-estate/ (Hämtad 2018-03-12)
- Illinois state University. *Absolute and Relative Error* http://www2.phy.ilstu.edu/~wenning/slh/Absolute%20Relative%20Error.pdf (Hämtad 2018-02-27)
- James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. Springer.
- Kohonene T, Deboeck G, (1998) *Visual Explorations in Finance with Self-Organizing Maps*, (Springer, London, 1998), s. 258
- Lawsona, E. , Smithb., Sofgea, D., Elmoreb, P., Petry, F. (2017). *Decision forests for machine learning classification of large, noisy seafloor feature sets*. Computers & Geosciences 99. s.116–124
- Learned-Miller, E. (2014). ”*Introduction to Supervised Learning*”. I: Department of Computer Science, University of Massachusetts.
- Lind, H. & Lundström, S. (2009). *Kommersiella fastigheter i samhällsbyggandet, Stockholm*: SNS Förlag
- Louridas P & Ebert C, *Machine Learning*, in *IEEE Software*, vol. 33, no. 5, s. 110-115, Sept.-Oct. 2016.
- Lundström, S. (2008). *Fastighetsföretagande. I Fastighetsekonomisk analys och fastighetsrätt: fastighetsnomenklatur*. (10. uppl.). Stockholm: Fastighetsnytt. s.449 – 466.
- Maxwell, S., Kelley, K., Rausch, J.R. (2008). *Sample size planning for statistical power and accuracy in parameter estimation*. I: Annual review of psychology 59, s.537–563.
- Mitchell, Tom M. 1951- (Tom Michael) 1997, *Machine learning*, McGraw-Hill, New York.

- Mohri, M, Rostamizadeh, A, & Talwalkar, A 2014, *Foundations of Machine Learning, MIT Press, Cambridge*. Available from: ProQuest Ebook Central. [29 January 2018].
- Mukherjee, S., Tamayo, P., Rogerso, S., Rifkin, R., Engle, A., Campbello, Golub, C., Mesirov, J.P. (2003). ” Estimating Dataset Size Requirements for Classifying DNA Microarray Data”. I: Journal of Computational Biology 10, s. 119–142
- Nationalencyklopedin, *Artificiell intelligens*. http://www.ne.se/uppslagsverk/encyklopedi/lång/artificiell-intelligens (hämtad 2018-02-02)
- Priddy K,. Keller P. (2005). *Artificial Neural Networks: An Introduction.* DOI: 10.1117/3.633187
- Prinzie, A., & Van den Poel, D. (2008). *Random forests for multiclass classification: Random MultiNomial logit* Elsevier. doi:10.1016/j.eswa.2007.01.029
- Remøy, H. & van der Voordt, T. (2007). *A new life: conversion of vacant office buildings into housing*. Facilities, 25 (3), s.88-103.
- Raschka, S. (2015), *What is hypothesis in machine learning?* Quora. Tillgänglig online: https://www.quora.com/What-is-hypothesis-in-machine-learning (2017-0516).
- Rossini P. (2000) *Using Expert Systems and Artificial Intelligence For Real Estate Forecasting.* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.930&rep=rep1&type=pdf (Hämtad 2018-04-10)
- Russell, S., Norvig, P. (1995), *Artificial Intelligence: A modern approach.* PrenticeHall, Englewood Cliffs, 25, s.27.
- Shawkat, A., Smith, K. (2004). *On learning algorithm selection for classification.* Science Direct. s.119-138
- Shotton, J., Nowozin, S., Sharp, T., John, J., Kohli, P., Criminisi, A. (2013). *Advances in Neural Information Processing Systems. Decision jungles*: Compact and rich models for classification.
- Shanmuganathan S. (2016) *Artificial Neural Network Modelling: An Introduction*. Studies in Computational Intelligence, vol 628. Springer, Cham

- Holmes S. (2000) RMS Error http://statweb.stanford.edu/~susan/courses/s60/split/node60.html (Hämtad 2018-02-26)
- Statisticshowto. .(2016) *Absolute Error & Mean Absolute Error* http://www.statisticshowto.com/absolute-error/ (Hämtad 2018-02-27)
- Sun B, Luo J, Shu S & Yu N, "*A new approach of Random Forest for multiclass classification problem*," *2010* 5th International Conference on Computer Science & Education, Hefei, 2010, s.6-8.
- Suthaharan, S. (2016). *Machine Learning Models and Algorithms for Big Data Classification*. 36:e Upplagan, Springer, New York.
- *Supervised learning*. https://www.mathworks.com/discovery/supervised-learning.html (Hämtad 2018-02-06)
- Taylor A. (2017) *6 Ways Artificial Intellience is Transforming Real Estate Investing* https://becominghuman.ai/6-ways-artificial-intelligence-is-transforming-real-estate-investing-6917e82c1521 (Hämtad 2018-02-27)
- Tan, P., Steinbach, M., Kumar, V. (2005). *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co
- Ting, K.M. (2010). *Encyclopedia of Machine Learning.* Confusion matrix. Vol. 18. Boston: Springer US, s.209.
- Torgo, L. (2000). *Inductive learning of tree-based regression models.* Ai Communications.
- Turing, A.M (1950) *Computing Machinery and Intelligence*. Mind: A Quarterly Review of Psychology and Philosophy, October 1950: 433-460
- Yang, Y. (1997). "*An evaluation of statistical approaches to text categorization*". I: Journal of Information Retrieval 1.
- Zhang, S., Yang, C., Yang, Q. (2003). ” *Data Preparation for Data Mining*”. I: Applied Artificial Intelligence 17.5, s.375–38

Bilaga A


Smote70.csv
Smote80.csv
Select Columns in Dataset
Select Columns in Dataset
Two-Class Neural Network
Two-Class Logistic Regression
Two-Class Decision Jungle
Two-Class Boosted Decision...
SMOTE
Train Model
Train Model
Train Model
Train Model
Permutation Feature Import...
Score Model
Score Model
Score Model
Score Model
Evaluate Model
Evaluate Model

Bilaga B

Arbetsbelastningen under examensarbetet har varit jämnt fördelat mellan de båda författarna. Detta gäller såväl rapportskrivande som övrigt arbete i form av bland annat databehandling i Microsoft Excel. Tack vare att rapporten huvudsakligen har skrivits i Google Docs har det varit möjligt att arbeta parallellt och korrekturläsa den andre författarens text kontinuerligt. Kapitelfördelningen som presenteras nedan visar vem som har haft det huvudsakliga ansvaret för respektive avsnitt. Båda författarna har dock varit delaktiga i samtliga delar genom diskussioner och löpande korrekturarbete.

Abstract Brook Alemayehu

Inledning Fredrik Johnson

Bakgrund Brook Alemayehu & Fredrik Johnson

Data Fredrik Johnson

Metod Brook Alemayehu

Resultat Brook Alemayehu & Fredrik Johnson

Diskussion & slutsats Brook Alemayehu