Definition

Paylevos verksamhet utgörs av online fakturering, där tanken är att överföra den risk som sker när köpet och betalning inte sker på samma gång. Således så kommer Paylevo att ”köpa” fakturan och överföra likvida medel till företagaren. Nu så är verksamheten mera komplex, risken är ofta på handlarens sida, men för att lösa denna uppgift så kommer vi att låtsas att Paylevo tar risken och på så vis äger varje faktura som skapas i systemet.

Alla fakturor är omvandla till svenska kronor. Följande villkor och statistik gäller initialt:

1) *Tidsperiod*: Start (2014-10-01) till (2016-12-24), vilket gör att en faktura idag är ansedd som en kredit förlust, cirka 4 månader.

2) *Utgivna fakturor*: Avser även fakturor som slut kunden ångrat sitt köp, totala antalet är 243 644 st.

3) *Antal unika kunder*: 74 516 st.

4) *Inflöde av pengar*: 100 031 141 SEK, är beräknad genom inflöde av likvida medel minus utflödet av pengar. Sker på grund av att vi ibland måste returnera pengar till slutkunden.

5) *Charges*: Är summan av alla intäkter från inkasso, ränta betalningspåminnelse, transaktionsavgift och fakturaavgift, vilket blir 17 112 513 SEK.

6) *Förväntad förlust*: Är alla fakturor som idag är ansedda som en kredit förlust och avser endast själva kapital beloppet, 15 139 692 SEK.

7) *Lönsam kund*: Detta sker på kundnivå, och genom förenkling så bortses här från den kostnad som är förknippad när Paylevo utfärdar en faktura. Men principen för en lönsam kund är *Charges* − *Förväntad förlust*.

Efter att rensat för de kunder där vi har kredit data samt endast för svenska så ser bilden

ut enligt följande:

1) *Utgivna fakturor*: 145 885 st.

2) *Antal unika kunder*: 52 223 st.

3) *Inflöde av pengar*: 68 719 278 SEK.

4) *Charges*: 13 769 029 SEK.

5) *Förväntad förlust*: 12 339 323 SEK.

6) *Procentuella förlust*: Baserad på antal fakturor, 18.8 procent, och total summa 18 procent. Dessa siffror är väldigt höga, och stämmer inte. Men för att avgöra de bra från de dåliga så låtsas Paylevo ta risken för alla fakturor.

$$\text{Customer profitably} = \begin{cases} 1 & \text{Yes} \\ 0 & \text{No} \end{cases}$$

Data städning

Första steget i analysen är att ta bort data som ”stör” modelleringen. Först ut är att ta bort alla variablar som innehar ”NA” inom sig. Vårt dataset är relativt stort och behöver således inte betyda så mycket om ett par hundra försvinner. I detta fall så var det ca 90 st observationer som försvann. Även de variabler som innehar en stor andel NA tas bort, tex *DEBTRELIEF_DATE*. Ytterligare arbete är att sammanställa vissa variablar för att minska variationen.

Minska variation på följande

DEBT_NUMBER =IF $x < 5$ ”Låg” ELSE ”Hög”

AGE =IF$x < 30$ ”Young” ELSE IF $x >= 30 \& x < 50$ ”Middle” ELSE Old

Samtidigt så måste även variablar som inte har någon som helst variation inom sig också tas bort[1], tex *INCOME_BUSINESS* och 3 liknande variablar försvinner. Outliers drabbar olika för olika modeller, tex beslutträd är inte känslig för outliers medans regressons modeller är det. Slutligen så avslutas ”städningen” genom att dela i analys data i två delar, ena är där modellerna skall tränas och andra är test datasetet där prognosen skall utvärderas på. Skulle vi inte göra på detta sätt så skulle det uppstå en bias eftersom vi utför en prognos på samma dataset som modelleringen utformats inom i.

Först ut är att granska den interna korrelationen för de numeriska variablerna. Ett stort problem i all modellering arbete är att jobba med väldigt högt korrelerat data i de explanativa variabler[2]. Orsaken till detta är den interna korrelationen är alldeles för hög och på så viss svårigheter att avgöra vad som korrelerar med y - variabeln.

Följande analys i figur 2 sker endast för de numeriska variablerna. Det man kan se är att det finns en väldigt stark korrelation sinsemellan alla x-variabler (översta grafen). Tex *INCOME* med *INCOME2*, dvs har man haft en hög inkomst föregående år så har man även

[1] Använder funktionen `nearZeroVar()` som tittar på antalet unika värden i relation till stickprovet.

[2] Dvs de variablar som ska användas för att utföra prognosen.

det kommande år. Således kommer dessa att tas bort. De som tas bort efter att använt en ”cutoff” på 90% procent är tex *TEXEBLE_INCOME* och *FINAL_TAX* samt ytterligare två till.

Beslutsträd

Inför arbetet med att hitta den optimala modellen så kommer först att utröna vilka variabler som är viktiga för att utföra en prognos att genomföras. Just nu är det variable enligt ekvation (1) som utforskas.

Vad som sker enligt nedanstående kodning är att tränings datasetet splittas i ytterligare 10 delar och där ett beslutträd skapas i varje, detta utförs 3 gånger. Vi använder ROC för att att hitta det bästa trädet.

Listing 1: Metod

```
fitControl <- trainControl(## 10-fold CV
   method = "repeatedcv",
      number = 10,
      summaryFunction = twoClassSummary,
        classProbs = TRUE,
      ## repeated 3 times
           epeats = 3)


bmFit1 <- train(y = yVar$Profitably,
                                  x =  xVar,
                           method = "rpart",
                           metric = 'ROC',
                         trControl = fitControl,
                         tuneLength = 20)
```

Så baserat på fig (3) så kommer ytterligare variabler att reducerats inför den första

testomgången och en baseline modell. I princip så väljs de variablar som modellen anser som viktigast, skär vid 20. Den variabel som är allra viktigast i beräkna vilken kund som är ansedd som vinstgivande är alltså *DEBT_SUM* följt av SCORING.

Jag kommer att använda Gini måttet i och reducerat trädet med hjälp av komplexitets parametern. Figur (4) i sida 12 fås då. Valet av komplexitet parametern fås genom ”1SE” regeln, vilket kollar på kolumn ”xerror” och ”xstd” [3]

Modellen är en aningen för komplex för att erhålla alla regler. Men kollar vi de regler som fångar upp den stora massan ser det ut enligt följande.

(3.1)
$$\begin{aligned} &\text{DEBT\_SUM} >= 226 \ \& \ \text{DEBT\_NUMBER} == \text{'High'} \ \ =\text{Regel 1} \\ &[\text{DEBT\_SUM} < 226 \ \ =\text{Regel 2} \\ &\text{DEBT\_SUM} >= 226 \ \& \ \text{DEBT\_NUMBER} != \text{'High'} \ \& \ \text{TOTAL\_INCOME} < 88000 \ \ =\text{Regel 3} \end{aligned}$$

Tabell 1: Summering av modell enligt fig (4)

Logisk regel	Kreditförlust (utgift)	Vinst (Charges)	Kreditförlust (%)
1)	-6 882 924	335 216	84.96
2)	-2 357 855	12 173 447	5.15
3)	-809 985	50 608	82.49

Så vad säger tabell 1, en positiv kreditförlust innebär att man sparat pengar genom att tillämpa respektive regel, och en negativ kreditförlust att det är pengarna som företaget sparat.

Skulle vi exempelvis tillämpa regel 1, så skulle vi bespara företaget en kreditförlust på -6 882 924 SEK (eftersom vi då köpt fakturor för den summa), men eftersom vi tillämpar regeln så kommer vi att förlora en förväntad intäkt[4], 335 216 SEK. På samma viss är det för regel två, där vi köper fakturor från folk som har en mindre skuld än 226 SEK. På så viss har vi en kreditförlust på -2 357 855 men en intäkt på 12 173 447 SEK.

Så för att göra en prognos baserad på nedanstående modell måste vi först hitta den optimala cutoffen. Detta görs bäst med hjälp av ROC analysen. Först skapas en vektor av sannolikheten baserad på modellen och test data.

[3] Cross validation error och dess standard fel

[4] Alla är ju inte dåliga kunder, ca 15.04 äv kunderna är ansedda som bra.

Tabell 2: Prognos av modell enligt fig (4) på test data antal 8152

Prognos/Utfall	False	True
False	947	234
True	374	6597
Accuracy	0.925	–
Sensitivity	0.966	–
Specificity	0.717	–
Precision (PPV)	0.946	–
Neg predictive value (NPV)	0.802	–
AUC	0.872	–

$$\text{Accuarcy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \tag{3.2}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3.3}$$

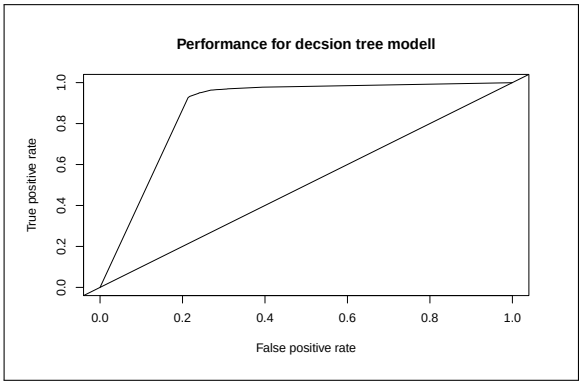
$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.4}$$

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.5}$$

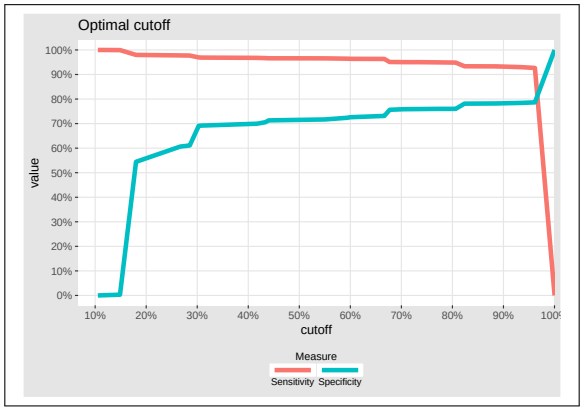
$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{3.6}$$

Figur 1: ROC

(a)



(b)



Bagging & boosting

The best way to evaluate and train models is to use Bagging or boosting. More specific bagging create lots of trees.

1. Bagging then involves averages of many trees and produces smoother decision boundaries.

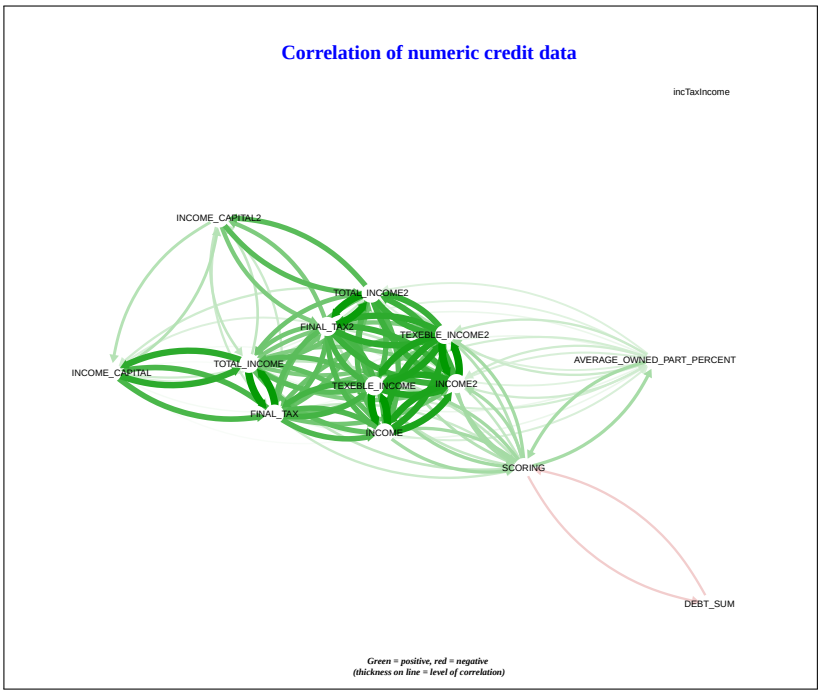
2. It reduces error by variance reductions

3. Bias is slightly increased because bagged trees ar slightly shallower

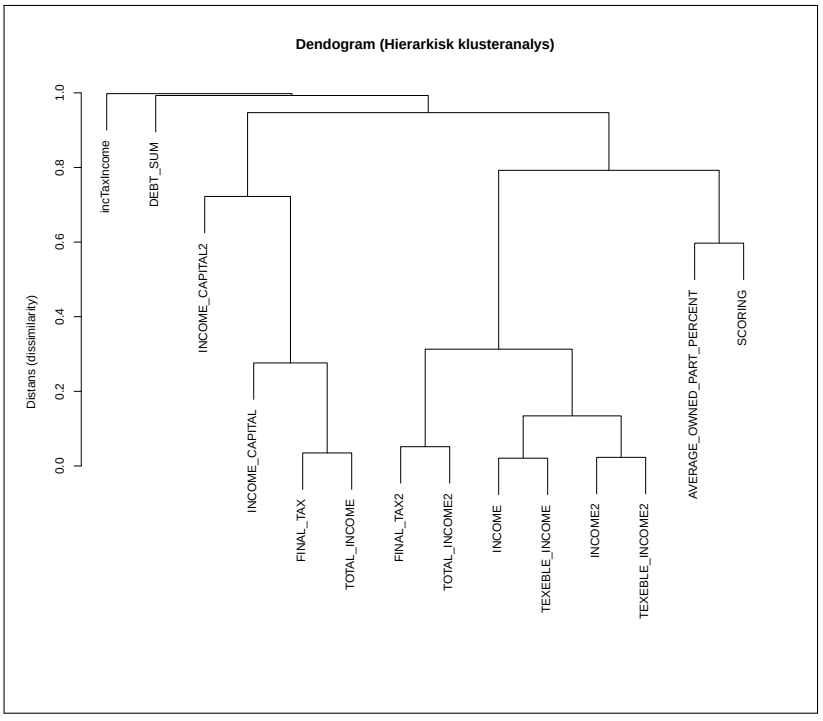
Korrelations grafer

Figur 2: Korrelations analys

(a)



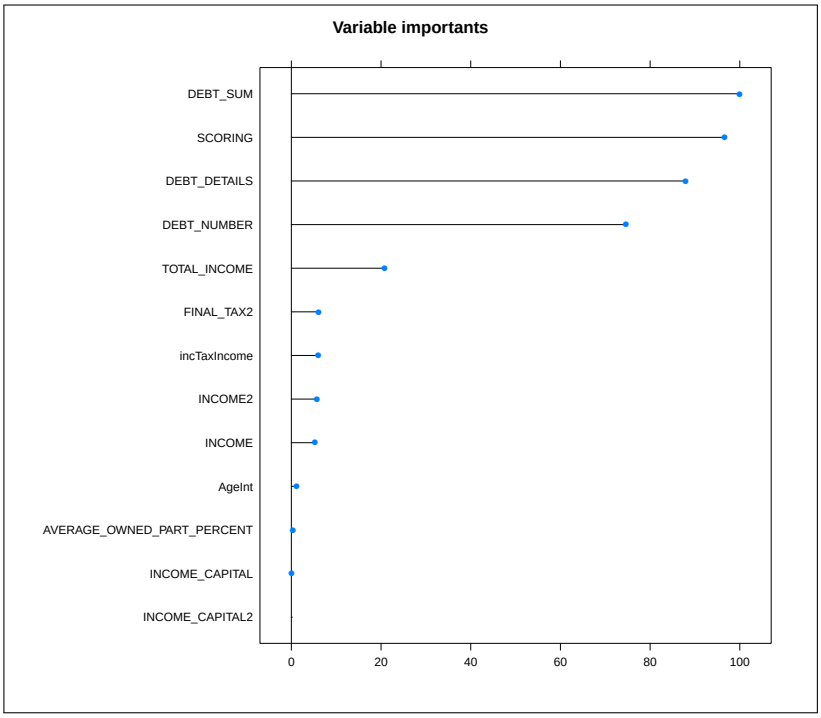
(b)



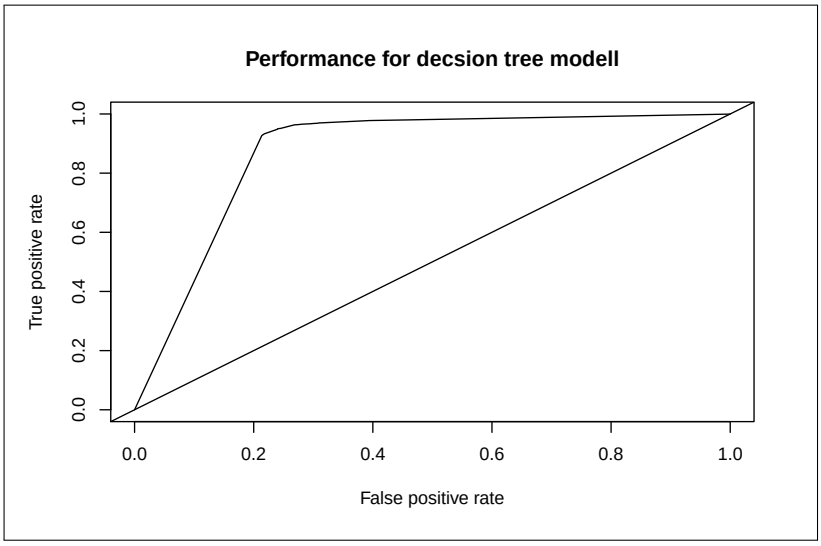
Beslutsträd

Figur 3: Caret och optimering

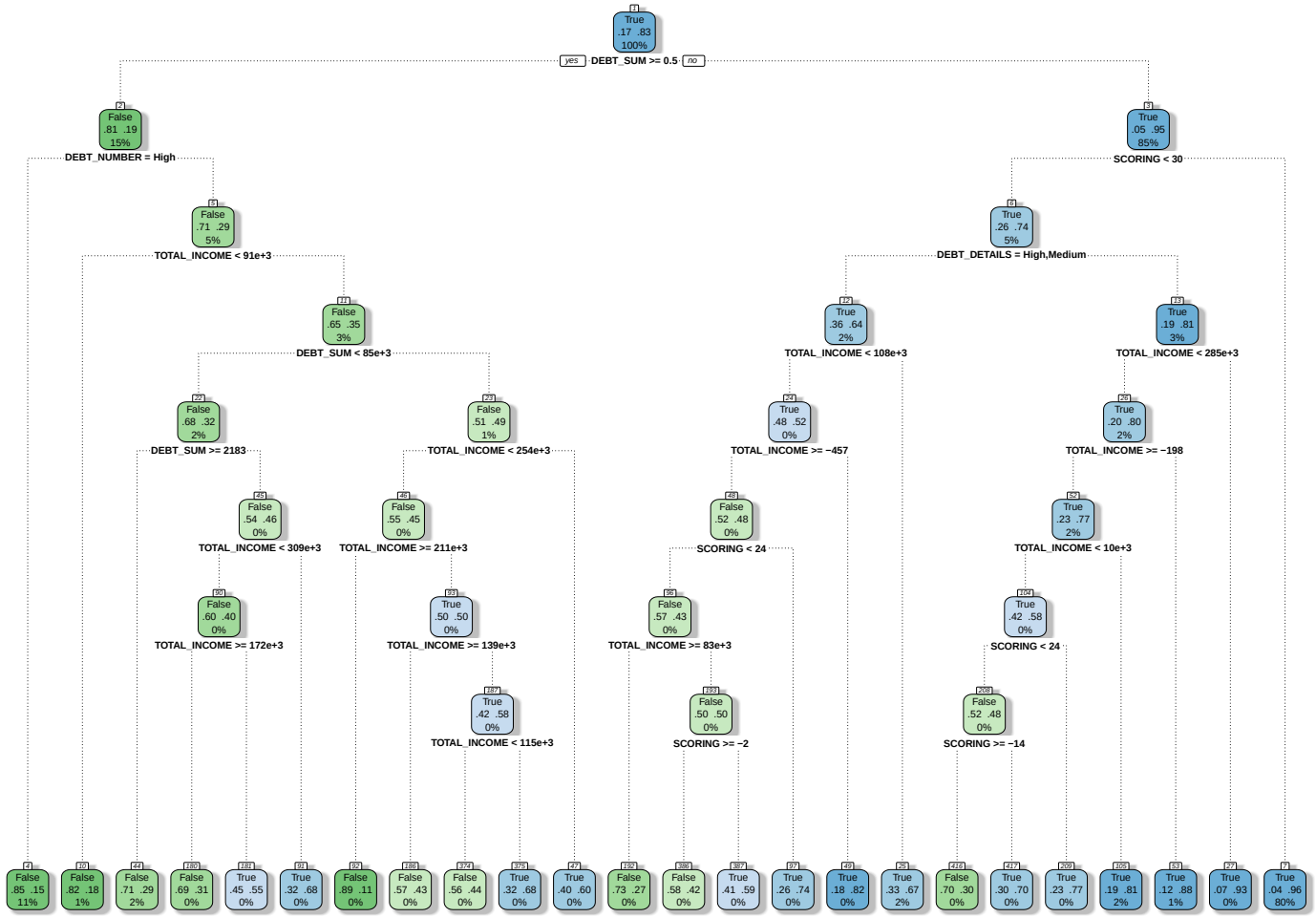
(a)



(b)



Figur 4: Beslutsträd, reducerat genom komplexitet parametern



Baserat på SE regel, 0.000369276218611521