met4_heks_v21 løsningsforslag

Q1 - Deskriptiv statistikk

Laster inn pakker og datasettet:

```
library(dplyr)
library(ggplot2)
library(stargazer)

load("met4_v21.RData")
```

Vi kan først ta på et overordnet blikk på treningsdatasettet. De første radene i datasettet:

```
head(df_train)
```

```
## # A tibble: 6 x 12
##   CHURN_30 CTL_TENURE_DAYS CUSATTR_GENDER N_OBJ_TOT NET_PREM_TOT TAR_PREM_TOT N_OBJ_AU_400 N_OBJ_FB N_OBJ_PK_HUS
N_OBJ_PK_INNBO N_OBJ_PK_FRHUS   AGE
##      <dbl>           <dbl>          <dbl>     <dbl>        <dbl>        <dbl>        <dbl>    <dbl>        <dbl>
<dbl>          <dbl> <dbl>
## 1        0            1239              1         1        1619.        1248.            0        0            0
0              0    35
## 2        0             147              0         2        3175.        3122.            0        0            0
1              0    23
## 3        0            2265              0         5        8925.       10583.            0        0            2
0              0    62
## 4        0             779              0         2        2583.        2337.            0        0            0
0              0    35
## 5        0            2196              0         8       17250.       22748.            1        0            1
1              0    38
## 6        0            3482              1         4       16345.       20625.            1        0            1
0              0    56
```

Vi kan også lage en enkel oversikt over deskriptiv statistikk for variablene ved hjelp av funksjonen `skim()` i pakken `skimr`.

```
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 4.0.5
```

```
skim(df_train)
```

Data summary

Name	df_train
Number of rows	25138
Number of columns	12
_____	
Column type frequency:	
numeric	12
_____	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
CHURN_30	0	1	0.01	0.10	0.00	0.0	0.00	0.00	1.0	▇▁▁▁▁

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
CTL_TENURE_DAYS	0	1	1776.56	1363.50	1.00	668.0	1442.50	2607.00	5844.0	
CUSATTR_GENDER	0	1	0.38	0.49	0.00	0.0	0.00	1.00	1.0	
N_OBJ_TOT	0	1	4.37	3.08	1.00	2.0	4.00	6.00	31.0	
NET_PREM_TOT	0	1	10430.76	8241.56	217.73	3262.9	9071.31	15236.42	120581.8	
TAR_PREM_TOT	0	1	13633.41	11978.49	-152.05	3332.6	11147.91	20414.04	118205.8	
N_OBJ_AU_400	0	1	0.74	0.80	0.00	0.0	1.00	1.00	6.0	
N_OBJ_FB	0	1	0.09	0.34	0.00	0.0	0.00	0.00	6.0	
N_OBJ_PK_HUS	0	1	0.74	1.02	0.00	0.0	0.00	1.00	13.0	
N_OBJ_PK_INNBO	0	1	0.66	0.60	0.00	0.0	1.00	1.00	6.0	
N_OBJ_PK_FRHUS	0	1	0.20	0.68	0.00	0.0	0.00	0.00	12.0	
AGE	0	1	46.72	14.63	18.00	36.0	44.00	56.00	100.0	

Et par vesentlige momenter å merke seg:

- Vi har *ingen* manglende observasjoner i datasettet. Det er betryggende for bruken av modellen.
- Alle variablene er numeriske, men de er likevel ulike:
  - `CUSATTR_GENDER*` er binær
  - `N_OBJ_*` -variablene er positive heltall, men med relativt lave verdier.
  - `AGE` er også heltall
  - `*_PREM_TOT` variablene er numeriske
  - `CTL_TENURE_DAYS` er positive heltall, men mange ulike verdier.
- `TAR_PREM_TOT` er litt underlig, ettersom den har enkelte *negative* verdier. Vi har ikke informasjon i oppgaven som kan forklare dette, men man skulle vel anta at tariffpremien nok burde vært positiv for alle kunde.
- Fordelingen til begge `*_PREM_TOT` -variablene er gode eksempler på variabler med en ekstrem fordeling. Fra histogrammene ser vi at det er typisk "normale" verdier på variablene, men enkelte svært høye verdier.

Når vi senere skal vurdere ulike modeller er derfor følgende viktige momenter å ta høyde for:

- For premievariablene vil nok ekstremobservasjoner kunne påvirke modellresultater mye dersom vi kun lar premiene inngå lineært i modellen. Mulige løsninger kan være å bruke en mer fleksibel modell, log-transformasjoner, eller å bruke dummy-variabler for å fange opp ekstreme observasjoner.
- For `N_OBJ_*` -variablene har vi i mindre grad en ekstremobservasjonproblematikk, men vi kan vurdere om f.eks. om antall hus forsikret bør inngå som et lineært ledd, eller om denne bør kodes om til dummy-variabler.

Q2 - Feature Engineering

Det er mye man kan vurdere med tanke på transformasjoner, men den grunnleggende utfordringen er vel at selv med et så lite datasett som vi har her er det *svært mange* mulige måter å representere variablene på.

Et annet poeng er at enkelte variabler - eksempelvis premier - er svært skjevfordelt. Dersom man bruker disse direkte i en modell vil effekten av ekstremobservasjoner kunne påvirke resultatet i stor grad. Alternativer er da f.eks. å la modellen være mer fleksibel i disse variablene - eksempelvis med polynomer - eller å transformere variablene før de brukes i modellen (eksempelvis en log-transformasjon).

Korrelasjonsmatrisen for variablene i datasettet er gitt ved:

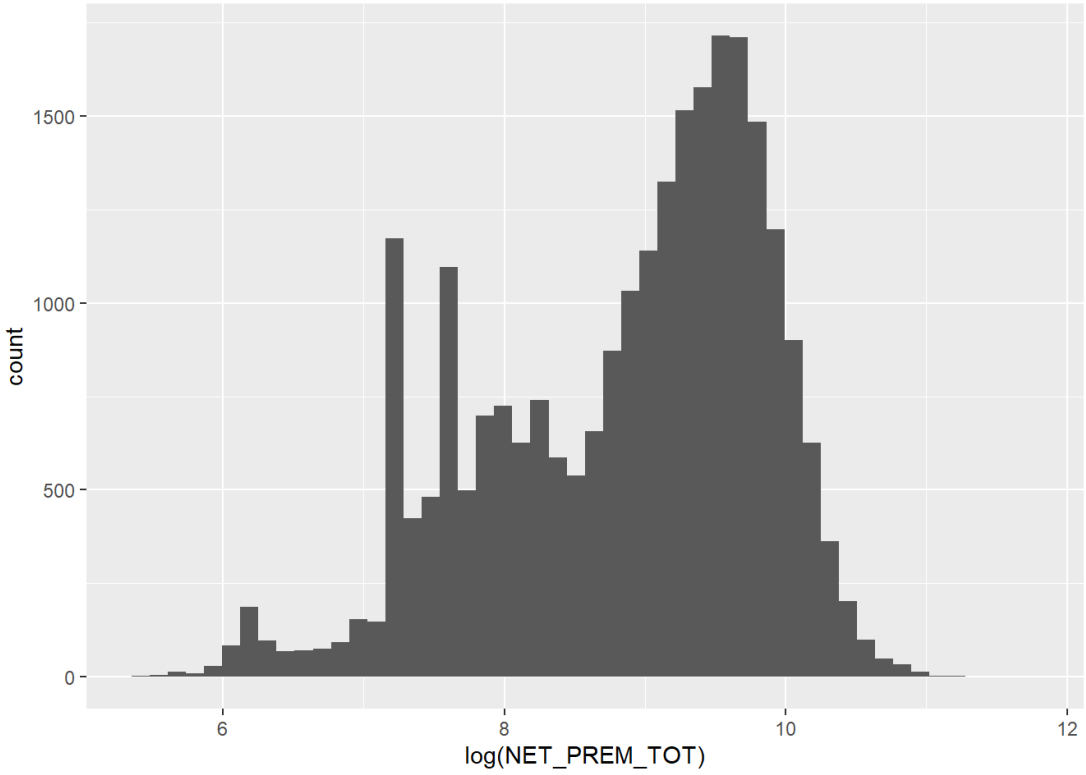
```
round(cor(df_train), digits = 2)
```

```
##                CHURN_30 CTL_TENURE_DAYS CUSATTR_GENDER N_OBJ_TOT NET_PREM_TOT TAR_PREM_TOT N_OBJ_AU_400 N_OBJ_F
B N_OBJ_PK_HUS N_OBJ_PK_INNBO N_OBJ_PK_FRHUS   AGE
## CHURN_30           1.00          -0.04         -0.01      0.03         0.05         0.04         0.04     0.0
2         0.02          0.00          0.01 -0.02
## CTL_TENURE_DAYS    -0.04           1.00          0.01      0.09         0.08         0.07         0.04     0.0
5        -0.01         -0.08          0.05  0.28
## CUSATTR_GENDER     -0.01           0.01          1.00     -0.13        -0.13        -0.14        -0.13    -0.1
0        -0.12          0.03         -0.05  0.06
## N_OBJ_TOT           0.03           0.09         -0.13      1.00         0.82         0.84         0.66     0.3
9         0.70          0.20          0.43  0.15
## NET_PREM_TOT        0.05           0.08         -0.13      0.82         1.00         0.98         0.74     0.3
3         0.59          0.11          0.30  0.13
## TAR_PREM_TOT        0.04           0.07         -0.14      0.84         0.98         1.00         0.78     0.3
5         0.58          0.10          0.30  0.12
## N_OBJ_AU_400        0.04           0.04         -0.13      0.66         0.74         0.78         1.00     0.2
0         0.40          0.02          0.17  0.12
## N_OBJ_FB            0.02           0.05         -0.10      0.39         0.33         0.35         0.20     1.0
0         0.19          0.03          0.28  0.13
## N_OBJ_PK_HUS        0.02          -0.01         -0.12      0.70         0.59         0.58         0.40     0.1
9         1.00         -0.01          0.12  0.12
## N_OBJ_PK_INNBO      0.00          -0.08          0.03      0.20         0.11         0.10         0.02     0.0
3        -0.01          1.00          0.04  0.01
## N_OBJ_PK_FRHUS      0.01           0.05         -0.05      0.43         0.30         0.30         0.17     0.2
8         0.12          0.04          1.00  0.24
## AGE                -0.02           0.28          0.06      0.15         0.13         0.12         0.12     0.1
3         0.12          0.01          0.24  1.00
```

Vi legger merkte til at det er en sterk positiv korrelasjon mellom enkelte variabler, og særlig nettopremier og tariffpremier. Ettersom det nå bygger seg opp en del argumenter for at tariffpremier er en problematisk variabel (negative verdier og høy korrelasjon med nettopremier), vil vi i dette løsnngsforslaget droppe denne variabelen. Man kunne eventuelt vurdert å lage en variabel som måler rabatten til hver kunde - f.eks. som differansen mellom tariff- og nettopremie, eventuelt som en prosent. Hvis man gjør dette bør man forsikre seg om at *fordelingen* til den transformerte variabelen er slik at den egner seg i en modell.

Den eneste transformasjonen som brukes i dette løsningsforslaget er en log-transformasjon av nettopremier. Under et histogram over log nettopremier, hvor vi kan se at variabelene har en betydelig mindre ekstrem fordeling. Ellers er det altså mange alternativer i løsningen av oppgaven.

```
df_train %>%
  ggplot(aes(x=log(NET_PREM_TOT))) +
  geom_histogram(bins=50)
```



Q3 - Estimere modeller

Med variabler på plass kan vi estimere noen modeller. Under estimerer vi tre logistiske regresjonsmodeller:

- En referansemodell der vi bruker alle variablene i datasettet uten noen form for utvalg eller transformasjoner.
- En modell der vi tar ut tariffpremien, og bruker en log-transformasjon av nettopremien.
- En modell der vi tar ut alle variablene som ikke har koeffisienter som er statistisk signifikant forskjellige fra null på 5% signifikansnivå.

```
reg1 <- glm(CHURN_30 ~ CTL_TENURE_DAYS + CUSATTR_GENDER + N_OBJ_TOT +
              NET_PREM_TOT + TAR_PREM_TOT + N_OBJ_AU_400 + N_OBJ_FB +
              N_OBJ_PK_HUS + N_OBJ_PK_INNBO + N_OBJ_PK_FRHUS,
            data = df_train,
            family = binomial(link = "logit"))

reg2 <- glm(CHURN_30 ~ CTL_TENURE_DAYS + CUSATTR_GENDER + N_OBJ_TOT +
              log(NET_PREM_TOT) + N_OBJ_AU_400 + N_OBJ_FB +
              N_OBJ_PK_HUS + N_OBJ_PK_INNBO + N_OBJ_PK_FRHUS,
            data = df_train,
            family = binomial(link = "logit"))

reg3 <- glm(CHURN_30 ~ CTL_TENURE_DAYS + log(NET_PREM_TOT),
            data = df_train,
            family = binomial(link = "logit"))

stargazer(reg1, reg2, reg3, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     -------------------------------
##                                 CHURN_30
##                        (1)        (2)        (3)
## ------------------------------------------------
## CTL_TENURE_DAYS    -0.0003*** -0.0003*** -0.0003***
##                     (0.0001)   (0.0001)   (0.0001)
##
## CUSATTR_GENDER       -0.087     -0.080
##                     (0.131)    (0.131)
##
## N_OBJ_TOT            -0.019     -0.062
##                     (0.045)    (0.045)
##
## NET_PREM_TOT       0.0001***
##                    (0.00002)
##
## TAR_PREM_TOT        -0.00002
##                    (0.00002)
##
## log(NET_PREM_TOT)              0.786***   0.596***
##                                (0.127)    (0.076)
##
## N_OBJ_AU_400        0.285***    0.095
##                     (0.109)    (0.108)
##
## N_OBJ_FB             0.156      0.190
##                     (0.158)    (0.156)
##
## N_OBJ_PK_HUS         -0.083     -0.128
##                     (0.089)    (0.090)
##
## N_OBJ_PK_INNBO       -0.151     -0.176
##                     (0.110)    (0.113)
##
## N_OBJ_PK_FRHUS       -0.054     -0.036
##                     (0.099)    (0.100)
##
## Constant           -4.379***  -10.680*** -9.413***
##                     (0.153)    (1.058)    (0.706)
##
## ------------------------------------------------
## Observations         25,138     25,138     25,138
## Log Likelihood    -1,477.357 -1,464.275 -1,473.010
## Akaike Inf. Crit. 2,976.715  2,948.549  2,952.020
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Noen kommentarer til resultatene:

- Det er kun *to* variabler med signifikante koeffisienter: Variabel som måler lengde på kundeforhold og nettopremie.
- Fortegnene på variablene er fornuftige: Jo lengre en kunde har vært kunde, jo lavere sannsynlighet for at kunden churner. Videre har kunder med høy premie høyere churnsannsynlighet. Dette er jo også et rimelig resultat, ettersom det nok er mer interessant for forsikringsselskaper å konkurrere om kunder med høye premier enn med lave premier.

Q4 - Evaluering av modell

Spørsmål a)

Definer variablene Y_i lik 1 dersom kunde nummer i var på vei til å churne og 0 ellers, og $R_i = 1$ dersom kunden ble oppringt og 0 ellers. Vi kan da skrive gevinst fra kunde i er i henhold til tabellen i oppgavesettet som

$$\text{Gevinst for kunde } i = Y_i R_i (1-c) + (1-Y_i) R_i (1-c) + Y_i (1-R_i)(0) + (1-Y_i)(1-R_i)(1)$$

Eller sagt med ord: Alle kunder som vi ringer til gir gevinst $1-c$, mens for alle kundene som vi ikke ringer til så får vi gevinst lik 1 dersom kunden ikke churner, og gevinst lik null dersom kunden churner.

La oss nå si at vi predikerer churnsannsynligheten for kunde nummer i, og skriver den som $\hat{p}_i$. Dersom predikert sannsynlighet overstiger terskelverdien p^* så ringer vi kunden, altså blir $R_i = 1$, og gevinsten blir $1 - c$. Dersom predikert churnsannsynlighet er mindre enn terskelverdien p^* så ringer vi ikke, og gevinsten blir lik $1 - Y_i$ (altså lik 1 dersom $Y_i = 0$ og lik 0 dersom $Y_i = 1$). Det kan vi skrive slik, som funksjon av Y_i, $\hat{p}_i$ og p^*:

$$\text{Gevinst for kunde } i = \begin{cases} 1 - c & \text{dersom } \hat{p}_i > p^* \\ 1 - Y_i & \text{dersom } \hat{p}_i \leq p^* \end{cases}.$$

Trygs totale gevinst er da:

$$\Pi(p^*) = (1 - c) \times \text{Antall kunder vi ringer til} + \text{Antall kunder som blir selv om vi ikke ringer til dem.}$$

Dette kan også uttrykkes presist ved hjelp av den såkalte indikatorfunsjonen (https://en.wikipedia.org/wiki/Indicator_function) som $\Pi(p^*) = (1 - c) \sum_{i=1}^{n} 1(\hat{p}_i > p^*) + \sum_{i=1}^{n} (1 - Y_i) 1(\hat{p}_i < p^*)$.

Spørsmål b)

Fra forrige oppgave ser vi at summen består av to distinkte grupper: de vi ringer til, og de vi ikke ringer til. I praksis er det nok enklest å *sortere* datasettet i synkende rekkefølge med hensyn på $\hat{p}_i$, for så å regne ut gevinsten for hver kunde, og til slutt beregne den kumulative summen av $1 - c$ gjennom hele datasettet.

Vi velger her å bruke den enkleste modellen fra oppgave 3) til å predikere sannsynligheten for churn for kundene i testdatasettet. Samtidig henter vi ut de tilhørende resultatene fra den avanserte manskinlæringsmodellen, og de faktiske utfallene. Vi lagrer også en vektor som inneholder de faktiske utfallene for disse kundene i vektoren `Y`:

```
predictions_logistic <- predict(reg3, newdata = df_test, type = "response")
predictions_advanced <- df_test$pred_gbm
Y <- df_test$CHURN_30
```

(Som en liten notis sjekker vi hvor mange kunder i testdatasettet som faktis churnet:)

```
sum(Y)
```

```
## [1] 51
```

Vi begynner med den første terskelverdien 0.02, og finner vektoren av R-verdier basert på de to vektorene med predikerte sannsynligheter:

```
threshold <- 0.02

R_logistic <- predictions_logistic > threshold
R_advanced <- predictions_advanced > threshold
```

For å bruke formelen for den totale gevinsten må vi finne ut hvor mange kunder som blir ($Y = 0$) selv om vi ikke ringer til dem ($R = 0$):

```
stays_anyway_logistic <- sum(!Y & !R_logistic)
stays_anyway_advanced <- sum(!Y & !R_advanced)
```

Verdien av c er i oppgaveteksten definert til å være 0.025, så vi kan nå regne ut gevinsten til Tryg for hver modell ved å bruken formelen vi regnet ut over:

```
cost <- 0.025

payoff_logistic <- (1 - cost)*sum(R_logistic) + stays_anyway_logistic # Dette svarer til de to leddene i formelen f
or total gevinst
payoff_advanced <- (1 - cost)*sum(R_advanced) + stays_anyway_advanced
```

Vi kan gjøre den samme operasjonen to ganger til for terskelverdiene 0.04 og 0.06, og får ut følgende tabell:

```
##              logistic advanced
## threshold_02 6227.225 6227.900
## threshold_04 6233.875 6235.725
## threshold_06 6234.000 6234.150
```

Vi ser at det ikke er store relative forskjeller mellom modellene eller terskelverdiene, men at det er kombinasjonen av den avanserte maskinlæringsmodellen og terskelverdi 0.04 som gir den største gevinsten for Tryg. Konklusjonen kan bli noe slik som at den avanserte modellen er (litt) bedre å predikere enn den forholdsvis enkle logistiske regresjonsmodellen, og at det ser ut til å eksistere en god

avveining mellom det å ringe for få (flere sier opp) og det å ringe for mange (det koster mye).

En mer komplett analyse av dette problemet ville inneholdt flere terskelverdier og flere prediksjonsmodeller for å muligens oppnå en enda større gevinst.

Spørsmål c)

Et viktig tall som kan danne utgangspunktet for diskusjonen rundt dette spørsmålet er hvilken gevinst vi hadde hatt dersom vi ikke ringte til noen kunder for å forhindre churn (og dermed kanskje ikke hadde trengt å ansette noen analytiker i hele tatt…).

Den gevinsten er rett og slett antallet kunder i testdatasettet vårt som ikke churner:

```
sum(!Y)
```

```
## [1] 6234
```

… som bare er litt dårligere enn det vi klarer å oppnå med den avanserte regresjonsmodellen, men uten at vi har klart for oss hva som er skalaen her.