M3S17 Coursework 2

Yuting Lu

December 9, 2020

Question 1 to 3

First, I split the data frame into a training and a test data set with a 2:1 ratio. With the built-in function from the package "survival", I constructed a Cox PH model of the time to default within the training set.

Here is the summary of the model:

Table 1: Summary of the outcome

Covariate	**Coefficient**	**Pr(>|z|)**
age	-0.0201	<0.05
Employment status =		
Homemaker	0.4474	<0.05
Retired	0.2087	0.3890
Self-employed	0.2258	<0.05
Student	-0.8729	0.6187
Other	0.6526	<0.05
Excluded category: Employed		
Home ownership status =		
Homeowner	-1.1740	<0.05
Living with parents	-0.5632	<0.05
Private tenant	-0.2947	<0.05
Other	-0.4240	<0.05
Excluded category: Council tenant		
Total months at current address	-0.00006	0.8786

Question 4

Below is the plot of an approximation to the baseline hazard rate $h_0(t)$.

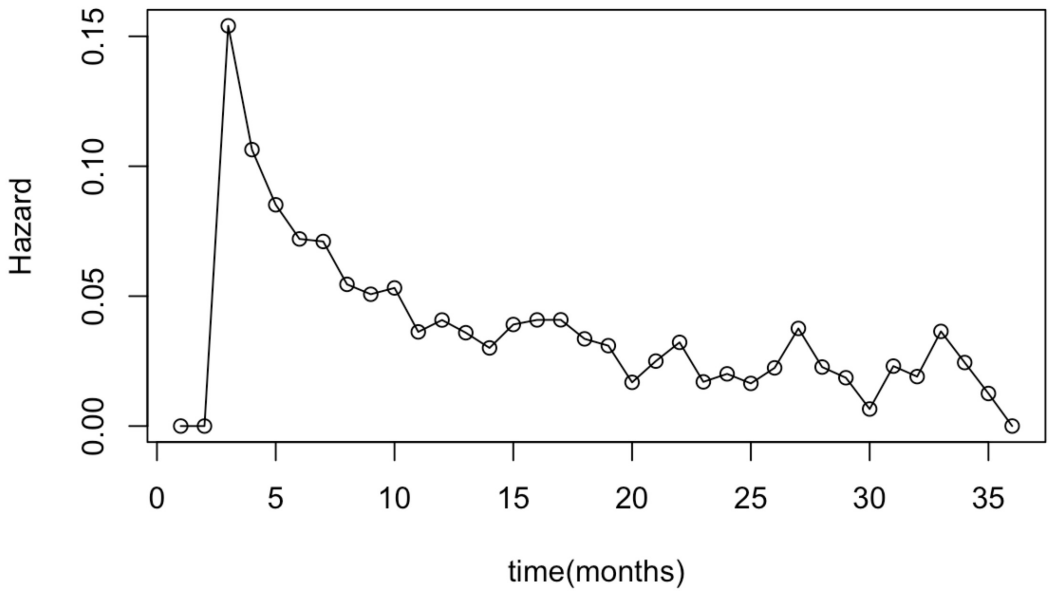

Figure 1: approximation to the baseline hazard rate

Question 5

Question: Compute $\sum log\widehat{S}_i(t_i)$ across all observations i in the training data set.
Answer: -1067.516

Question 6-7

- Deviance on the training data set: 10499.23
- Deviance on the testing data set: 5311.635

Question 8-9

New categorical variable $orig\_6mth$ is added with a level for every 6 month block of originations. ie. 2012.1 to 2012.6 is 1, 2012.7 to 2012.12 is 2, 2013.1 to 2013.6 is 3, 2013.7 to 2013.12 is 4, 2014.1 to 2014.6 is 5, 2014.7 to 2014.12 is 6, 2015.1 to 2015.6 is 7.
Below is the summary of the Cox PH model with the added variable.

Table 2: Summary of Outcome

Covariate	**Coefficient**	**Pr(>|z|)**
age	-0.0203	<0.05
Employment status =		
Homemaker	0.4242	<0.05
Retired	0.1978	0.4134
Self-employed	0.2118	<0.05
Student	-0.1484	0.3986
Other	0.6343	<0.05
Excluded category: Employed		
Home ownership status =		
Homeowner	-1.0500	<0.05
Living with parents	-0.5170	<0.05
Private tenant	-0.2819	<0.05
Other	-0.4197	<0.05
Excluded category: Council tenant		
Total months at current address	-0.00006	0.8942
Origination date=		
2012.7-2012.12	-0.1945	0.0711
2013.1-2013.6	0.1616	0.1660
2013.7-2013.12	-0.0959	0.5079
2014.1-2014.6	0.3302	<0.05
2014.7-2014.12	0.4986	<0.05
2015.1-2015.6	0.3834	0.1105
Excluded category: 2012.1-2012.6		

Question 10

Interpretation:

- Origination dates 2014.1-2014.6 and 2014.7-2014.12 are statistically significant at the 5% level.
- Dates 2012.7-2012.12 and 2013.7-2013.12 have a negative coefficient which indicates that these origination dates reduce default hazard and are less risky. In addition, 2012.7-2012.12 have the lowest risk of being default.
- All the other dates have a positive coefficient which means that they raise default hazard and are more risky. Besides, 2014.7-2014.12 have the highest coefficient and therefore, are the most risky date. The second and third riskiest are 2015.1-2015.6 and 2014.1-2014.6 respectively.

Question 11

Below is the plot of an approximation to the baseline hazard rate $h_0(t)$ for the new model. The first two values are zero because it only counted as default when someone delinquent for three continues months.

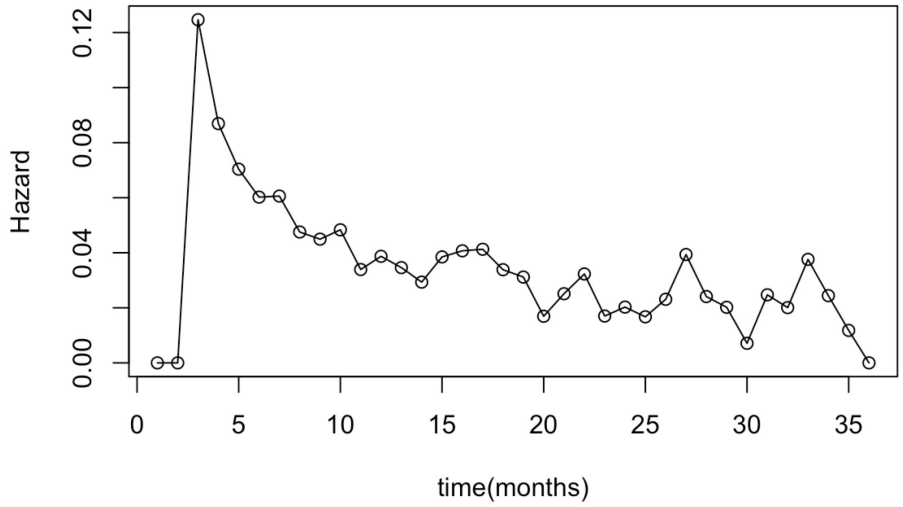

Figure 2: baseline hazard plot for new model

To compare how this plot is different from the baseline hazard plotted for the old model, I put these two plots together. Here red line represents the old model and green line represents the new model.

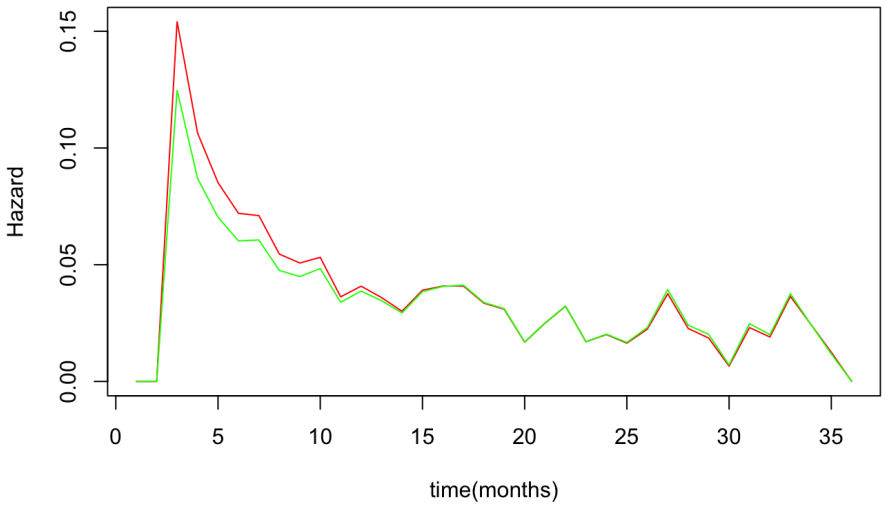

Figure 3: baseline hazard plot for two model

The baseline hazard plot for the new model has the same shape as the one for the old model. However, it can be seen that before t=15 the local maxima for the old model are higher, and then they get very close to each other.

Question 12

The time in baseline hazard plot represents number of months since account opening. It can be seen that for time less than 10 months it has high value. Especially when time is 3 months. This indicates that the longer the account is opened, the less risky the lender is.

Comparing to the answer in question 10 which concluded that over the period 2012 to 2014 the origination date 2014.7-2014.12 has the highest risk and then 2014.1-2014.6. This suggest that lender who opened their account in the year 2014 and have their account opened less time are more likely to default.

The result in question 11 suggest that within the first few months the risk of default is the highest, and then go down with time. This indicates that risky customer tends to default shortly after they open the account.

Question 13

- Deviance on training data set: 10420.44
- Deviance on testing data set: 5290.469

The deviance on the training data set for the new model is 78.79 less than the one for the old model. The deviance on the testing data set for the new model is 21.17 less than the one for the old model. Because higher values of deviance represent worse model fits, this result indicates that the new model which contain the categorical variable $orig\_6mth$ is better than the old one.

Appendix

The associated code is quoted bellow.

```
'''{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(survival)
library(ggplot2)
library(Information)
library(pROC)
'''
```

```
## Question 1
'''{r}
load("CCsurv2.Rdata")                # loading the data
mydf=CC
'''
```

```
## Question 2
'''{r}
set.seed(6666)
ix <- sample(nrow(mydf), round((2/3)*nrow(mydf)))     # spliting the
   data into training and test set
trainset <- mydf[ix,]
testset  <- mydf[-ix,]
'''
```

```
## Question 3
'''{r}
fit <- coxph(Surv(t, def) ~ age + emp + tenure + maa, data=trainset)  #
   building the Cox PH model with training data set
summary(fit)
'''
```

```
## Question 4
'''{r}
mydf_2=mydf
mydf_2$emp_EM=as.factor(mydf$emp=="EM")             # setting all the
   categorical data to dummy variables
mydf_2$emp_HO=as.factor(mydf$emp=="HO")
mydf_2$emp_RE=as.factor(mydf$emp=="RE")
mydf_2$emp_SE=as.factor(mydf$emp=="SE")
mydf_2$emp_ST=as.factor(mydf$emp=="ST")
mydf_2$emp_XX=as.factor(mydf$emp=="XX")
mydf_2$tenure_CT=as.factor(mydf$tenure=="CT")
```

```
mydf_2$tenure_HO=as.factor(mydf$tenure=="HO")
mydf_2$tenure_LP=as.factor(mydf$tenure=="LP")
mydf_2$tenure_PT=as.factor(mydf$tenure=="PT")
mydf_2$tenure_XX=as.factor(mydf$tenure=="XX")

for (p in 8:18){
mydf_2[,p]<- as.numeric(mydf_2[,p]=="TRUE")           # turning the
   categorical variables to 0/1
}
```

```{r}
bHat <- fit$coef
s<-survfit(fit , newdata=testset)                # survival function
sv=s$surv
mydf_3  <- mydf_2[-ix,c(1,9:13,15:18,4)]         # data frame with cols
   that has estimated beta
H0= (- log(sv[,1]) ) / exp( sum( mydf_3 [1 , ]  * bHat ) )  # cumulative
    baseline hazard
h0=diff(H0)                                  # baseline hazard rate
plot(x=1:36, h0 , xlab="time(months)", ylab="Hazard")
lines(x=1:36, h0)
```

```
## Question 5
```{r}
s2<-survfit(fit , newdata=trainset)          # survival function
sv2=s2$surv
si=0*(1:4620)                                # initializing
tt=trainset$t

for (p in 1:4620){
  si[p]= sv2[ tt[p]  , p ]                   # calculating Si(ti)
}

ls2=log(si)                                  # logritham
sum(ls2)                                     # requested outcome
```
```

```
## Question 6
v1
```{r}
Ctrain=trainset
row_to_delete = which(Ctrain$def=="0" )   # deleting censored data
rd2= which(Ctrain$t==36)                  # deleting data with h0=0
```
```

```
Ctrain=Ctrain[-c(row_to_delete,rd2), ]

t1=Ctrain$t
lh=0*(1:1050)                                  # initializing
for (n in 1:1050){
  lh[n] <- log( h0[ t1[n] ] )                  # calculating log(h0(ti))
}
v1=sum(lh)                                     # calculating the first sum
```

v2

```{r}
np2=predict(fit, trainset, type="lp")
rp2=np2 + sum(fit$coefficients*fit$means) # calculating beta times x(ti)

c2=trainset$def                                # getting ci
v2=sum(c2*rp2)                                 # calculating the second sum
```

v3

```{r}
v3=sum(ls2)                                    # calculating the third sum

dev= -2*  (v1+v2+v3  - log(1)  )               # calculating the deviance
dev
```

```
## Question 7
v1
```

```{r}
Ctest=testset                                    # using testing data set
row_to_delete2 = which(Ctest$def=="0" )          # deleting censored data
rd22=  which(Ctest$t==36)                        # deleting data with h0=0
Ctest=Ctest[-c(row_to_delete2,rd22), ]
t12=Ctest$t
lh2=0*(1:530)                                    # initializing
for (n in 1:530){
  lh2[n] <- log( h0[ t12[n] ] )                  # calculating log(h0(ti))
}

v1_2=sum(lh2)                                    # getting the first sum
```

v2

```{r}
```

```
np2_2=predict(fit, testset, type="lp")
rp2_2=np2_2 + sum(fit$coefficients*fit$means)    # calculating beta
   times x(ti)
c2_2=testset$def                                  # getting ci
v2_2=sum(c2_2*rp2_2)                              # the second sum
'''


v3
'''{r}
s3<-survfit(fit , newdata=testset)          # survival function with
   testing data set
sv3=s3$surv
si3=0*(1:2310)                              # initializing
tt3=testset$t                               # ti
for (j in 1:2310){
  si3[j]= sv3[ tt3[j]  , j ]                # calculating Si(ti)
}

ls3=log(si3)                                # log(Si)
v3_2=sum(ls3)                               # the third sum

dev2= -2*  (v1_2+v2_2+v3_2  - log(1)  )     # deviance
dev2
'''
```

```
## Question 8
'''{r}
C8=CC
C8$orig_6mth [ which(C8$orig_month>=1 & C8$orig_month<=6 ) ] <- "1"
C8$orig_6mth [ which(C8$orig_month>=7 & C8$orig_month<=12 ) ] <- "2"
C8$orig_6mth [ which(C8$orig_month>=13 & C8$orig_month<=18 ) ] <- "3"
C8$orig_6mth [ which(C8$orig_month>=19 & C8$orig_month<=24 ) ] <- "4"
C8$orig_6mth [ which(C8$orig_month>=25 & C8$orig_month<=30 ) ] <- "5"
C8$orig_6mth [ which(C8$orig_month>=31 & C8$orig_month<=36 ) ] <- "6"
C8$orig_6mth [ which(C8$orig_month>=37 & C8$orig_month<=42 ) ] <- "7"
                          # creating the new categorical variable
trainset2 <- C8[ix,]
testset2  <- C8[-ix,]
'''
```

```
#Question 9
'''{r}
fit2 <- coxph(Surv(t, def) ~ age + emp + tenure + maa + orig_6mth , data
   =trainset2)  # building the Cox PH model with training data set
summary(fit2)
```

```
```

```
## Question 11
```{r}
mydf_11=mydf
mydf_11$emp_EM=as.factor(mydf$emp=="EM")             # setting all the
   categorical data to dummy variables
mydf_11$emp_HO=as.factor(mydf$emp=="HO")
mydf_11$emp_RE=as.factor(mydf$emp=="RE")
mydf_11$emp_SE=as.factor(mydf$emp=="SE")
mydf_11$emp_ST=as.factor(mydf$emp=="ST")
mydf_11$emp_XX=as.factor(mydf$emp=="XX")
mydf_11$tenure_CT=as.factor(mydf$tenure=="CT")
mydf_11$tenure_HO=as.factor(mydf$tenure=="HO")
mydf_11$tenure_LP=as.factor(mydf$tenure=="LP")
mydf_11$tenure_PT=as.factor(mydf$tenure=="PT")
mydf_11$tenure_XX=as.factor(mydf$tenure=="XX")
mydf_11$orig_6mth1=as.factor(C8$orig_6mth=="1")
mydf_11$orig_6mth2=as.factor(C8$orig_6mth=="2")
mydf_11$orig_6mth3=as.factor(C8$orig_6mth=="3")
mydf_11$orig_6mth4=as.factor(C8$orig_6mth=="4")
mydf_11$orig_6mth5=as.factor(C8$orig_6mth=="5")
mydf_11$orig_6mth6=as.factor(C8$orig_6mth=="6")
mydf_11$orig_6mth7=as.factor(C8$orig_6mth=="7")

for (e in 8:25){
mydf_11[,e]<- as.numeric(mydf_11[,e]=="TRUE")        # turning the
   categorical variables to 0/1
}
```

```{r}
bHat_11 <- fit2$coef
s_11<-survfit(fit2 , newdata=testset2)               # survival function
sv_11=s_11$surv
mydf_12  <- mydf_11[-ix,c(1,9:13,15:18,4,20:25)]     # data frame with
   cols that has estimated beta
H0_11= (- log(sv_11[,1]) ) / exp( sum( mydf_12 [1 , ]  * bHat_11 ) )
       # cumulative baseline hazard
h0_11=diff(H0_11)                                 # baseline hazard rate
plot(x=1:36, h0_11 , xlab="time(months)", ylab="Hazard")
lines(x=1:36, h0_11)
```
```{r}
matplot(x=1:36, cbind(h0,h0_11),type="l",col=c("red","green"),lty=c(1,1)
```
```

```
   , xlab="time(months)", ylab="Hazard")
```

```
## Question 13
deviance_training set
v1
```{r}
Ctr=trainset2                                  # using training data set
row_to_delete_tr = which(Ctr$def=="0"  )       # deleting censored data
rd_tr=  which(Ctr$t==36)                    # deleting the data with h0=0
Ctr=Ctr[-c(row_to_delete_tr,rd_tr), ]
t_tr=Ctr$t
lh_tr=0*(1:1050)                               # initializing
for (n in 1:1050){
  lh_tr[n] <- log( h0_11[ t_tr[n] ] )          # log(ho(ti))
}
v1_tr=sum(lh_tr)                               # the first sum
```

v2
```{r}
np_tr=predict(fit2, trainset2, type="lp")
rp_tr=np_tr + sum(fit2$coefficients*fit2$means)  # calculating beta
   times x(ti)
c2_tr=trainset2$def                              # getting ci
v2_tr=sum(c2_tr*rp_tr)                           # the second sum
```

v3
```{r}
s_tr<-survfit(fit2 , newdata=trainset2)          # survival function
sv_tr=s_tr$surv
si_tr=0*(1:4620)                                 # initializing
tt_tr=trainset2$t                                # ti

for (p in 1:4620){
  si_tr[p]= sv_tr[ tt_tr[p]  , p ]               # calculating Si(ti)
}

ls_tr=log(si_tr)                                 # log(Si(ti))
v3_tr=sum(ls_tr)                                 # the third sum

dev_tr= -2*  (v1_tr+v2_tr+v3_tr  - log(1)  )     # deviance
dev_tr
```
```

```
deviance_testing set
v1
'''{r}
Cte=testset2                                    # using testing data set
row_to_delete_te = which(Cte$def=="0")          # deleting censored data
Cte=Cte[-c(row_to_delete_te), ]
t_te=Cte$t                                      # ti
lh_te=0*(1:530)                                 # initializing

for (n in 1:530){
  lh_te[n] <- log( h0_11[ t_te[n] ] )           # log(h0(ti))
}

v1_te=sum(lh_te)                                # the first sum
'''

v2
'''{r}
np_te=predict(fit2, testset2, type="lp")
rp_te=np_te + sum(fit2$coefficients*fit2$means)   # beta times x(ti)
c2_te=testset2$def                                # ci
v2_te=sum(c2_te*rp_te)                            # the second sum
'''

v3
'''{r}
s_te<-survfit(fit2 , newdata=testset2)            # survival function
sv_te=s_te$surv
si_te=0*(1:2310)                                  # initializing
tt_te=testset2$t                                  # ti

for (j in 1:2310){
  si_te[j]= sv_te[ tt_te[j]  , j ]                # calculating Si(ti)
}

ls_te=log(si_te)                                  # log(Si(ti))
v3_te=sum(ls_te)                                  # the third sum

dev_te= -2*  (v1_te+v2_te+v3_te  - log(1)  )      # deviance
dev_te
'''
```