

LSM

Bojana Milošević

12/26/2019

- Predvideti zavisnu promenljivu
- Šta utiče na raspodelu zavisne promenljive?

$$Y_i|X_i : \begin{pmatrix} 0 & 1 \\ 1 - \pi(X_i) & \pi(X_i) \end{pmatrix}$$

Regresiona funkcija je

$$E(Y_i|X_i) = \pi(X_i)$$

Zašto linearni model nije adekvatan?

- 1 Greške modela ne mogu se modelirati normalnom raspodelom, ili nekom drugom absolutno neprekidnom i simetričnom oko nule.
- 2 Disperzija grešaka modela nije konstantna. Važi:
$$D(Y_i|X_i) = \pi(X_i)(1 - \pi(X_i)) = D(\varepsilon_i).$$
- 3 Regresiona funkcija verovatnoća pa treba da bude zadovoljeno da je $\pi(X_i) \in [0, 1]$.

$$F^{-1}(\pi(X_i)) = X\beta$$

- 1 $F(x) = \Phi(x)$ PROBIT regresija
- 2 $F(x) = \frac{1}{1+e^{-x}}$ LOGISTIČKA regresija
- 3 $F(x) = 1 - e^{-e^x}$ LOG-VEJBULOVA regresija

$$\Phi^{-1}(\pi(X_i)) = \beta_0^* + \beta_1^* X_i$$

Pretpostavimo da ispitujemo zavisnost temperature Y od vlažnosti vazduha X i da se proglašava vanredno stanje ukoliko temperatura pređe neki kritični nivo C . Neka je Y_c indikator vanrednog stanja. Pod pretpostavkom da je $Y_i = aX_i + b + \varepsilon_i$, gde $\{\varepsilon_i\}$ je Gausov beli шум, modeliranje $E(Y_c|X)$ se svodi na probit regresiju.

Parametri se mogu oceniti metodom maksimalne verodostojnosti.

Log – Wejbulova regresija

$$F_G^{-1}(\pi(X_i)) = \log(-\log(1 - \pi(X_i))) = \beta_0 + \beta_1 X_1$$

zbog svoje asimetričnosti, se najčešće koristi za modelovanje malih i velikih verovatnoća uspeha

Parametri se mogu oceniti metodom maksimalne verodostojnosti.

$$F_L^{-1}(\pi(X_i)) = \log\left(\frac{\pi(X_i)}{1 - \pi(X_i)}\right) = \beta_0 + \beta_1 X_i$$

$\lambda(p) = \log\left(\frac{p}{1-p}\right)$ logit transformacija

$$\lambda(X_i) = \log\left(\frac{\pi(X_i)}{1-\pi(X_i)}\right)$$

Količnik $\frac{\pi(X_i)}{1-\pi(X_i)}$ se naziva kvota.

Interpretacija kvote

Ocena parametara

Metod maksimalne verodostojnosti

$$\begin{aligned}l(\beta) &= \sum_{i=1}^n \left(Y_i \log \left(\frac{\pi(X_i)}{1 - \pi(X_i)} \right) + \log(1 - \pi(X_i)) \right) \\ &= \sum_{i=1}^n Y_i (\beta_0 + \beta_1 X_i) - \sum_{i=1}^n \log(1 + e^{\beta_0 + \beta_1 X_i})\end{aligned}$$

Numerički se rešava sistem $\frac{\partial l(\beta)}{\partial \beta} = 0$

Ocenjena logit funkcija je

$$\hat{\lambda}(X) = \hat{\beta}_0 + \hat{\beta}_1 X$$

U opštem slučaju

$$\hat{\lambda}(X) = X\hat{\beta}$$

$\hat{\beta}$ ima normalnu $\mathcal{N}(\beta, I^{-1}(\beta))$ raspodelu kao ocena maksimalne verodostojnosti pa se mogu napraviti intervali poverenja za $\lambda(X)$ a zatim i za $\pi(X)$.

Testiranje značajnosti koeficijenta

Valdova statistika

$$Z = \frac{\hat{\beta}_i}{SE(\hat{\beta}_i)}$$

- test količnika verodostojnosti

$$2 \log \left(\frac{L(\hat{\beta}_0, \hat{\beta}_1)}{L(\hat{\beta}_0)} \right) \sim \chi_1^2$$

$$2 \log \left(\frac{L(\hat{\beta})}{L(\hat{\beta}_0)} \right) \sim \chi_q^2$$

broj koeficijenata za koje se pretpostavlja da su 0

- meri razliku između pretpostavljenog modela i saturiranog modela
- Definiše se sa $D_0 = 2(l(y, \hat{\theta}_s) - l(y, \hat{\beta}_0))$ gde je $\hat{\theta}_s$ ocena u saturiranom modelu
- $D = 2(l(y, \hat{\theta}_s) - l(y, \hat{\beta}))$
- $D_0 - D \sim \chi_q^2$

Slučaj grupisanih podataka

najčešće kad je neki prediktor kategorička promenljiva

Za svako X_i iz uzorka formiramo podskup koji čine oni elementi uzorka čija je nezavisna komponenta jednaka odabranom X_i .

Neka je m_j broj elemenata u j -toj podgrupi posmatranog uzorka, $j = 1, 2, \dots, J$.

U okviru svake podgrupe ocenimo $\pi(X_j) = P\{Y = 1|X_j\}$. Neka je n_j broj elemenata u podgrupi za koje je vrednost zavisne promenljive jednaka 1.

Tada je $\hat{\pi}\{Y = 1|X_j\} = \frac{1}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_j}}$. Očekivan broj elemenata iz svake od grupa:

$$\hat{n}_j = m_j \hat{\pi}_j = m_j \frac{1}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X_j}}.$$

Pirsonovi reziduali

$$r_j = \frac{n_j - m_j \hat{\pi}_j}{\sqrt{m_j \hat{\pi}_j (1 - \hat{\pi}_j)}} = \frac{n_j - \hat{n}_j}{\sqrt{\hat{n}_j (1 - \frac{\hat{n}_j}{m_j})}}$$

$$C = \sum_{j=1}^J r_j^2.$$

C ima približno χ_{J-2}^2 . Kada se u modelu javlja $p + 1$ ocenjen parametar onda C ima približno χ_{J-p-1}^2 raspodelu

Reziduali devijacije

Rezidual devijacije, za $n_j - \hat{n}_j > 0$, je definisan sa:

$$\begin{aligned}d_j &= \sqrt{2 \left(n_j \ln \frac{n_j}{m_j \hat{\pi}_j} + (m_j - n_j) \ln \frac{m_j - n_j}{m_j (1 - \hat{\pi}_j)} \right)} \\ &= \sqrt{2 \left(n_j \ln \frac{n_j}{\hat{n}_j} + (m_j - n_j) \ln \frac{m_j - n_j}{m_j - \hat{n}_j} \right)}\end{aligned}$$

Za $n_j - \hat{n}_j < 0$ za j -ti rezidual se uzima $-d_j$, u suprotnom nula.

Specijalni slučajevi:

za $n_j = 0$

$$d_j = -\sqrt{2m_j \left| \ln \frac{m_j}{m_j - \hat{n}_j} \right|},$$

dok je za $n_j = m_j$

$$d_j = \sqrt{2m_j \left| \ln \frac{m_j}{\hat{n}_j} \right|}.$$

Reziduali devijacije

Test statistika je

$$D = \sum_{j=1}^J d_j^2.$$

D ima približno χ_{J-2}^2 raspodelu. Kada se u modelu javlja $p + 1$ ocenjen parametar onda D ima približno χ_{J-p-1}^2 raspodelu. Primetimo da je D zapravo devijacija modela.

Hosmer-Lemešov test

- Podaci se grupišu u g kategorija na osnovu sličnosti ocenjenih verovatnoća. Granice koje određuju grupu se dobijaju kao odgovarajući kvantili. Npr. prva grupa sadrži sve elemente za koje je ocenjena verovatnoća između 0 i 0.1, druga, od 0.1 do 0.2 itd. Statistika se pravi analogno Pirsonovoj u slučaju grupisanih podataka

$$C = \sum_{k=1}^g \frac{(o_k - M_k \bar{\pi}_k)^2}{M_k \bar{\pi}_k (1 - \bar{\pi}_k)},$$

gde je M_k broj elemenata u k -toj grupi, c_k je broj različitih elemenata u k -toj grupi i $o_k = \sum_{j=1}^{c_k} Y_j$ i

$$\bar{\pi}_k = \sum_{j=1}^{c_k} \frac{m_j \hat{\pi}_k}{M_k}$$

Ukoliko je dobar model C ima χ_{g-2}^2 raspodelu

Primer – babyfood

- disease: number with disease
- nondisease: number without disease
- sex: a factor with levels Boy Girl
- food: a factor with levels Bottle Breast Suppl

Primer – babyfood

```
library(faraway)
bolesti <- glm(cbind(disease, nondisease) ~ sex+food,
family=binomial,
babyfood)
summary(bolesti)

##
## Call:
## glm(formula = cbind(disease, nondisease) ~ sex + food, family = binomial,
##      data = babyfood)
##
## Deviance Residuals:
##      1      2      3      4      5      6
## 0.1096 -0.5052  0.1922 -0.1342  0.5896 -0.2284
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6127     0.1124 -14.347 < 2e-16 ***
## sexGirl      -0.3126     0.1410  -2.216  0.0267 *
## foodBreast   -0.6693     0.1530  -4.374 1.22e-05 ***
## foodSuppl    -0.1725     0.2056  -0.839  0.4013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 26.37529  on 5  degrees of freedom
## Residual deviance:  0.72192  on 2  degrees of freedom
## AIC: 40.24
##
## Number of Fisher Scoring iterations: 4
```

Primer – babyfood

```
bolestiSex <- glm(cbind(disease, nondisease) ~ sex,  
family=binomial,  
babyfood)  
summary(bolestiSex)
```

```
##  
## Call:  
## glm(formula = cbind(disease, nondisease) ~ sex, family = binomial,  
## data = babyfood)  
##  
## Deviance Residuals:  
## 1 2 3 4 5 6  
## 2.32857 -0.03126 -2.41107 1.75294 1.04279 -2.34573  
##  
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.89991 0.08966 -21.190 <2e-16 ***  
## sexGirl -0.32613 0.14036 -2.323 0.0202 *  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
## Null deviance: 26.375 on 5 degrees of freedom  
## Residual deviance: 20.899 on 4 degrees of freedom  
## AIC: 56.417  
##  
## Number of Fisher Scoring iterations: 4
```

Primer – babyfood

```
bolestiFood <- glm(cbind(disease, nondisease) ~ food,
family=binomial,
babyfood)
summary(bolestiFood)
```

```
##
## Call:
## glm(formula = cbind(disease, nondisease) ~ food, family = binomial,
## data = babyfood)
##
## Deviance Residuals:
## 1 2 3 4 5 6
## 1.16335 0.05492 1.08856 -1.32306 -0.05930 -1.18475
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.74676 0.09693 -18.022 < 2e-16 ***
## foodBreast -0.67645 0.15281 -4.427 9.57e-06 ***
## foodSuppl -0.17435 0.20531 -0.849 0.396
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 26.375 on 5 degrees of freedom
## Residual deviance: 5.699 on 3 degrees of freedom
## AIC: 43.217
##
## Number of Fisher Scoring iterations: 4
```

Primer – babyfood

```
anova(bolestiSex,bolesti,test="Chi")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model 1: cbind(disease, nondisease) ~ sex
```

```
## Model 2: cbind(disease, nondisease) ~ sex + food
```

```
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
## 1         4      20.8992
```

```
## 2         2       0.7219  2   20.177 4.155e-05 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Primer – babyfood

```
anova(bolestiFood,bolesti,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(disease, nondisease) ~ food
## Model 2: cbind(disease, nondisease) ~ sex + food
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         3      5.6990
## 2         2      0.7219  1   4.9771  0.02569 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

b1=data.frame(Bolest=rep(1,77),Sex=rep("Boy",77),Food=rep("Bottle",77))
b2=data.frame(Bolest=rep(0,381),Sex=rep("Boy",381),Food=rep("Bottle",381))
b3=data.frame(Bolest=rep(1,19),Sex=rep("Boy",19),Food=rep("Supl",19))
b4=data.frame(Bolest=rep(0,128),Sex=rep("Boy",128),Food=rep("Supl",128))
b5=data.frame(Bolest=rep(1,47),Sex=rep("Boy",47),Food=rep("Breast",47))
b6=data.frame(Bolest=rep(0,447),Sex=rep("Boy",447),Food=rep("Breast",447))
b7=data.frame(Bolest=rep(1,48),Sex=rep("Girl",48),Food=rep("Bottle",48))
b8=data.frame(Bolest=rep(0,336),Sex=rep("Girl",336),Food=rep("Bottle",336))
b9=data.frame(Bolest=rep(1,16),Sex=rep("Girl",16),Food=rep("Supl",16))
b10=data.frame(Bolest=rep(0,111),Sex=rep("Girl",111),Food=rep("Supl",111))
b11=data.frame(Bolest=rep(1,31),Sex=rep("Girl",31),Food=rep("Breast",31))
b12=data.frame(Bolest=rep(0,433),Sex=rep("Girl",433),Food=rep("Breast",433))
babyfood1=rbind(b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12)

```

```

babyfood1$Food=factor(babyfood1$Food)
babyfood1$Sex=factor(babyfood1$Sex)
babyfood1$Bolest=factor(babyfood1$Boles)
summary(babyfood1)

```

```

##  Bolest      Sex      Food
##  0:1836   Boy :1099  Bottle:842
##  1: 238   Girl: 975  Supl  :274
##                               Breast:958

```

```
bolestin <- glm(Bolest ~ Sex+Food,  
family=binomial,  
babyfood1)  
summary(bolestin)
```

```
##  
## Call:  
## glm(formula = Bolest ~ Sex + Food, family = binomial, data = babyfood1)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max  
## -0.6030  -0.5218  -0.4409  -0.3795   2.3094  
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  -1.6127     0.1124 -14.347 < 2e-16 ***  
## SexGirl      -0.3126     0.1410  -2.216  0.0267 *  
## FoodSupl    -0.1725     0.2056  -0.839  0.4013  
## FoodBreast  -0.6693     0.1530  -4.374 1.22e-05 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
##      Null deviance: 1478.1  on 2073  degrees of freedom  
## Residual deviance: 1452.4  on 2070  degrees of freedom  
## AIC: 1460.4
```


Da li je odabrani model zadovoljavajući?

```
D=sum(residuals.glm(bolesti,c("deviance"))^2)
```

```
D
```

```
## [1] 0.7219218
```

```
C=sum(residuals.glm(bolesti,c("pearson"))^2)
```

```
pchisq(D,df=6-4,lower.tail = FALSE)
```

```
## [1] 0.6970062
```

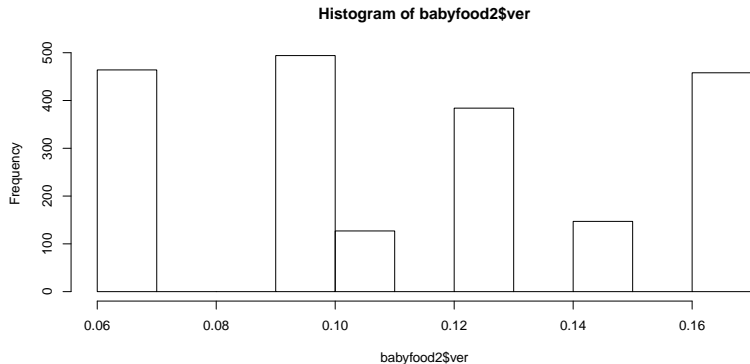
```
pchisq(C,df=6-4,lower.tail = FALSE)
```

```
## [1] 0.6944444
```

Ovu analizu ne smemo da primenimo u slučaju da podaci nisu grupisani.

$$Y_i = \begin{cases} 0, & \hat{p}_i < C \\ 1, & \hat{p}_i \geq C \end{cases}$$

```
predictBolestiN <-predict(bolestiN, type = 'response')
babyfood2=cbind(babyfood1,ver=predictBolestiN,prognoza=(predictBolestiN>=0.5))
hist(babyfood2$ver)
```



C iz skupa {0.5, 0.15, 0.1}

##

FALSE

0 1836

1 238

##

FALSE TRUE

0 1455 381

1 161 77

##

FALSE TRUE

0 880 956

1 78 160

Koje C je adekvatno?

| | | |
|------------------|----------|----------|
| <i>St. : Pr.</i> | 0 | 1 |
| 0 | <i>a</i> | <i>b</i> |
| 1 | <i>c</i> | <i>d</i> |

tačnost klasifikacije je $A = \frac{a+d}{a+b+c+d}$

TNR (true negative rates) - specifičnost $TNR = \frac{a}{a+b}$

FPR $FPR = 1 - TNR$

TPR (true positive rates) - senzitivnost $TPR = \frac{d}{c+d}$

PPV (positive predictive value) $PPV = \frac{d}{b+d}$

F₁ skor $F_1 = 2 \frac{PPV \cdot TPR}{PPV + TPR}$

ROC kriva

Zavisnost sezivnosti od FPR (1- specifičnost) , kad se C menja

Najbolja je u gornjem levom uglu grafika koja odgovara savršenom klasifikatoru. Ukoliko želimo da poredimo dva klasifikatora (modela) možemo koristiti i površinu ispod krive (AUC). Što je veća površina bolja je prediktivna moć modela.

Površina (koja ima oblik Vilkokson-Manove statistike) predstavlja ocenu verovatnoće da će ocenjena verovatnoća uspeha (na osnovu koje se donosi odluka o klasifikaciji) biti veća u slučaju da slučajno odabrani element uzorka dolazi iz pozitivne klase, nego ukoliko dolazi iz negativne klase.

Površina nam zapravo govori koliko dobro ovaj model razdvaja klase, i kako ne zavisi od praga, češće se koristi za poređenje modela.

Mera $2 * AUC - 1$ se naziva Đinijev koeficijent.

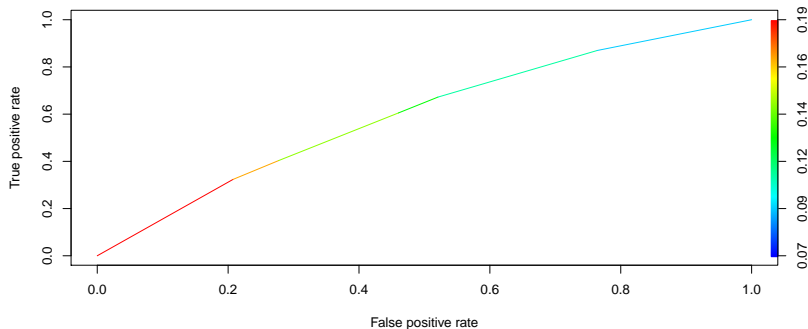
Još jedna mera koja se često koristi je

$$KS = \sup_C TPR(C) - FPR(C)$$

Veća vrednost ove statistike upućuje na bolji klasifikator.

```
library(ROCR)
ROCRpred <- prediction(predictBolestiN, babyfood1$Bolest)
ROCRperf <- performance(ROCRpred, 'tpr', 'fpr')

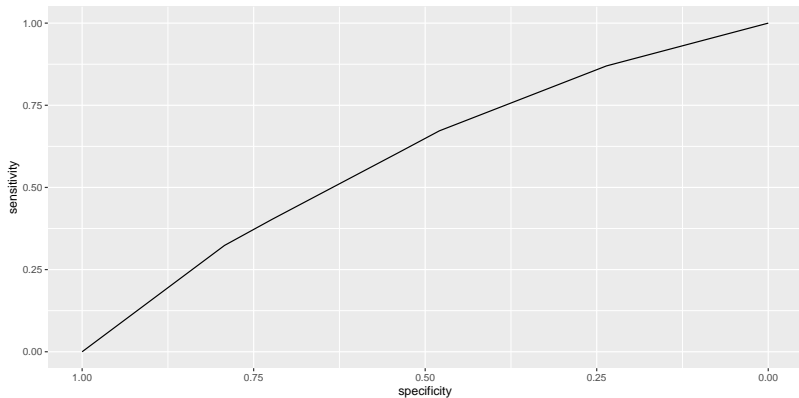
plot(ROCRperf, colorize = TRUE)
```




```
library(pROC)
g <- roc(babyfood2$Bolest ~ babyfood2$ver, data = babyfood2)
auc(g)
```

```
## Area under the curve: 0.598
```

```
ggroc(g)
```



KS statistika

```
tpr <- coords(g, g$thresholds, input="thr", ret="tpr", transpose = FALSE)
fpr <- coords(g, g$thresholds, input="thr", ret="fpr", transpose = FALSE)

tpr-fpr

## [1] 0.0000000 0.1055867 0.1515717 0.1448024 0.1261282 0.1160131 0.0000000
max(tpr-fpr)

## [1] 0.1515717
```

Optimni prag

```
coords(g, 'best', best.method=c("youden"), ret='threshold')
```

```
## [1] 0.1009679
```

```
coords(g, 'best', best.method=c("closest.topleft"), ret='threshold')
```

```
## [1] 0.1182937
```

```
coords(g, 'best', best.method=c("youden"),  
       ret=c('threshold', "1-specificity", 'sensitivity'))
```

```
##      threshold 1-specificity  sensitivity  
##      0.1009679      0.5206972      0.6722689
```

```
coords(g, 'best', best.method=c("closest.topleft"),  
       ret=c('threshold', "1-specificity",  
             'sensitivity'))
```

```
##      threshold 1-specificity  sensitivity  
##      0.1182937      0.4602397      0.6050420
```

Primer - upis na fakultet

```
mydata <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")

mydata$rank <- factor(mydata$rank)

mylogit <- glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial")

summary(mylogit)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = "binomial",
##      data = mydata)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6268  -0.8662  -0.6388   1.1490   2.0790
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre           0.002264   0.001094   2.070 0.038465 *
## gpa           0.804038   0.331819   2.423 0.015388 *
## rank2        -0.675443   0.316490  -2.134 0.032829 *
## rank3        -1.340204   0.345306  -3.881 0.000104 ***
## rank4        -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
```

```
mylogit1 <- glm(admit ~ gre + rank, data = mydata, family = "binomial")
summary(mylogit1)
```

```
##
## Call:
## glm(formula = admit ~ gre + rank, family = "binomial", data = mydata)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5199  -0.8715  -0.6588   1.1775   2.1113
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.802365   0.672982  -2.678 0.007402 **
## gre          0.003224   0.001019   3.163 0.001562 **
## rank2       -0.721737   0.313033  -2.306 0.021132 *
## rank3       -1.291305   0.340775  -3.789 0.000151 ***
## rank4       -1.602054   0.414932  -3.861 0.000113 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 464.53  on 395  degrees of freedom
## AIC: 474.53
##
## Number of Fisher Scoring iterations: 4
```

```
anova(mylogit1,mylogit,test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre + rank
## Model 2: admit ~ gre + gpa + rank
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
mylogit2 <- glm(admit ~ gre +gpa, data = mydata, family = "binomial")
summary(mylogit2)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa, family = "binomial", data = mydata)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2730  -0.8988  -0.7206   1.3013   2.0620
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.949378   1.075093  -4.604 4.15e-06 ***
## gre          0.002691   0.001057   2.544  0.0109 *
## gpa          0.754687   0.319586   2.361  0.0182 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 480.34  on 397  degrees of freedom
## AIC: 486.34
##
## Number of Fisher Scoring iterations: 4
```

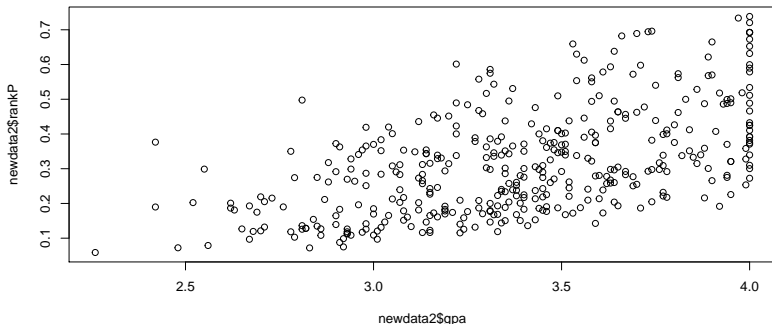
```
anova(mylogit2,mylogit,test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: admit ~ gre + gpa
## Model 2: admit ~ gre + gpa + rank
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         397     480.34
## 2         394     458.52  3    21.826 7.088e-05 ***
```

```
newdata1 <- with(mydata, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
newdata1$rankP <- predict(mylogit, newdata = newdata1, type = "response")
newdata1
```

```
##      gre    gpa rank   rankP
## 1 587.7 3.3899   1 0.5166016
## 2 587.7 3.3899   2 0.3522846
## 3 587.7 3.3899   3 0.2186120
## 4 587.7 3.3899   4 0.1846684
```

```
newdata2 <- mydata
newdata2$rankP <- predict(mylogit, newdata = newdata2, type = "response")
plot(newdata2$gpa, newdata2$rankP)
```



domaći: nacrtati ocenjene verovatnoće za svaku od klasa

```
mydataTr<-mydata[1:300,]
mydataTrPr<-mydata[1:300,]
mydataTest<-mydata[301:400,]
mydataTestPr<-mydata[301:400,]

mylogitTr <- glm(admit ~ gre + gpa + rank, data = mydataTr, family = "binomial")
mylogitTrPr <- glm(admit ~ gre + gpa + rank, data = mydataTrPr, family =binomial(link='probit'))

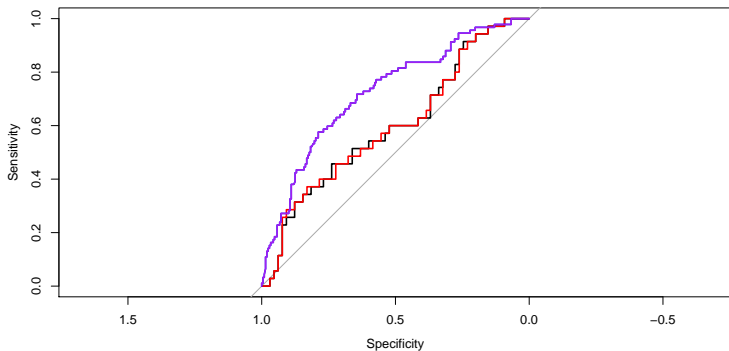
mydataTr$rankP <- predict(mylogitTr, newdata = mydataTr, type = "response")
mydataTest$rankP <- predict(mylogitTr, newdata = mydataTest, type = "response")
mydataTestPr$rankP <- predict(mylogitTrPr, newdata = mydataTest, type = "response")

mydataTrPr$rankP <- predict(mylogitTr, newdata = mydataTr, type = "response")
```



```
g <- roc(mydataTest$admit ~ mydataTest$rankP, data = mydataTest)
plot(g)
gPr <- roc(mydataTestPr$admit ~ mydataTestPr$rankP, data = mydataTest)
plot(gPr,add=TRUE,col='red')

gTr <- roc(mydataTr$admit ~ mydataTr$rankP, data = mydataTr)
plot(gTr,add=TRUE,col='blue')
gTrPr <- roc(mydataTrPr$admit ~ mydataTrPr$rankP, data = mydataTr)
plot(gTrPr,add=TRUE,col='purple')
```



```
auc(gTr)
```

```
## Area under the curve: 0.7258
```

```
auc(gTrPr)
```

```
## Area under the curve: 0.7258
```

```
coords(gTr, 'best', best.method=c("closest.topleft"), ret=c("thr"))
```

```
## Warning in coords.roc(gTr, "best", best.method = c("closest.topleft"), ret
## = c("thr")): An upcoming version of pROC will set the 'transpose' argument
## to FALSE by default. Set transpose = TRUE explicitly to keep the current
## behavior, or transpose = FALSE to adopt the new one and silence this
## warning. Type help(coords_transpose) for additional information.
```

```
## [1] 0.2950917
```

```
coords(g, x=0.3277611, ret=c("tpr", "fpr"), input="thr")
```

```
## Warning in coords.roc(g, x = 0.3277611, ret = c("tpr", "fpr"), input =
## "thr"): An upcoming version of pROC will set the 'transpose' argument
## to FALSE by default. Set transpose = TRUE explicitly to keep the current
## behavior, or transpose = FALSE to adopt the new one and silence this
## warning. Type help(coords_transpose) for additional information.
```

```
##      tpr      fpr
## 0.5428571 0.4307692
```

domaći: uporediti ovaj model sa modelom bez nekog od prediktora prediktora