LATVIJAS UNIVERSITĀTE
FIZIKAS, MATEMĀTIKAS UN OPTOMETRIJAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

ROBUSTA BINĀRĀ KLASIFIKĀCJA AR LOĢISTISKO REGRESIJU

BAKALAURA DARBS

Autore: **Līva Jansone**
Studentes apliecības Nr.: lj08058
Darba vadītāja: lekt. Leonora Pahirko

RĪGA 2019

ANOTĀCIJA

Darbs veltīts loģistiskās regresijas izpētei, kas ir viena no populārākajām metodēm, risinot klasifikācijas uzdevumus. Kļūdas atkarīgajā mainīgajā var būtiski ietekmēt klasificēšanas veiktspēju. Darbā uzsvars likts uz *robustu* loģistiskās regresijas paplašinājumu, kas ir balstīts uz kļūdaino klašu ietveršanu modeļa apmācībā, lai modelis nezaudētu savu klasifikācijas veiktspēju. Modeļa apmācība noris līdzīgi kā loģistiskajā regresijā. Darbā aplūkota gan loģistiskā regresija, gan *robustā* loģistiskā regresija ar nobīdes parametriem. Abas metodes pielietotas praktiski programmā *R Studio* uz reāliem datiem. Darba mērķis ir pārbaudīt, kā abas metodes darbojās un salīdzināt sniegtos rezultātus.

Atslēgas vārdi: loģistiskā regresija, *robusta* loģistiskā regresija, klasifikācija, *glm, glmnet*

ABSTRACT

This work is devoted to the study of logistic regression, which is one of the most popular methods for solving classification tasks. In classification mislabeled data can have a significant impact on the classification performance. The emphasis in the work is on a robust logistic regression expansion based on including mislabeled data into the model training, to improve classification performance. Model training is similar to logistic regression. The paper discusses both logistic regression and robust logistic regression with shift parameters. Both methods are used in practice in program R Studio for real data. The aim of the work is to test performance of both methods and to compare results.

Keywords: Logistic Regression, Robust Logistic Regression, Classification, Glm, Glmnet

Saturs

APZĪMĒJUMI

x_{ij} - neatkarīgie mainīgie, kur $i = 1, \ldots, n$ un $j = 1, .., q$, kur n ir novērojumu skaits un q ir prediktoru skaits.

y_i - atkarīgais mainīgais, $i = 1, \ldots, n$, kur n - novērojumu skaits.

β_i - modeļa parametri, $i = 1, \ldots n$, kur n- parametru skaits.

$\mathbb{E}(Y|x)$ - nosacītā vidējā vērtība (sagaidāmā Y vērtība pie dotā x).

$Y \sim \mathcal{N}(\mu, \sigma^2)$ - gadījuma lielums Y ir sadalīts normāli ar vidējo vērtību μ un dispersiju σ^2.

D - dizaina mainīgais.

IEVADS

Klasifikācijas uzdevumi bieži sastopami medicīnā un mašīnapmācībā. Problēma tiek identificēta kā klasifikācijas problēma, ja atkarīgais mainīgais ir kategorisks un neatkarīgie mainīgie var būt jebkādi. Klasifikācijas uzdevums ir mācīt datorprogrammai izmantot informāciju no ievadītajiem datiem, lai klasificētu jaunos novērojumus. Ja atbildes mainīgajam ir divas iespējamās vērtības, kas kodētas ar 0 un 1, tad ir nepieciešams modelis, kas paredz šīs vērtības kā 0 vai 1.[6] Ir vairāki algoritmi, ko lieto datu klasificēšanai, piemēram, loģistisko regresiju, Naiva Beijesa klasifikatoru (*Naive Bayes Classifier*), atbalsta vektoru mašīnas (*Support Vector Machines*), u.c.[4]. Ja iznākuma mainīgais ir binārs, tad biežāk lietotais ir loģistiskās regresijas modelis. Tātad loģistiskā regresija ir prognozes modelēšanas algoritms, ko izmanto, ja *Y* mainīgais ir binārs jeb dihotoms kategoriskais mainīgais, tas nozīmē, ka tas var pieņemt tikai divas vērtības, ko parasti kodē ar 0 vai 1 (par esošu vai neesošu konkrēto raksturlielumu). Loģistiskās regresijas mērķis ir tāds pats kā jebkuras citas regresijas, tas ir, atrast modeli, kurš vislabāk apraksta attiecības starp atkarīgo (iznākuma vai atbildes mainīgo) un neatkarīgo (prediktoru vai skaidrojošo) mainīgo kopu. Tas ir - noteikt matemātisko vienādojumu, ko izmanto, lai prognozētu notikuma iestāšanās varbūtību.

Lielās datu kopās var būt sastopamas ievades kļūdas, kā, piemēram, nepareizi nokodēts atkarīgais mainīgais. Šādu kļūdu rezultātā atkarīgais mainīgais ir ar nepareizu klasi, kas var ietekmēt modeļa apmācību un modeļa prognozēšanas kvalitāti. Lai izvairītos no šādas problēmas, nepieciešams modelis, kas ir noturīgs jeb *robusts* pret kļūdām datos. Ne vienmēr iespējams identificēt kļūdainos novērojumus, tāpēc tos nevar atmest un ir nepieciešams modelis, kurš apmācības procesā ietver nepareizās klasifikācijas iespējamību modelī. Ir vairākas metodes, kā uzlabot loģistiskās regresijas modeli, lai tas būtu *robusts*.

Darba mērķis ir iepazīties ar loģistiskās regresijas modeli un metodi modeļa uzlabošanai pret kļūdām datos, risinot binārās klasifikācijas uzdevumus, kā arī pielietot praktiski izveidotos modeļus.

Darbs sastāv no trīs daļām. Pirmajā daļā ir aprakstīta loģistiskās regresijas teorija un parametru novērtēšana, aplūkots gan vienfaktora, gan daudzfaktoru modelis, kā arī paskaidrota modeļa interpretācija. Otrajā daļa aprakstīta metode, kā uzlabot loģistiskās regresijas modeli, lai tas būtu *robusts*. Darba trešā daļa ir praktiskā daļa, kurā izveidoti divi modeļi - standarta loģistiskās regresijas modelis un *robusts* loģistiskās regresijas modelis ar nobīdes parametriem, kā arī salīdzināts to sniegums.

Darba teorētiskā daļa balstīta uz *D. Hosmer, S. Lemeshow* un *R. Sturdivant* grāmatu [1] par loģistiskās regresijas pielietojumiem, *J. Tibshirani* publikāciju [8] par *robusta* modeļa izveidi, izmantojot nobīdes parametrus. Darba praktiskā daļa - modeļu izveide un to analīze veikta programmā *R Studio*. Izmantotie dati nav publiski pieejami.

1. LOĢISTISKĀ REGRESIJA

Loģistiskās regresijas modelis ir paplašinājums lineārajai regresijai. Vispārināts lineārs modelis (*GLM*) sastāv no trīs komponentēm:

1) Izlases komponentes, kur $Y_1, \dots, Y_N$ ir atbildes mainīgais. Pieņem, ka tie ir neatkarīgi gadījuma lielumi, katrs ar sadalījumu no eksponenciālās saimes. Tas nozīmē, ka Y_i nav vienādi sadalīti, bet tie nāk no vienas sadalījumu saimes;
2) Sistemātiskās komponentes jeb modeļa - tā ir prediktora x_i funkcija, kas ir lineāra parametros un saistīta ar Y_i vidējo vērtību;
3) Saites funkcijas $g(\mu)$, kas sasaista abas modeļa komponentes nodrošinot, ka:

$$g(\mu_i) = \beta_0 + \beta_1 x_i, \tag{1.1}$$

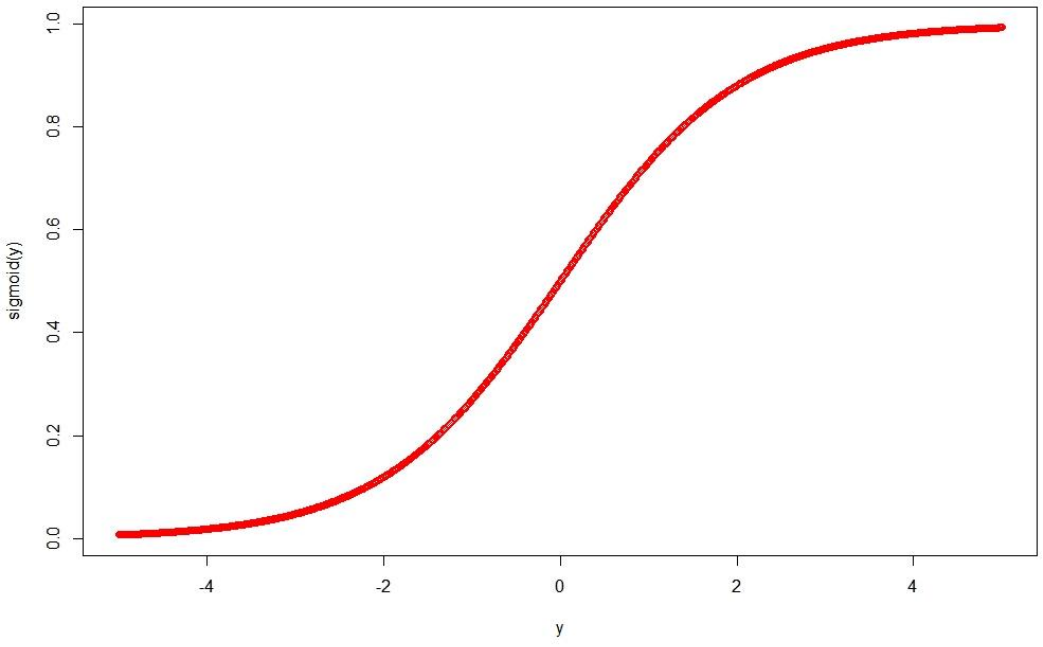
kur $\mu_i = \mathbb{E}Y_i$.

GLM piemērs vienkāršas regresijas gadījumā:

$$\mathbb{E}(Y_i|x_i) = \beta_0 + \beta_1 x_i, \tag{1.2}$$

kur izlases komponentes ir atbildes mainīgais Y_i, $Y_i \sim \mathcal{N}(\mu, \sigma^2)$, sistemātiskā komponente ir regresijas funkcija jeb modelis: $\beta_0 + \beta_1 x_i$, un pieņem, ka $\mu_i = \mathbb{E}Y_i = \beta_0 + \beta_1 x_i$, tas nozīmē, ka saites funkcija ir $g(\mu) = \mu$ [2].

Gadījumā, kad iznākuma mainīgais ir binārs, izmanto loģistiskās regresijas modeli. Loģistiskajā regresijā, lai veiktu prognozes, izmanto loģistisko funkciju jeb *sigmoid* funkciju.



1.1. attēls. ***Sigmoid* funkcija**

Tā ir S formas līkne (sk.1.1.att.), kas pieņem vērtības intervālā [0;1]. To definē ar vienādojumu:

$$sig(y) = \frac{1}{1 + e^{-y}}. \tag{1.3}$$

Līknes galvenais uzdevums ir pareizi klasificēt novērojumus un noteikt nosacīto notikuma iestāšanās varbūtību. Lai veiktu klasifikāciju nepieciešams noteikt slieksni - tas ir punkts, pie kura novērojumi tiek sadalīti atbilstošajās klasēs. Piemēram, ja prognozētā vērtība $\geq 0{,}5$ tad novērojums klasificēts kā 1, pretējā gadījumā kā 0. Dati ir piemēroti lineāram modelim un, izmantojot loģistisko funkciju pie noteikta sliekšņa, tie tiek klasificēti atbilstošajās klasēs 0 un 1.

1.1. Vienfaktora loģistiskās regresijas modelis

Visvienkāršākais gadījums ir vienfaktora loģistiskās regresijas modelis, tas nozīmē, ka ir tikai viens prediktors. Modelī atbildes mainīgais Y ir neatkarīgs gadījuma lielums un $Y \sim Bernoulli(\pi)$, tas nozīmē, ka $Y = 1$ ar varbūtību π un $Y = 0$ ar varbūtību $1 - \pi$ [10].

Aplūkosim nosacīto vidējo vērtību $E(Y|x)$, ko apzīmēsim ar $\pi(x) = E(Y|x)$. Lai analizētu atkarīgo mainīgo $\pi(x)$, izmanto loģistisko sadalījumu. Izmantojot šo sadalījumu, loģistiskās regresijas modeli definē ar vienādojumu:

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}. \tag{1.4}$$

Nosacītā vidējā vērtība ir ierobežota funkcija $0 \leq \pi(x) \leq 1$.

Būtiska nozīme ir *logit* transformācijai, ar kuras palīdzību $\pi(x)$ tiek transformēts uz lineāru funkciju no prediktora x. Ņemot vērā $\pi(x)$, *logit* transformācija tiek definē kā:

$$g(x) = ln\left[\frac{\pi(x)}{1 - \pi(x)}\right] = \beta_0 + \beta_1 x. \tag{1.5}$$

Funkcijai $g(x)$ piemīt daudzas lineārās regresijas modeļa īpašības, tās parametri ir lineāri, tā var būt nepārtraukta un robežās no $(-\infty; +\infty)$[1]. Pārrakstot funkciju $g(x)$ eksponenciālājā formā:

$$g(x) = \pi^y (1 - \pi)^{1-y} = (1 - \pi) exp\left\{y\, ln\left(\frac{\pi(x)}{1 - \pi(x)}\right)\right\} \tag{1.6}$$

redzams, ka $log\left(\frac{\pi}{1-\pi}\right)$ ir eksponenciālās saimes patiesais parametrs un izmantota saites funkcija $g(\pi) = \frac{\pi}{1-\pi}$ [2].

1.2. Loģistiskās regresijas parametru novērtēšana

Pieņemsim, ka dota izlase ar n neatkarīgiem novērojumu pāriem $(x_i, y_i), i = 1,2, \dots, n$, kur x_i ir i- tā neatkarīgā mainīgā vērtība un y_i apzīmē bināro atkarīgo mainīgo. Lai iegūtu loģistiskās regresijas modeli, ir nepieciešams novērtēt nezināmos β parametrus. Viena no metodēm parametru novērtēšanai ir maksimālās ticamības metode (*maximum likelihood*). Lai izmantotu šo metodi, nepieciešams konstruēt ticamības funkciju (*likelihood function*).

Apzīmēsim $P(Y = 1|x_1, \dots, x_n) = \pi(x)$. Tā kā Y ir binārs, tad loģistiskās regresijas vienādojums (1.1) dod nosacīto varbūtību, tas nozīmē, $\pi(x)$ ir nosacītā varbūtība, ka iznākuma mainīgais pieņem vērtību 1 pie dotā x neatkarīgiem β koeficientiem savukārt $1 - \pi(x)$ ir nosacītā varbūtība, ka iznākuma mainīgais pieņem vērtību 0 pie dotā x, $P(Y = 0|x_1, \dots, x_n)$. Tātad tiem (x_i, y_i) pāriem, kur $y_i = 1$ ieguldījums ticamības funkcijai ir $\pi(x_i)$, un tiem pāriem (x_i, y_i), kur $y_i = 0$ ieguldījums ticamības funkcijai ir $1 - \pi(x_i)$, kur $\pi(x_i)$ apraksta $\pi(x)$ vērtību aprēķinātu x_i. Viens no veidiem kā izteikt pāru (x_i, y_i) ieguldījumu ticamības funkcijā ir ar izteiksmi:

$$\pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i}. \tag{1.7}$$

Tā kā novērojumi ir neatkarīgi, tad ticamības funkcija ir izteiksmes (1.7) locekļu reizinājums:

$$l(\beta) = \prod_{i=1}^{n} \pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i}. \tag{1.8}$$

Maksimālā ticamības metode nosaka, ka parametra β novērtējuma vērtība maksimizē izteiksmi (1.8). Lai vienkāršotu aprēķinus, izteiksmi logaritmē:

$$L(\beta) = \ln[l(\beta)] = \sum_{i=1}^{n} \{y_i \ln[\ \pi(x_i)] + (1 - y_i)\ln[1 -\ \pi(x_i)]\}. \tag{1.9}$$

Lai atrastu β vērtības, kas maksimizētu $L(\beta)$ izteiksme (1.9) ir jādiferencē attiecībā pret β un jāpielīdzina nullei:

$$\sum[y_i - \pi(x_i)] = 0 \tag{1.10}$$

un

$$\sum x_i[y_i - \pi(x_i)] = 0. \tag{1.11}$$

Šos vienādojumus sauc par ticamības vienādojumiem (*likelihood equations*). Loģistiskajā regresijā izteiksmes vienādojumos (1.10) un (1.11) ir nelineāras, tāpēc to risināšanai nepieciešamas skaitliskās metodes [1].

1.3. Daudzfaktoru loģistiskās regresijas modelis

Loģistiskās regresijas priekšrocība ir modeļa izveide no vairākiem prediktoriem, no kuriem daži var būt no atšķirīgām mērījumu skalām. Aplūkosim izlasi ar q neatkarīgajiem mainīgajiem, kas apzīmēti ar vektoru $x' = (x_1, x_2, \dots, x_q)$. Pieņemsim, ka katrs no šiem mainīgajiem ir no intervāla skalas. Nosacītā varbūtība iznākuma mainīgajam apzīmēta ar $P(Y = 1|x) = \pi(x)$, tad *logit* transformācija daudzfaktoru loģistiskās regresijas modelim definēta kā:

$$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q. \qquad (1.12)$$

Daudzfaktoru loģistiskās regresijas modelis:

$$\pi(x) = \frac{e^{g(x)}}{1 + e^{g(x)}}. \qquad (1.13)$$

Ja daži no neatkarīgajiem mainīgajiem ir diskrēti, nominālas skalas mainīgie (piemēram, rase, dzimums, ārstēšanas grupa) ir nepareizi tos iekļaut modelī. Skaitļi, ko izmanto, lai attēlotu nominālo mainīgo lielumu ir identifikatori, un tiem nav skaitliskas nozīmes. Šādā gadījumā nepieciešami dizaina (*design*) mainīgie vai fiktīvie (*dummy)* mainīgie. Piemēram, ka viens no neatkarīgajiem mainīgajiem ir rase, kas ir kodēta kā „baltais", „melnādainais" un „cita". Šajā gadījumā ir nepieciešami dizaina mainīgie. Situācijā, kad respondenta rase ir „baltais" abi dizaina mainīgie būtu vienādi ar 0 ($D_1 = D_2 = 0$), kad respondenta rase ir „melnais", tad $D_1 = 1$ un $D_2 = 0$, kad respondenta rase ir „cita", tad $D_1 = 0$ un $D_2 = 1$. Tātad, ja nominālās skalas mainīgajam ir k iespējamās vērtības, tad nepieciešami $k - 1$ dizaina mainīgie [1].

1.4. Daudzfaktoru loģistiskās regresijas modeļa pielāgotība

Pieņemsim, ka dota izlase no n neatkarīgiem novērojumu pāriem (x_{ij}, y_i), kur $i = 1, \ldots, n$ un $j = 1, .. p$. Modeļa pielāgošanai nepieciešams novērtēt vektoru $\beta' = (\beta_0, \beta_1, \ldots, \beta_p)$. Tāpat, kā vienfaktora regresijas gadījumā, lai novērtētu β' izmanto maksimālās ticamības metodi. Ticamības funkcijai (1.8) mainās $\pi(x)$, kas šajā gadījumā definēts, kā vienādojumā (1.13). Iegūsim $p - 1$ ticamības vienādojumus, kuri iegūti diferencējot logaritmēto ticamības funkciju attiecībā pret $p + 1$ koeficientiem. Iegūtos ticamības vienādojumus definē sekojoši:

$$\sum_{i=1}^{n} [y_i - \pi(x_i)] = 0 \tag{1.14}$$

un

$$\sum_{i=1}^{n} x_{ij} [y_i - \pi(x_i)] = 0, \tag{1.15}$$

visiem $j = 1, \ldots, p$. Izmantojot skaitliskās metodes, iegūstam vienādojumu (1.14) un (1.15) atrisinājumu, ko apzīmēsim ar $\hat{\beta}$. Pielāgotās vērtības daudzfaktoru loģistiskās regresijas modelim ir $\hat{\pi}(x_i)$. Vienādojumā (1.13) izteiksmes vērtība aprēķināta izmantojot $\hat{\beta}$ un x_i.

Tālāk aplūkosim metodi dispersijas un novērtēto koeficientu kovariācijas novērtēšanai, kas izriet no maksimālā ticamības novērtējuma. Novērtējumi iegūti no logaritmētās ticamības funkcijas otrās kārtas parciālo atvasinājumu matricas. Otrās kārtas parciālie atvasinājumi:

$$\frac{\partial^2 L(\beta)}{\partial \beta_j^2} = -\sum_{i=1}^{n} x_{ij}^2 \pi_i (1 - \pi_i) \tag{1.16}$$

un

$$\frac{\partial^2 L(\beta)}{\partial \beta_j \partial \beta_l} = -\sum_{i=1}^{n} x_{ij} x_{il} \pi_i (1 - \pi_i) \tag{1.17}$$

visiem $j, l = 0,1, \ldots, p$, kur π_i apzīmē $\pi(x_i)$. Ar $I(\beta)$ apzīmēsim $(p + 1) \times (p + 1)$ matricu, kas satur negatīvās izteiksmes (1.16) un (1.17). Dispersija un kovariācija novērtētajiem koeficientiem ir iegūta no šīs matricas inversās matricas, ko apzīmēsim ar $Var(\beta) = I^{-1}(\beta)$. Tā kā matricas elementus nevar izteikt ar izteiksmi, izmantosim apzīmējumus, lai izteiktu matricas vērtības:

- $Var(\beta_j)$, lai apzīmētu j-to diagonāles elementu, kas ir $\hat{\beta}_j$ dispersija;

- $Cov(\beta_j, \beta_l)$, lai apzīmētu patvaļīgu elementu (ne uz diagonāles), kas ir $\hat{\beta}_j$ un $\hat{\beta}_l$, $j, l = 0, \ldots, p$ kovariācija.

Šie dispersijas un kovariācijas novērtējumi, ko apzīmēsim ar $\widehat{Var}(\hat{\beta})$ un $\widehat{Cov}(\hat{\beta}_j, \hat{\beta}_l), j, l = 0,1, \ldots, p$ ir iegūti novērtējot $Var(\beta)$ un $Cov(\beta_j, \beta_l)$ pie $\hat{\beta}$. Izmantosim $\widehat{Var}(\hat{\beta})$ un $\widehat{Cov}(\hat{\beta}_j, \hat{\beta}_l)$, $j, l = 0,1, \ldots, p$, lai apzīmētu matricas vērtības. Lielākoties tiek izmantotas tikai novērtēto koeficientu novērtētās standartkļūdas:

$$\widehat{SE}(\hat{\beta}_j) = \left[\widehat{Var}(\hat{\beta}_j)\right]^{1/2} \tag{1.18}$$

visiem $j = 0, \ldots, p$.

Tālāk formulēsim informācijas matricu, kas nepieciešama, lai spriestu par modeļa pielāgotību: $\hat{I}(\hat{\beta}) = X'\hat{V}X$, kur

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1q} \\ 1 & x_{21} & x_{22} & \ldots & x_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nq} \end{bmatrix}$$

ir $n \times p + 1$ matrica, kas satur datus par katru prediktoru un

$$\hat{V} = \begin{bmatrix} \widehat{v_1} & 0 & \ldots & 0 \\ 0 & \widehat{v_2} & \ldots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \ldots & 0 & \widehat{v_n} \end{bmatrix},$$

kur V ir $n \times n$ diagonālmatrica, kur $\widehat{v_i} = \hat{\pi}_i(1 - \hat{\pi}_i)$, $i = 1, \ldots, n$ un $\hat{\pi}_i = \hat{\pi}(x_i)$ ir vienādojuma (1.13) vērtība izmantojot $\hat{\beta}$ un i-tā prediktora x_i kovariāciju [1].

1.5. Loģistiskās regresijas modeļa interpretācija

Pēc modeļa izveidošanas, ir svarīgi veikt praktiskus secinājumus no aprēķinātajiem koeficientiem modelī. Pieņemsim, ka izveidotais modelis ir pielāgots un mainīgie modelī ir statistiski nozīmīgi. Modeļa interpretācijai tiek aplūkoti neatkarīgo mainīgo novērtētie koeficienti. Šie koeficienti attēlo atkarīgā mainīgā funkcijas slīpuma vienības izmaiņas neatkarīgajam mainīgajam. Modeļa interpretācija ietver divus galvenos uzdevumus:

1) Noteikt saites funkciju;
2) Definēt vienības izmaiņas neatkarīgajam mainīgajam.

Vispirms jānosaka kāda atkarīgā mainīgā funkcija dos lineāru funkciju neatkarīgajam mainīgajam. Loģistiskas regresijas modelim saites funkcija ir $logit$ transformācija:

$$g(x) = \ln\left\{\frac{\pi(x)}{[1-\pi(x)]}\right\} = \beta_0 + \beta_1 x. \quad (1.19)$$

Slīpuma koeficients ir izmaiņas logit funkcijā, kas atbilst vienas vienības izmaiņām neatkarīgajam mainīgajam, tas nozīmē, ka $\beta_1 = g(x-1) - g(x)$.

Aplūkosim piemēru, lai labāk saprastu koeficientu interpretāciju. Tabulā 1.1. dots nominālas skalas, dihotoms neatkarīgais mainīgais x, kas kodēts kā $x = 1$ un $x = 0$.

1.1. tabula. **Piemērs**

	Neatkarīgais mainīgais	
Atkarīgais mainīgais	$\boldsymbol{x = 1}$	$\boldsymbol{x = 0}$
$\boldsymbol{y = 1}$	$\pi(1) = \frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}}$	$\pi(0) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$
$\boldsymbol{y = 0}$	$1-\pi(1) = \frac{1}{1+e^{\beta_0+\beta_1}}$	$1-\pi(0) = \frac{1}{1+e^{\beta_0}}$

Subjektam ar $x = 1$ un $x = 0$ *logit* starpība ir:

$$g(1) - g(0) = (\beta_0 + \beta_1 \times 1) - (\beta_0 + \beta_1 \times 0) = (\beta_0 + \beta_1) - (\beta_0) = \beta_1 \quad (1.20)$$

Vispirms nepieciešams definēt abas dotās neatkarīgā mainīgā vērtības. Šajā gadījumā tās ir $x = 1$ un $x = 0$. Tālāk šīs vērtības vienādojumā aizstāj priekš $logit$ ar $g(1)$ un $g(0)$. Tad tiek aprēķināta starpība $g(1) - g(0)$. Gadījumā, kad neatkarīgais mainīgs ir dihotoms, kas kodēts ar 0 un 1, iznākums starpībai $g(1) - g(0) = \beta_1$. Tādējādi slīpuma koeficients jeb *logit* starpība ir starpība starp logaritmētajām izredzēm (*log odds*), kad $x = 1$ un $x = 0$. Tālāk definēsim izredžu attiecību (*odds ratio*), ko apzīmēsim ar OR:

$$OR = \frac{\frac{\pi(1)}{[1-\pi(1)]}}{\frac{\pi(0)}{[1-\pi(0)]}}, \quad (1.21)$$

kur izredzes, ka rezultāts ir starp indivīdiem $x = 1$ ir $\pi(1)/[1-\pi(1)]$ un $x = 0$ ir $\pi(0)/[1-\pi(0)]$. Tātad izredžu attiecība ir attiecība izredzēm $x = 1$ pret izredzēm $x = 0$. Ievietojot vienādojumā (1.21) izteiksmes priekš loģistiskās regresijas modeļa varbūtībām, kas dotas tabulā 1.1. iegūstam:

$$OR = \frac{\left(\frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}}\right)}{\left(\frac{1}{1+e^{\beta_0+\beta_1}}\right)} = \frac{\left(\frac{e^{\beta_0}}{1+e^{\beta_0}}\right)}{\left(\frac{1}{1+e^{\beta_0}}\right)} = \frac{e^{\beta_0+\beta_1}}{e^{\beta_0}} = e^{(\beta_0+\beta_1)-\beta_0 = e^{\beta_1}}. \quad (1.22)$$

Līdz ar to loģistiskās regresijas modelim ar dihotomu neatkarīgo mainīgo, kas kodēts kā 0 un 1, attiecības starp izredžu attiecību un regresijas koeficientu ir $OR = e^{\beta_1}$.

Izredžu attiecības izmanto saistības noteikšanai, jo tās aptuveni nosaka, cik iespējams vai neiespējams iznākumam ir būt subjektiem $x = 1$ salīdzinājumā pret subjektiem $x = 0$ [1].

2. *ROBUSTA* LOĢISTISKĀ REGRESIJA

Lielās datu kopās bieži sastopamas kļūdas un tās var būtiski ietekmēt klasifikācijas veiktspēju, tāpēc loģistiskās regresijas modelim nepieciešams *robusts* paplašinājums, kas ietver nepareizās klasifikācijas iespējamību modelī. Piemēram, veicot pētījumus ar lielu datu apjomu, kļūdas var rasties neuzmanības dēļ. Ievadot datus programmā var kļūdīties, kodējot atkarīgo mainīgo, sajaucot klases dažiem novērojumiem. Šādas kļūdas ne vienmēr iespējams identificēt un izlabot, kā rezultātā rodas tādi novērojumi, kur ir klasificēti kā 1 (pozitīvi), bet kuriem notikuma iestāšanās varbūtība ir tuvāka 0 un otrādi. Tātad binārajā gadījumā ar kļūdām datos domāti tādi novērojumi y: ja $y = 1$ un varbūtība $p \to 0$ vai $y = 0$ un varbūtība $p \to 1$ [9].

Viena no pieejām kā uzlabot loģistiskās regresijas modeli ir ieviešot nobīdes (*shift*) parametrus. *Robustā* modeļa apmācība noris līdzīgi kā loģistiskās regresijas gadījumā.

Binārajā loģistiskajā regresijā novērojumam x_i varbūtība tikt prognozētam kā pozitīvam modelēta kā:

$$sig(\beta^T x_i) = \frac{1}{1 + e^{-\beta^T x_i}}. \tag{2.1}$$

Pieņemsim, ka modeļa konstante ir iekļauta svara vektorā β, $\beta \in \mathbb{R}^{m+1}$, kur m ir neatkarīgo mainīgo skaits. *Robustais* paplašinājums ir piešķirt reālu vērtību nobīdes parametru Υ_i katram novērojumam $i = 1, \ldots, n$, tad *sigmoid* funkcija tiek definēta kā:

$$sig(\beta^T x_i + \Upsilon_i) = \frac{1}{1 + e^{-\beta^T x_i - \Upsilon_i}}. \tag{2.2}$$

Uzskatam, ka vairākums novērojumu ir pareizi klasificēti un ar L_1(*lasso* (*least absolute shrinkage and selection operator))* regulējam nobīdes parametrus, lai veicinātu izretinātību (*sparsity*). $y_i \in \{0,1\}$ ir novērojuma i klase un fiksējot $\lambda \geq 0$, mērķa funkcija dota kā:

$$l(\beta, \Upsilon) = \sum_{i=1}^{n} [y_i \; log \; sig(\beta^T x_i + \Upsilon_i) + (1 - y_i) \, log(1 - sig(\beta^T x_i + \Upsilon_i))] - \lambda \sum_{i=1}^{n} |\Upsilon_i|. \tag{2.3}$$

Parametri Υ_i ļauj konkrētiem novērojumiem pārvietoties gar *sigmoid* funkciju, iespējams pat mainīt klasi. Ja novērojums i ir ar pareizu klasi, tad atbilstošais nobīdes parametrs $\Upsilon_i = 0$. Ja novērojums ir pozitīvs, bet iedalīts klasē pie negatīviem, tad iespējams, ka Υ_i ir pozitīvs, un otrādi.

Izmantojot L_1 soda funkciju, varam regulēt arī parametru β, tad mērķa funkcija definēta kā:

$$l(\beta, \Upsilon) = \sum_{i=1}^{n} [y_i \ log\, sig(\beta^T x_i + \Upsilon_i) + (1 - y_i)\, log(1 - sig(\beta^T x_i + \Upsilon_i))] - k \sum_{j=1}^{m} |\beta_j| - \lambda \sum_{i=1}^{n} |\Upsilon_i| \quad [5]. \tag{2.4}$$

Regulēšanas mērķis un nozīme ir:

1) Lai maiņas parametriem nebūtu pārāk liels svars, jo tad tie var būtiski ietekmēt iznākuma mainīgo Y;
2) Lai iznākuma mainīgais Y būtu *robusts*, kad modeli pielieto jauniem datiem;
3) Noregulēšana piešķir 0 svaru neatkarīgajiem mainīgajiem, kuri mums nav svarīgi;
4) Nepiešķir nevienam neatkarīgajam mainīgajam pārlieku lielu svaru, lai tiem nebūtu pārlieku liela ietekme uz Y.

Maiņas parametru pievienošana ir ekvivalenta n netkarīgu mainīgo ieviešanai, kur i-tais jaunais neatkarīgais mainīgais ir 1 novērojumam i un 0 pretējā gadījumā. No tā seko, ka mēs varam pārveidot modeļa matricu un parametru vektoru, un iemācīties loģistiskās regresijas modeli kā parasti. Ja $\beta' = (\beta_0, \dots, \beta_m, \Upsilon_1, \dots, \Upsilon_n)$ un $X' = [X|I_n]$ mērķa funkcija vienkāršojas uz:

$$l(\beta') = \sum_{i=1}^{n} \left[y_i \ log\, sig\left(\beta'^T x_i'\right) + (1 - y_i)\, log\left(1 - sig\left(\beta'^T x_i'\right)\right)\right] - \lambda \sum_{j=m+1}^{m+n} \left|\beta'^{(j)}\right|. \tag{2.5}$$

Pierakstot mērķa funkciju šādā formā redzams, ka tā ir izliekta, tāpat kā loģistiskā regresija ar L_1 soda funkciju ir izliekta.

Lai iegūtu loģistiskās regresijas modeli, mēs atstājam tikai β parametrus un prognozes iegūstam ar:

$$I\{sig(\hat{\beta}^T x) > 0{,}5\}. \tag{2.6}$$

Parametru λ no vienādojuma (2.3) parasti izvēlās ar krosvalidāciju. Pieņemsim, ka parametrs β arī ir regulēts, kā vienādojumā (2.4) izmantojot L_1 regulēšanu, tad nepieciešams optimizēt gan pēc k, gan pēc λ. Šādos gadījumos viss vienkāršāk ir konstruēt viendimensiju modeli, lai būtu jāveic krosvalidācija vienam parametram. Tiek veiktas sekojošas procedūras:

1) Katram apskatāmajam k, tiek meklēta λ vērtība, kas apvienojumā ar k izvēli, maksimizē *robusta* modeļa precizitāti treniņa datu kopai;
2) Veic krosvalidāciju, lai atrastu labāko k izmantojot atbilstošās λ vērtības, kas tika iegūtas pirmajā solī [8].

3. PRAKTISKĀ DAĻA

Praktiskajā daļā ir izmantoti dati par pacientiem, kuri vismaz jau gadu slimo ar diabētu. Datu kopa sastāv no atkarīgā mainīgā *TG_merk_kat*, kas ir tiglicerīdu daudzums organismā, kas kodēts kā:

$$TG\_merk\_kat = \begin{cases} 0\text{ , ja tiglicerīdu daudzums } \geq 1{,}7\ mmol/L \\ 1\text{, ja tiglicerīdu daudzums } < 1{,}7\ mmol/L \end{cases}$$

un vairākiem neatkarīgajiem mainīgajiem. Mērķis ir noskaidrot, kādi faktori ietekmē iespējamību sasniegt vēlamo tiglicerīdu daudzumu organismā (tiglicerīdu daudzums $<$ $1{,}7\ mmol/L$).

Lai izveidotu kvalitatīvu loģistiskās regresijas modeli, jāpievērš uzmanība mainīgo skaitam modelī un tam, lai modelis nebūtu pārpielāgots (*overfit*). Optimālo mainīgo izvēlei izmantosim soļu regresiju un, lai izvairītos no modeļa pārpielāgotības, dati tiks sadalīti divās grupās - treniņa datos un testa datos.

Veicot standarta loģistisko regresiju un *robusto* loģistisko regresiju ar nobīdes parametriem, varēsim salīdzināt iegūtos rezultātus un secināt vai paplašinājums modelim sniedza uzlabojumus.

3.1. Datu sagatavošana

Dati tiek ielasīti programmā *R studio* kā *txt* fails. Tālāk tiek veikti vairāki soļi, lai sagatavotu datus:

1. Mainīgie, kuri izmantoti modeļa veidošanai, tiek pārveidoti par faktoriem;
2. Atlasīti dati regresijai atkarīgajam mainīgajam *TG_merk_kat*;
3. Veikta datu tīrīšana, lai nebūtu trūkstošo vērtību.

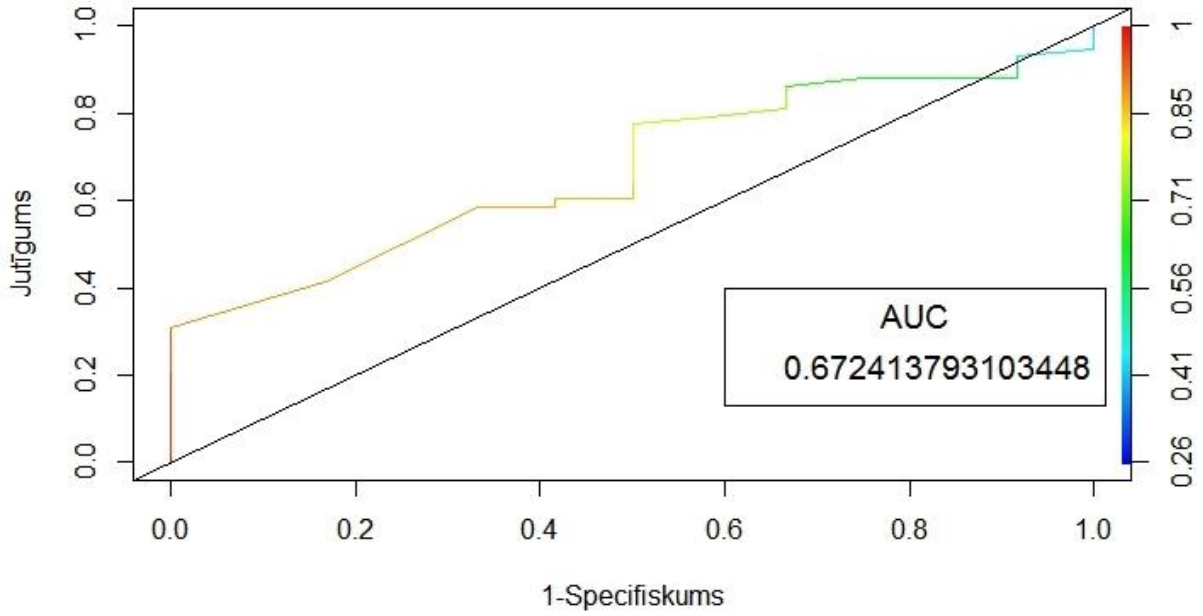
3.2. Loģistiskās regresijas modelis

No sākuma izveidosim standarta loģistiskās regresijas modeli. Vispirms nepieciešams izveidot fiktīvos mainīgos ar funkciju *dummy()* un sadalīt datus treniņa grupā (70% novērojumi)

un testa grupā (30% novērojumi). Modeļa apmācība noris uz treniņa datu kopas, kas sastāv no 163 novērojumiem un 33 prediktoriem. Loģistiskās regresijas modeli veido ar iebūvēto funkciju *glm()*. Šī funkcija piemēro vispārinātu lineāru modeli. Funkcija sastāv no šādiem argumentiem: formulas jeb modeļa, datu kopas un saimes, kas ir kļūdas sadalījums un saites funkcija. Loģistiskajā regresijā modelis sastāv no atkarīgā mainīgā un izvēlētajiem prediktoriem. Par datu kopu ir izvēlēta treniņa datu kopa, jo tajā tiek veikta modeļa apmācība un izvēlētā saime ir binomiālā ar *logit* saites funkciju [7]. Prediktorus modelim izvēlas, izmantojot soļu atlases pievienošanas procedūru (*forward selection*) ar iebūvēto funkciju *step()*. Sāk ar nullto modeli, tas ir modelis, kurā ir tikai konstante (β_0). Ar katru nākamo soli tiek pievienoti mainīgie līdz iegūts optimālais modelis. Modelim tika pievienoti šādi mainīgie: depresija (grupa, kam nav), cd_nefropātija (normoalbuminūrija (vīriešiem $< 2{,}5$; sievietēm $< 3{,}5$)), cd_reinopātija (grupa, ja ārstēta ar LFK), HbA1c (grupa no 5,40 - 8) un fizisko aktivitāšu biežums (fiziskās aktivitātes biežāk kā divas reizes nedēļā).

Tā kā tikai divi no izvēlētajiem mainīgajiem ir statistiski nozīmīgi, aplūkosim tiem izredžu cerības (skat. 2.pielikumu). Varam redzēt, ka, piemēram, viens no faktoriem, kas ietekmē tiglicerīdu vēlamo daudzumu organismā ir depresija. Pacientiem, kam nav depresija, ir $1{,}3679$ reizes lielākas iespējas uz vēlamo tiglicerīdu daudzumu nekā pacientiem ar depresiju. Mainīgajam cd_nefropātija (normoalbuminūrija: vīriešiem $< 2{,}5$ sievietēm < 3,5) izredzes iegūt vēlamo tiglicerīdu daudzumu ir $0{,}9087$ mazākas nekā cd_nefropātija (mikroalbuminūrija: vīriešiem $2{,}5 - 25$, sievietēm $3{,}5 - 35$).

Lai noteiktu modeļa klasifikācijas spējas, testa datiem tiek zīmēta *ROC* (*Receiver operating characteristic*) līkne un apskatīts laukums zem līknes (*AUC*). *ROC* līkne ir grafiska diagramma, ko izmanto, lai parādītu bināro klasifikatoru diagnostikas spēju un rādītājs *AUC* darbojas kā vispārējs prognozēšanas precizitātes rādītājs. Jo *AUC* tuvāks 1, jo precīzāks ir modelis. Pie $AUC = 0.5$ modelis neizšķir pozitīvo un negatīvo vērtību.



3.1. attēls. ***ROC*** **līkne loģistiskās regresijas modelim testa datiem**

Rādītājs $AUC = 0{,}6724$ varam secināt, ka modelis spēj izšķirt novērojumus. *ROC* analīze sniedz arī optimālā atdalīšanas punkta vērtību (*cut-off point*). Atdalīšanas punkts ir punkts, kurš nosaka, kurā klasē dati tiks ievietoti, tātad, tas nodrošina klasifikāciju (0 vai 1). *ROC* līknes optimālais atdalīšanas punkts: $C = 0{,}29$.

Tālāk aplūkosim reālo datu un prognozēto datu diagnostikas krostabulu, pamatojoties uz atdalīšanas punktu (sk. 3.1. tab.).

3.1. tabula. **Diagnostikas krostabula loģistiskās regresijas modelim testa datiem**

		Patiesie dati	
		0	**1**
Prognozētie dati	**0**	1	7
	1	11	51

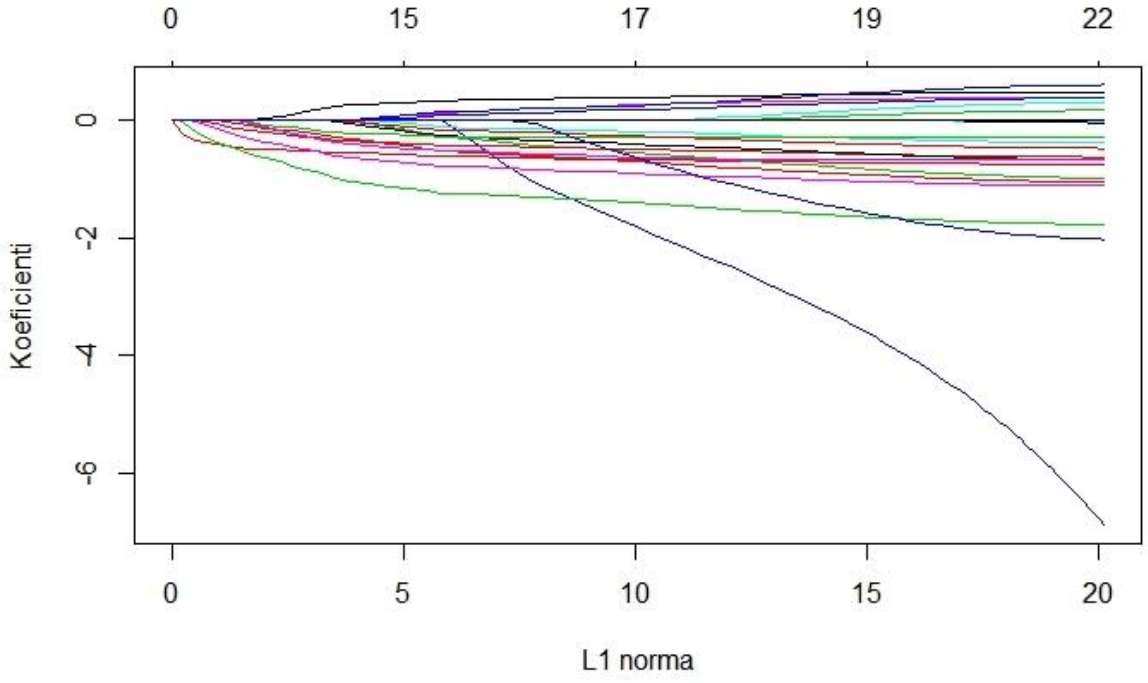
Kā redzams 88% ir pareizi klasificēti, kā novērojumi, kam ir vēlamais tiglicerīdu daudzums un 8% - pareizi klasificēti, kam nav vēlamais tiglicerīdu daudzums.

3.3. Loģistiskās regresijas modelis ar nobīdes parametriem

Modelis veidots izmantojot iebūvēto funkciju *glmnet()*. Funkcija *glmnet()* pielāgo vispārinātu lineāru modeli, izmantojot sodītu maksimālās ticamības funkciju. Regulēšanas ceļš

tiek aprēķināts priekš *lasso un elasticnet* soda funkcijas pie dažādām regulēšanas parametra λ vērtībām. Veidojot modeli izmantota *lasso* metode. Tā ir regresijas analīzes metode, kas veic gan mainīgo izvēli, gan regulēšanu, lai uzlabotu prognozēšanas precizitāti un interpretējamību izveidotajam modelim. *Glmnet()* funkcija sastāv no šādiem argumentiem: prediktoru matricas, atkarīgā mainīgā un saimes. Funkcija *glmnet()* nepieņem kategoriskus neatkarīgos mainīgos, tāpēc prediktoru kopu nepieciešams definēt ar iebūvēto funkciju *model.matrix*, kas izveido matricu un pārveido kategoriskos mainīgos uz fiktīvajiem mainīgajiem. Atkarīgajam mainīgajam mūsu gadījumā jābut faktoram ar diviem līmeņiem un izvēlētā saime loģistiskajā regresijā ir binomiāla. Funkcijā *glmnet()* svarīgi ir norādīt, ka $\alpha = 1$, jo tas nozīmē, ka uz modeli iedarbosies *lasso* metode. Pie dažādām α izvēlēm iespējamas dažādas sodīšanas metodes [3].

Aplūkosim pielāgoto modeli, kas izveidots ar funkciju *glmnet()*. Attēlā 3.2. katra līkne atbilst vienam prediktoram.

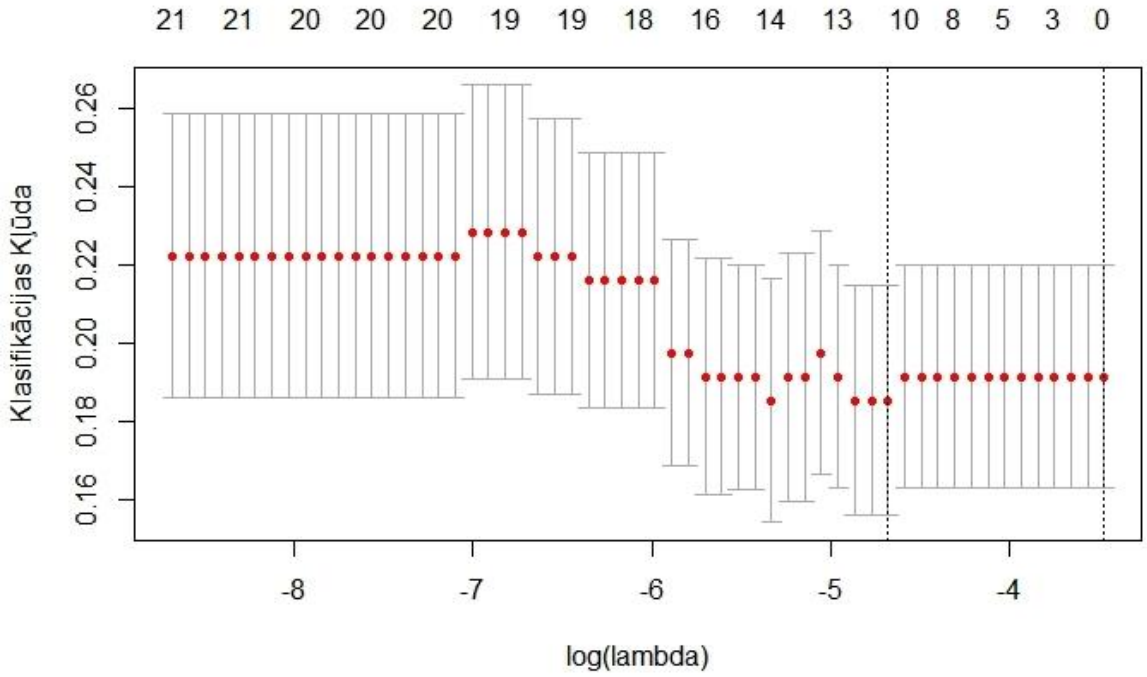


3.2. attēls. ***Glmnet() modelis***

Varam redzēt modeļa prediktoru koeficientu ceļu pret visu koeficientu vektora l_1 normu pie tā kā mainās λ vērtības. Līknes virs ass norāda nenulles koeficientu skaitu pie konkrētas λ vērtības.

Funkcija *glmnet()* atgriež vairākus modeļus no kuriem iespējams izvēlēties vēlamo modeli. Viens no veidiem, kā izvēlēties modeli, ir ar krosvalidāciju, ko piedāvā *cv.glmnet()* funkcija.

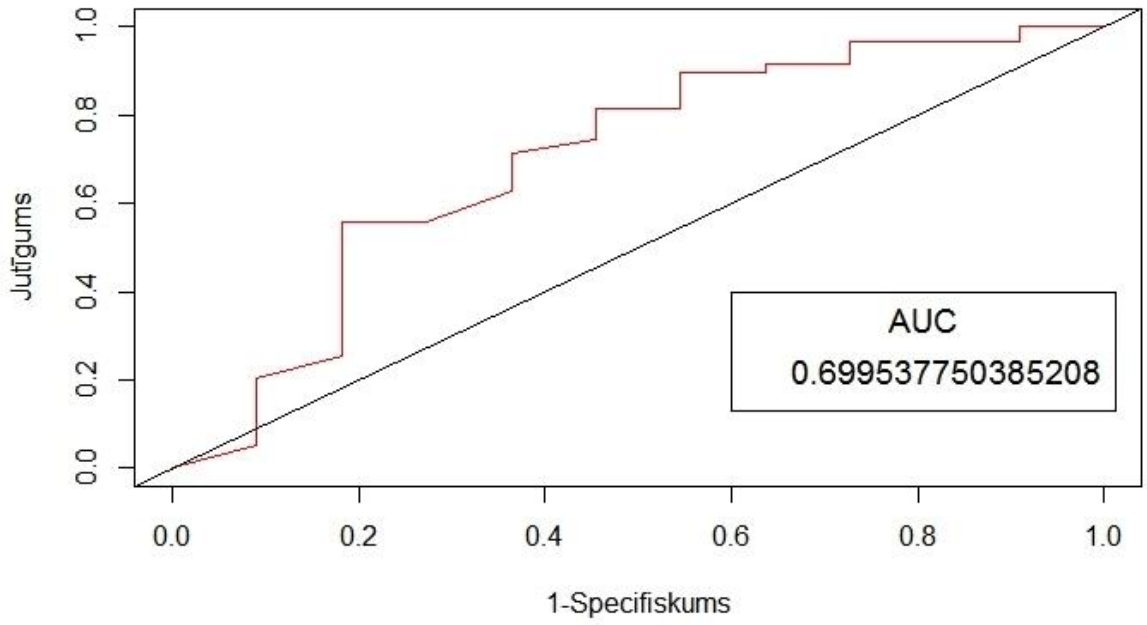
Aplūkosim pielāgoto modeli, kas izveidots ar šo funkciju.



3.3. attēls. ***Cv.glmnet()*** **modelis**

Attēlā 3.3. redzama krosvalidācija (sarkanie punkti) un kļūdas apgabals (*error bars*). Ar punktotu līniju ir atzīmētas divas λ vērtības. Pirmā atzīmētā $\lambda = lambda.min$ atbilst modelim, ar minimālo nepareizas klasifikācijas kļūdu un otrā atzīmētā $\lambda = lambda.1se$ dod λ vērtību, kura dos kļūdu, kas ir vienas standartkļūdas attālumā no minimālās kļūdas [5].

Arī šajā gadījumā, lai analizētu izveidoto modeli aplūkosim *ROC* līkni un *AUC* rādītāju.



3.4. attēls. ***ROC*** **līkne loģistiskās regresijas modelim ar nobīdes parametriem testa datiem**

Rādītājs $AUC = 0{,}6995$ varam secināt, ka modelis spēj izšķirt novērojumus. *ROC* līknes optimālais atdalīšanas punkts: $C = 0{,}76$. Aplūkosim reālo datu un prognozēto datu diagnostikas krostabulu, pamatojoties uz atdalīšanas punktu (sk. 3.2. tab.).

3.2. tabula.

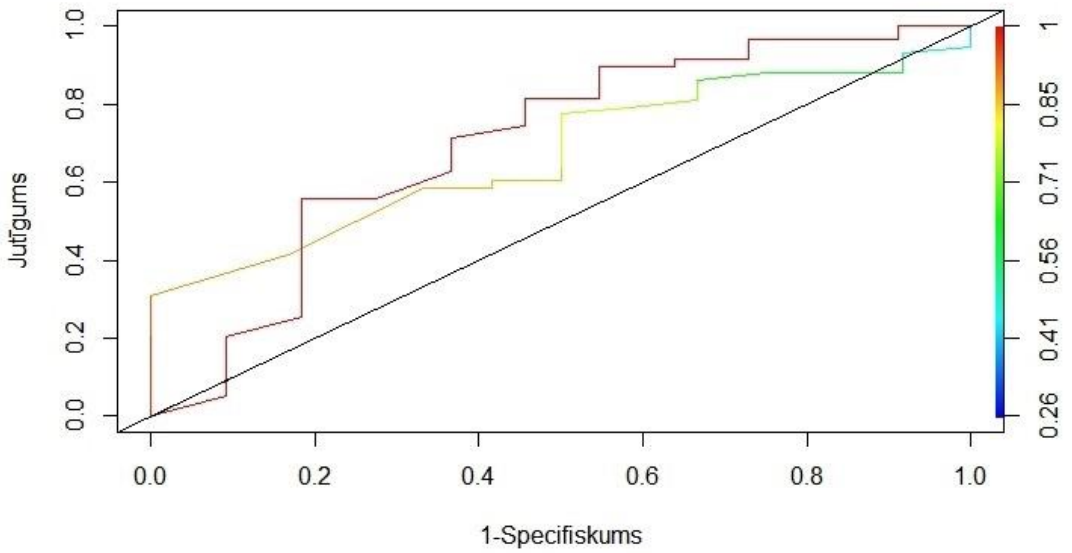
Diagnostikas krostabula loģistiskās regresijas modelim ar nobīdes parametriem testa datiem

		Patiesie dati	
		0	**1**
Prognozētie dati	**0**	1	1
	1	10	58

Kā redzams 98% ir pareizi klasificēti, kā novērojumi, kam ir vēlamais tiglicerīdu daudzums un 9% - pareizi klasificēti, kam nav vēlamais tiglicerīdu daudzums.

3.4. Izveidoto modeļu salīdzināšana

Abus izveidotos modeļus salīdzināsim pēc *ROC* līknes un *AUC* rādītāja. Attēlā 3.5. ar sarkano krāsu attēlota *ROC* līkne loģistiskās regresijas modelim ar nobīdes parametriem, un krāsainā *ROC* līkne standarta loģistiskās regresijas modelim.



3.5. attēls. **Loģistiskās regresijas un *robustas* loģistiskās regresijas ar nobīdes parametriem *ROC* līknes**

Abi izveidotie modeļi sniedz diezgan līdzīgus rezultātus, tomēr modelis, kas veidots ar funkciju *glmnet()* ir nedaudz labāks. Tas apstiprinās arī salīdzinot AUC rādītājus: $AUC_{glm} = 0{,}6724 < AUC_{glmnet} = 0{,}6995$.

SECINĀJUMI

Abas aplūkotās metodes ir noderīgas, lai risinātu binārās klasifikācijas problēmas. Tā kā datos nebija pamatotas aizdomas par kļūdām atkarīgā mainīgā klasēs, abi modeļi sniedza diezgan līdzīgus rezultātus. Varam secināt, ka analizējot konkrētus datus, ir jāizvēlas atbilstošais modelis, lai iegūtu labākos rezultātus. Ne vienmēr nepieciešams *robusts* loģistiskās regresijas modelis, reizēm standarta modelis pietiekami precīzi veiks savu darbu. Loģistiskās regresijas modeli ar nobīdes parametriem ir nepieciešams lietot, kad ir izteiktas aizdomas par kļūdām datos.

Modeļu klasifikācijas veiktspēju ietekmēja arī datu apjoms. Jo lielāks datu apjoms, jo labāk var veikt modeļa apmācību. Pie konkrētā datu apjoma abi modeļi sniedza diezgan līdzīgus rezultātus, secinot pēc *AUC* rādītāja un *ROC* līknes.

Darbā izdevās izveidot modeļus un pielietot tos reāliem datiem un izvērtēt to veiktspēju, kā arī saprast, kuru modeli kādiem datiem labāk lietot.

IZMANTOTĀ LITERATŪRA UN AVOTI

[1] - *D. Hosmer, S. Lemeshow* un *R. Sturdivant* "*Applied Logistic Regression*" 2013.gads

[2] - *G. Casella, R. L. Berger* "*Statistical Inference*" *second edition,* 2002.gads

[3] - https://cran.r-project.org/web/packages/glmnet/glmnet.pdf (skatīts -28.05.2019 plkst. 12:02)

[4] - https://medium.com/@Mandysidana/machine-learning-types-of-classification-9497bd4f2e14 (skatīts - 10.01.2019 plkst. 20:54)

[5] - https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html (skatīts -30.05.2019 plkst. 10:00)

[6] - https://www.machinelearningplus.com/machine-learning/logistic-regression-tutorial-examples-r/ (skatīts - 10.01.2019 plkst. 21.50)

[7] - https://www.rdocumentation.org/packages/stats/versions/3.6.0/topics/glm (skatīts - 28.05.2019 plkst. 12:02)

[8] - *J. Tibshirani* „ *Robust Logistic Regression using Shift Parameters*", *Stanford University*

[9] - *Journal of the royal statistical society. Series B (Methodological)*

[10] - *S. Hosseinian, S.Morgenthales* „*Robust binary regression*" *Journal of Statistical Planning and Inference* 141 (2011) 1497–1509

1.pielikums. **Programmas *R Studio* kods *Sigmoid* funkcijai**

```
y  <-  seq(-5, 5, 0.01)
sigmoid = function(y) { 1 / (1 + exp(-y))}
plot(y, sigmoid(y), col='red')
```

2.pielikums. **Loģistiskās regresijas modelis**

```
Call:
glm(formula = Y ~ Depresija0 + CD_nefropat1 + CD_retinop3 + Fiz.akt_biez3 +
HbA1c_grupas1, family = "binomial", data = train)
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.6581  0.3492  0.5019  0.5464  1.2933
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.2684     0.6118  -0.439  0.66088
Depresija01     1.3679     0.5167   2.647  0.00812 **
CD_nefropat11   0.9087     0.4627   1.964  0.04952 *
CD_retinop31    0.7362     0.5380   1.369  0.17115
Fiz.akt_biez31 -0.7710     0.4417  -1.746  0.08086 .
HbA1c_grupas11  0.7588     0.5334   1.423  0.15487
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
 Null deviance: 155.25  on 161  degrees of freedom
Residual deviance: 137.35  on 156  degrees of freedom
AIC: 149.35
Number of Fisher Scoring iterations: 5
```

3.pielikums. **Programmas *R Studio* kods datu ievadei un sagatavei**

```
library(readxl)
library(dummies)
library(dplyr)
library(ROCR)
#### Ielasa datus
dati<-read.table(file="dati_labotie.txt",header=T,dec=".")
### Dati regresijai atkarīgajamgajam mainīgajam TG_merk_kat, pēc vif
dati_kop<-subset(dati, select=c("Dzimums","TG_merk_kat",  "CD_nefropat", "CD_retinop",
                        "KVS_not","CD_polineirop","AH","HDL.MS","Izglitiba","Invalid",
                        "Depresija","Kop_slodze","Fiz.akt_biez","Soc_ekon",  "Strada",
                        "Fiz.akt.inten","Smeke2","Vecums","KMI","SAS"," CD_stazs",
                        "LDL","GFA.EPI", "HbA1c"))
### Izslēdzu novērojumus, kam ir trūkstošās vērtības
d <- na.omit(dati_kop)
### minīgie, kas netiks izmantoti
d$KVS_not <- NULL
d$KMI <- NULL
d$CD_stazs <-NULL
d$LDL <- NULL
d$GFA.EPI <- NULL
d$SAS <-NULL
### Vecumam, HbA1c jaizvedoju grupas
findInterval(d$Vecums, c(18,25,35,45,55,86))
cut(d$Vecums, breaks=c(18,25,35,45,55,86), right = FALSE)
Vecuma_grupas <- cut(d$Vecums, breaks=c(18,25,35,45,55,86), right = FALSE, labels =
FALSE)
findInterval(d$HbA1c, c(5.4, 8, 11, 13, 17))
cut(d$HbA1c, breaks=c(5.4, 8, 11, 13, 17), right = FALSE)
HbA1c_grupas <- cut(d$HbA1c, breaks=c(5.4, 8, 11, 13, 17), right = FALSE, labels = FALSE)
d$Vecums <- NULL
d$HbA1c <- NULL
### Pievienoju kopai d  izveidotās grupas
d <- cbind(d, Vecuma_grupas)
```

```
d <- cbind(d, HbA1c_grupas)
### Pārveidoju par faktoriem
d$Dzimums <-as.factor(d$Dzimums)
d$TG_merk_kat <- as.factor(d$TG_merk_kat)
d$CD_nefropat <- as.factor(d$CD_nefropat)
d$CD_retinop <- as.factor(d$CD_retinop)
d$CD_polineirop <- as.factor(d$CD_polineirop)
d$AH <-as.factor(d$AH)
d$HDL.MS <- as.factor(d$HDL.MS)
d$Invalid <-as.factor(d$Invalid)
d$Strada <-as.factor(d$Strada)
d$Depresija <-as.factor(d$Depresija)
d$Fiz.akt_biez <-as.factor(d$Fiz.akt_biez)
d$Smeke2 <-as.factor(d$Smeke2)
d$Vecuma_grupas <- as.factor(d$Vecuma_grupas)
d$HbA1c_grupas <- as.factor(d$HbA1c_grupas)
d<-na.omit(d)
### Pārveidoju datu kopu par data frame un noņemu atkarīga mainigā TG_merk_kat kolonnu
dd <- data.frame(d)
ddd<-dd[c("Dzimums","Vecuma_grupas","CD_nefropat","CD_retinop","CD_polineirop",
"AH","HDL.MS","Strada","Depresija","Fiz.akt_biez","Smeke2","HbA1c_grupas")]
### Apzīmēju atkarīgo mainīgo TG_merk_kat ar Y1
Y1 <- dd$TG_merk_kat
### Izveidoju dummY1 mainīgos
fiktivie_mainigie <- dummy.data.frame(ddd, names = NULL, omit.constants=TRUE,
dummy.classes = getOption("dummy.classes"))
### Apvienoju dummY1 mainīgos ar atkarīgo mainīgo vienā kopā
X1 <- cbind(fiktivie_mainigie, Y1)
### Pārveidoju par faktoriem jauno kopu ar dummys
X1$Y1 <- as.factor(X1$Y1)
X1$Dzimums1 <-as.factor(X1$Dzimums1)
X1$Dzimums2 <-as.factor(X1$Dzimums2)
X1$CD_nefropat1 <- as.factor(X1$CD_nefropat1)
X1$CD_nefropat2 <- as.factor(X1$CD_nefropat2)
X1$CD_nefropat3 <- as.factor(X1$CD_nefropat3)
```

```
X1$CD_retinop1 <- as.factor(X1$CD_retinop1)
X1$CD_retinop2 <- as.factor(X1$CD_retinop2)
X1$CD_retinop3 <- as.factor(X1$CD_retinop3)
X1$CD_polineirop0 <- as.factor(X1$CD_polineirop0)
X1$CD_polineirop1 <- as.factor(X1$CD_polineirop1)
X1$AH0 <- as.factor(X1$AH0)
X1$AH1 <- as.factor(X1$AH1)
X1$HDL.MS0 <- as.factor(X1$HDL.MS0)
X1$HDL.MS1 <- as.factor(X1$HDL.MS1)
X1$Strada0 <-as.factor(X1$Strada0)
X1$Strada1 <-as.factor(X1$Strada1)
X1$Depresija0 <-as.factor(X1$Depresija0)
X1$Depresija1 <-as.factor(X1$Depresija1)
X1$Fiz.akt_biez1 <- as.factor(X1$Fiz.akt_biez1)
X1$Fiz.akt_biez2 <- as.factor(X1$Fiz.akt_biez2)
X1$Fiz.akt_biez3 <- as.factor(X1$Fiz.akt_biez3)
X1$Smeke20 <-as.factor(X1$Smeke20)
X1$Smeke21 <-as.factor(X1$Smeke21)
X1$HbA1c_grupas1  <- as.factor(X1$HbA1c_grupas1)
X1$HbA1c_grupas2  <- as.factor(X1$HbA1c_grupas2)
X1$HbA1c_grupas3  <- as.factor(X1$HbA1c_grupas3)
X1$HbA1c_grupas4  <- as.factor(X1$HbA1c_grupas4)
X1$Vecuma_grupas1 <- as.factor(X1$Vecuma_grupas1)
X1$Vecuma_grupas2 <- as.factor(X1$Vecuma_grupas2)
X1$Vecuma_grupas3 <- as.factor(X1$Vecuma_grupas3)
X1$Vecuma_grupas4 <- as.factor(X1$Vecuma_grupas4)
X1$Vecuma_grupas5 <- as.factor(X1$Vecuma_grupas5)
X1$CD_nefropat4 <- NULL
```

4.pielikums. **Programmas *R Studio* kods ļoģistiskās regresijas modelim**

```
### Izveidoju treniņa un testa datu kopu
set.seed(3)
split <-sample(2, nrow(X1), replace = TRUE, prob = c(0.7,0.3))
train <- X1[split==1,]
test <- X1[split==2,]
#Loģistiisās regresijas modelis glm()
fitall<- glm(Y1 ~ Dzimums1 + Dzimums2
        + CD_nefropat1 + CD_nefropat2 + CD_nefropat3
        + CD_retinop1 + CD_retinop2 + CD_retinop3
        + CD_polineirop0 + CD_polineirop1
        + AH0 + AH1 + HDL.MS0 + HDL.MS1
        + Strada0 + Strada1 + Depresija0 + Depresija1
        + Fiz.akt_biez1 + Fiz.akt_biez2 + Fiz.akt_biez3
        + Smeke20 + Smeke21
        + Vecuma_grupas1 + Vecuma_grupas2 + Vecuma_grupas3 + Vecuma_grupas4 +
Vecuma_grupas5
        + HbA1c_grupas1 + HbA1c_grupas2 + HbA1c_grupas3 + HbA1c_grupas4
        , data = train, family = "binomial")
m1 <- glm(Y1 ~ 1, data = train, family = "binomial") #modelis tikai ar konstanti
step(m1, direction = "forward", scope = formula(fitall))
log_reg_mod <- glm(Y1 ~ Depresija0 + CD_nefropat1 + CD_retinop3 + Fiz.akt_biez3 +
HbA1c_grupas1
              , family = "binomial", data = train)
summary(log_reg_mod) # Izveidotais modelis
### modeļa testēšana un kvalitāte
model.test <- predict(log_reg_mod, test, type = "response")
prog1 <- ifelse(model.test > 0.5,1,0)
tab1 <- table(predicted = prog1, Actual =  test$Y1)
tab1
pred1 <- prediction(model.test, test$Y1)
eval1 <- performance(pred1, "acc")
maX1 <- which.max(slot(eval1, "y.values")[[1]])
```

```
acc1<- slot(eval1, "y.values")[[1]][max]
cut1 <- slot(eval1, "x.values")[[1]][max]
print(c(ACCuracy=acc1, Cutoff =cut1))
### ROC likne
roc1 <- performance(pred1, "tpr","fpr")
plot(roc1,colorize=T, Y1lab = "Jutīgums", X1lab="1-Specifiskums")
abline(a=0,b=1)
### AUC
auc1 <-performance(pred1, "auc")
auc1 <-unlist(slot(auc1, "y.values"))
print(auc1)
```

5.pielikums. **Programmas *R Studio* kods *robustas* ļoģistiskās regresijas modelim ar nobīdes parametriem**

```
DK <- na.omit(dati_kop1)
### Izveidoju prediktoru matricu
X <- model.matrix( ~ Dzimums + CD_nefropat + CD_retinop + CD_polineirop + AH
                + HDL.MS + Invalid + Strada + Depresija + Fiz.akt_biez + Smeke2
                + Vecuma_grupas + HbA1c_grupas, DK)
### Atkarīgais mainīgais
Y <- DK$TG_merk_kat
### Modelis ar funkciju glmnet ()
rob.fit <- glmnet(X, Y, family="binomial", standardize=FALSE, alpha = 1)
plot(rob.fit,ylab="Koeficienti", xlab="L1 norma")
#### Sadalu Treniņa un Testa datos
set.seed(3)
N <-232
train_rows <-sample(1:N, .7*N)
x.train <-X[train_rows, ]
x.test <-X[-train_rows, ]
y.train <-Y[train_rows]
y.test <-Y[-train_rows]
### Modelis ar funkciju cv.glmnet()
fit.model <- cv.glmnet(x.train , y.train, family = "binomial", alpha = 1
                    , standardize=FALSE, type.measure = "class")
plot(fit.model, ylab="Klasifikācijas Kļūda",xlab="log(lambda)")
#### Prognoze, ROC, AUC
test.mod <- predict(rob.fit, newx = x.test, s = fit.model$lambda.min)
prog <- ifelse(test.mod > 0.5, 1, 0)
tab <- table(predicted = prog, Actual = y.test)
pred <- prediction(test.mod, y.test)
eval <- performance(pred, "acc")
max <- which.max(slot(eval, "y.values")[[1]])
acc<- slot(eval, "y.values")[[1]][max]
cut <- slot(eval, "x.values")[[1]][max]
```

```
print(c(ACCuracy=acc, Cutoff =cut))
roc <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(roc,col="red", ylab = "Jutīgums", xlab="1-Specifiskums")
abline(a=0,b=1)
### AUC
auc <-performance(pred, "auc")
auc <-unlist(slot(auc, "y.values"))
auc
legend(.6, .4,auc,  title="AUC", cex =1.2)
```

Bakalaura darbs „Robusta binārā klasifikācija ar loģistisko regresiju” izstrādāts LU Fizikas, matemātikas un optometrijas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autore: Līva Jansone

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītāja: lektore Leonora Pahirko

Recenzents: doktorante Līga Bethere

Darbs iesniegts Matemātikas nodaļā 2019.gada___.jūnijā.

Dekāna pilnvarotā persona:

Darbs aizstāvēts valsts pārbaudījuma komisijas sēdē

2019.gada___.jūnijā.

Komisijas sekretāre: