Математички факултет,
Универзитет у Београду

Мастер рад

Проблем класификације небалансираних података

Аутор:
Нађа Обреновић

Ментор:
проф. др Бојана Милошевић

5. август 2024.

Ментор

проф. др Бојана МИЛОШЕВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије

др Марко ОБРАДОВИЋ, доцент
Универзитет у Београду, Математички факултет

проф. др Јована КОВАЧЕВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

Сажетак

Модели машинског учења конструисани за проблеме класификације суочени су са многим изазовима. У ситуацији када расподела категорија зависне променљиве на скупу за обучавање није равномерна, велики број модела има тенденцију да не успостави везу између независних и зависне променљиве оних тачака чија категорија није довољно заступљена. Због тога ова појава има за последицу велике грешке у класификацији таквих тачака.

Класификација небалансираних података завредела је велику пажњу почетком двехиљадитих година. Методе предложене за ефикаснији приступ овом проблему настајале су сукцесивно и данас се могу разврстати у три групе: методе реузорковања, методе осетљиве на цену грешке и методе засноване на ансамблима. У овом раду илустрована је описана појава и дат је преглед метода за решавање овог проблема.

Рад се састоји из четири поглавља и закључка. У првом поглављу уведени су основни појмови који се срећу у овој области и представљен је проблем небалансираности. У другом поглављу дат је преглед појединих модела који су осетљиви на небалансираност и описани су могући узрочници који до ње доводе. Треће поглавље је посвећено методама за решавање проблема небалансираности, а четврто експерименталним резултатима који су добијени коришћењем програмског језика R.

На самом крају рада дата је закључна реч и могући правац даљег истраживања.

Садржај

Поглавље 1

Увод

С порастом популарности машинског односно статистичког учења расла је и потреба за његовим теоријским утемељењем. У том процесу предњачиле су махом математичке дисциплине попут линеарне алгебре, математичке анализе, статистике са теоријом вероватноћа и сл. Стога ово поглавље почињемо дефинисањем основних појмова који се срећу у области машинског учења, са посебним осврtом на задатак класификације.

1.1 Теоријски основи класификације

Задатак класификације јесте предвиђање вредности *категоричке* променљиве на основу датог скупа тзв. *независних* променљивих, при чему под категоричком променљивом подразумевамо ону величину која може узети једну од коначно много унапред познатих вредности. Како би се овакав циљ постигао, проблему се приступа на следећи начин.

Претпоставимо да нам је доступан узорак од n парова тачака $(\boldsymbol{x}_i, y_i)_{i=1}^n$, при чему $\boldsymbol{x_i}$ представља вектор[1] вредности независних променљивих (надаље *атрибута*) $i-$те тачке узорка, а y_i вредност њене *зависне*, односно *циљне* категоричке променљиве. Иако категоричка променљива не мора нужно бити нумеричке природе, зарад лакше поставке рачуна подразумеваћемо да је y_i дато у нумеричком запису. Дати узорак можемо искористити за описивање везе између $\boldsymbol{x_i}$ и y_i на основу које ћемо у будућности давати предикције, што нас доводи до прве дефиниције.

[1] Премда постоје ситуације када се атрибути не морају приказати векторски, у овом раду ћемо подразумевати искључиво такав запис.

Дефиниција 1.1.1. *Функција која изражава везу између атрибута и циљне променљиве назива се модел.*

Одабир одговарајућег модела представља један од највећих изазова у машинском учењу из више разлога. Не само да не постоји коначан број модела који се могу размотрити, већ је и јако тешко пронаћи онај који ће на жељени начин описати однос између променљивих. Поступак идентификације модела на основу датог скупа тачака $(\boldsymbol{x}_i, y_i)_{i=1}^n$ назива се *тренирање* или *обучавање* модела, а поменути скуп се назива скуп за тренирање, односно обучавање (модела). Елементе скупа за тренирање (тестирање) краће ћемо називати тренажним (тестним) тачкама или инстанцама. Пре но што дубље заронимо у сам поступак обучавања модела, осврнућемо се на пар основних термина из теорије вероватноћа.

Нека је $\boldsymbol{X}$ случајан вектор димензије $p \geq 1$ дефинисан на простору вероватноћа $(\Omega, \mathcal{A}, \mathcal{P})$.

Дефиниција 1.1.2. *Функција $\mathcal{P}_{\boldsymbol{X}}$ дефинисана на Бореловој сигма-алгебри $\mathcal{B}^p$ са:*

$$\mathcal{P}_{\boldsymbol{X}}(E) \coloneqq \mathcal{P}(\boldsymbol{X}^{-1}(E)),\ \forall E \in \mathcal{B}^p,$$

назива се вероватноћа индукована случајним вектором $\boldsymbol{X}$.

Дефиниција 1.1.3. *Функција F дефинисана на $\mathbb{R}^p$ са:*

$$F(a) \coloneqq \mathcal{P}_{\boldsymbol{X}}((-\infty, a_1] \times \cdots \times (-\infty, a_p]),\ \forall a = (a_1, \ldots, a_p) \in \mathbb{R}^p,$$

назива се функција расподеле случајног вектора $\boldsymbol{X}$.

Дефиниција 1.1.4. *Функција p дефинисана на $\mathbb{R}^p$ за коју постоји мера μ дефинисана на $\mathcal{B}^p$ таква да важи:*

$$\mathcal{P}_{\boldsymbol{X}}(E) = \int_E p\, d\mu,\ \forall E \in \mathcal{B}^p,$$

назива се функција густине (или само густина) расподеле случајног вектора $\boldsymbol{X}$.

Како ће густина расподеле бити централни појам у наставку овог одељка, наводимо неколико важних примера функције густине.

Пример 1.1.1. *Нека је случајан вектор $\boldsymbol{X}$ дискретног типа, S скуп свих могућих вредности које $\boldsymbol{X}$ може узети и нека је p функција на $\mathbb{R}^p$*

дефинисана са:

$$p(\boldsymbol{x}) = \mathcal{P}\{\boldsymbol{X} = \boldsymbol{x}\},\ \boldsymbol{x} \in \mathbb{R}^p.$$

Означимо са δ бројачку меру на $\mathcal{B}^p$, која је дефинисана на следећи начин:

$$\delta(E) = \begin{cases} |E|, & E \text{ је коначан} \\ +\infty, & E \text{ је бесконачан} \end{cases} \tag{1.1}$$

Тада је p густина[2] *расподеле случајног вектора $\boldsymbol{X}$.*

Доказ:

$$\mathcal{P}_{\boldsymbol{X}}(E) = \mathcal{P}(\boldsymbol{X}^{-1}(E)) = \sum_{\boldsymbol{x_k} \in S \cap E} \mathcal{P}\{\boldsymbol{X} = \boldsymbol{x_k}\} = \sum_{\boldsymbol{x_k} \in S} p(\boldsymbol{x_k})\delta(\{\boldsymbol{x_k}\} \cap E)$$
$$= \int_E p\, d\delta,\ \forall E \in \mathcal{B}^p.$$

■

Пример 1.1.2. *Нека је случајан вектор $\boldsymbol{X}$ апсолутно непрекидног типа и нека је m^p Лебегова мера на $\mathcal{B}^p$. Тада постоји јединствена (до на скуп Лебегове мере 0) ненегативна функција f таква да важи:*

$$\mathcal{P}_{\boldsymbol{X}}(E) = \int_E f\, dm^p,\ \forall E \in \mathcal{B}^p.$$

Доказ о егзистенцији и јединствености функције f може се наћи у [1].

Пример 1.1.3. *Нека су X и Y случајне величине дефинисане на истом простору вероватноћа $(\Omega, \mathcal{A}, \mathcal{P})$, при чему је X дискретног, а Y апсолутно непрекидног типа и нека је S скуп свих могућих вредности које X може узети. Густина случајног вектора (X, Y) је дефинисана са:*

$$f(x, y) = \begin{cases} f_{Y|X=x}(y) \cdot \mathcal{P}\{X = x\}, & x \in S,\ y \in \mathbb{R} \\ 0, & x \notin S,\ y \in \mathbb{R}, \end{cases} \tag{1.2}$$

где је $f_{Y|X=x}$ условна густина расподеле Y при услову $X = x$.

Доказ:

$$\mathcal{P}_{(X,Y)}(E) = \mathcal{P}\{(X, Y) \in E\} = \sum_{x_k \in S} \mathcal{P}\{X = x_k, Y \in E_{x_k}\},$$

[2] Када је реч о дискретним расподелама, термин који чешће користимо је *закон расподеле.*

где је

$$E_x \coloneqq \{y \in \mathbb{R} : (x, y) \in E\}.$$

Даље,

$$\begin{aligned}\sum_{x_k \in S} \mathcal{P}\{X = x_k, Y \in E_{x_k}\} &= \sum_{x_k \in S} \mathcal{P}\{Y \in E_{x_k} | X = x_k\} \cdot \mathcal{P}\{X = x_k\} \\ &= \sum_{x_k \in S} \int_{E_{x_k}} f_{Y|X=x_k}(y)\, dm(y) \cdot \mathcal{P}\{X = x_k\} \\ &= \int_S \int_{E_x} f_{Y|X=x}(y)\, dm(y) \cdot \mathcal{P}\{X = x\}\, d\delta(x) \\ &= \int_S \int_{E_x} f_{Y|X=x}(y) \cdot \mathcal{P}\{X = x\}\, dm(y)\, d\delta(x),\end{aligned}$$

што је по теореми Фубинија једнако

$$= \int_E f_{Y|X=x}(y) \cdot \mathcal{P}\{X = x\}\, d(\delta \times m)(x, y), \quad \forall E \in \mathcal{B}^2,$$

где под $\delta \times m$ подразумевамо производ мера δ и m.

■

Суштина густине расподеле огледа се у разумевању појма *тежине* тачака. Наиме, већа вредност густине у произвољној тачки $\boldsymbol{x} \in \mathbb{R}^p$ значи и већу вероватноћу, односно већу тежину у смислу мере $\mathcal{P}$, скупа тачака које се налазе у њеној околини. Рецимо, на примеру 1.1.1. се јасно види да највећу густину има оно $\boldsymbol{x} \in S$ за које је вероватноћа догађаја $\{\boldsymbol{X} = \boldsymbol{x}\}$ највећа, док ћемо код апсолутно непрекидних расподела посматрати вероватноћу догађаја $\{\boldsymbol{X} \in (\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{h})\}$, за коју важи:

$$\mathcal{P}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{h}) = \int_{(\boldsymbol{x}, \boldsymbol{x}+\boldsymbol{h})} f\, dm^p \approx f(\boldsymbol{x}) \cdot \|\boldsymbol{h}\|, \quad \boldsymbol{x}, \boldsymbol{h} \in \mathbb{R}^p,\ \boldsymbol{h} \to \mathbf{0},$$

где под $\mathbf{0}$ подразумевамо $p-$торку $\underbrace{(0, \ldots, 0)}_{p}$.

Као што смо већ истакли, први корак у обучавању модела јесте одабир узорка $(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)$. Стога је природно посматрати расподелу из које ове тачке долазе.

Нека су $\boldsymbol{X}$ и Y, редом, случајан вектор који одговара вредностима атрибута и случајна величина која одговара вредностима зависне променљиве, дефинисани на заједничком простору вероватноћа $(\Omega, \mathcal{A}, \mathcal{P})$, и нека је p густина расподеле од $(\boldsymbol{X}, Y)$.

За одговарајући модел бисмо могли изабрати оно f за које је $f(\boldsymbol{X})$ на неки начин блиско Y. Формално, потребно је прецизирати метрику којом ћемо мерити разлику између $f(\boldsymbol{X})$ и Y.

Дефиниција 1.1.5. *Функција $L : \mathbb{R}^2 \mapsto \mathbb{R}^+$ која описује разлику између $f(\boldsymbol{X})$ и Y са*

$$L(f(\boldsymbol{X}), Y),$$

назива се функција грешке или функција губитака.

Од суштинског значаја су они модели који што мање греше, односно они за које функција L узима што мање вредности. Међутим, како није природно очекивати да модел f савршено опише оне најмање вероватне парове тачака $(\boldsymbol{x}, y)$, тежићемо ка томе да одабир f буде такав да L буде што мање у што вероватнијим паровима $(\boldsymbol{x}, y)$, односно у оним паровима $(\boldsymbol{x}, y)$ за које је вредност $p(\boldsymbol{x}, y)$ велика. С тим циљем, посматраћемо колико f у просеку греши, јер ће највише утицаја на просек имати баш оне најзаступљеније тачке.

Дакле, погодним моделом можемо сматрати оно f за које је очекивање $EL(f(\boldsymbol{X}), Y)$ мало.

Дефиниција 1.1.6. *Функција R дефинисана са*

$$R(f) \coloneqq EL(f(\boldsymbol{X}), Y) = \int_{\Omega} L(f(\boldsymbol{x}), y) p(\boldsymbol{x}, y)\, d\mathcal{P}_{(\boldsymbol{X},Y)}(\boldsymbol{x}, y)$$

назива се функционал ризика, односно стварни ризик (надаље само ризик).

Ради једноставности записа, убудуће ћемо изостављати скуп по коме се интеграли (уколико је то цео простор), а уместо $d\mathcal{P}_{(\boldsymbol{X},Y)}$ писати само $d\mathcal{P}$.

Сада је јасно да се процес тражења f може свести на следећи оптимизациони проблем:

$$\min_f R(f).$$

Међутим, овакав проблем је тешко решити из више разлога. Први је тај што нам је густина p неретко непозната, те не можемо експлицитно задати R, а други је чињеница да је укупан број модела који се могу размотрити неограничен. Један начин на који можемо премостити другу препреку јесте да одабир потенцијалних модела ограничимо на нешто ужи скуп. Модели који ће се наћи у том скупу обично ће зависити од непознатог параметра ω (који може бити и вишедимензионалан), те ће се потрага за одговарајућим

моделом свести на потрагу за одговарајућим параметром ω. У том случају, нови оптимизациони проблем се своди на:

$$\min_{\omega} R(f_\omega).$$

Но, недоступност p представља нешто већи изазов. Иако се p може моделовати на основу извученог узорка, ради једноставности се приступа моделовању самог ризика.

Дефиниција 1.1.7. *Случајна величина $E(\omega, n)$ дефинисана са*

$$E(\omega, n) = \frac{1}{n} \sum_{i=1}^{n} L(f_\omega(\boldsymbol{X}_i), Y_i)$$

назива се емпиријски ризик.

Дакле, обучавање модела на скупу за тренирање се може посматрати као решавање минимизационог проблема

$$\min_{\omega} E(\omega, n),$$

односно као минимизација емпиријског ризика.

Напоменимо да овакав приступ није теоријски оправдан, јер иако по закону великих бројева важи[3]:

$$E(\omega, n) \xrightarrow{P} R(f_\omega), \quad n \to +\infty,$$

то не имплицира

$$\arg\min_{\omega} E(\omega, n) \to \arg\min_{\omega} R(f_\omega), \quad n \to +\infty$$

Да ли ће доћи до претходне конвергенције с порастом обима узорка, зависиће како од особина скупа функција по којима радимо минимизацију, тако и од избора функције грешке. Може се показати да за посебне класе модела минимизација емпиријског ризика заиста може верно заменити минимизацију стварног ризика. Више о томе може се наћи у [2].

Када је реч о класификацији, чест пример функције грешке дат је са

$$L(u, v) \coloneqq I\{u \neq v\}, \quad \forall (u, v) \in \mathbb{R}^2,$$

[3] за погодно изабрано L

где је I индикаторска функција. Мана ове функције лежи у недостатку информација о томе колико се разликују u и v, јер је за сваки пар различитих тачака вредност L једнака 1. Нарочито у нашој проблематици овакав избор функције грешке не би био користан, јер због своје симетричности L придаје исти значај различитим грешкама класификације. У наставку рада између осталог ћемо видети и како одговарајући избор L може допринети жељеном понашању модела.

Коначно, када смо обучили модел f, можемо извршити класификацију нових тачака.

Дефиниција 1.1.8. *Функција r која пресликава простор атрибута $\mathcal{X}$ у скуп $\{1, 2, \ldots, K\}$, где је K укупан број категорија зависне променљиве, дефинисана са*

$$r(\boldsymbol{x}) = k \iff f(\boldsymbol{x}) \in S_k, \quad \forall \boldsymbol{x} \in \mathcal{X},$$

назива се класификациона функција или класификатор.
Скуп S_k назива се класификационо правило.

1.2 Небалансираност података

Небалансираност података је у многим проблемима свеприсутна појава, која се може дефинисати на различите начине. У контексту моделирања можемо говорити о небалансираности независних или зависне променљиве и о небалансираности нумеричких или категоричких података. Као синоним за неуравнотеженост, небалансираност се може тицати и нерепрезентативности самог узорка.

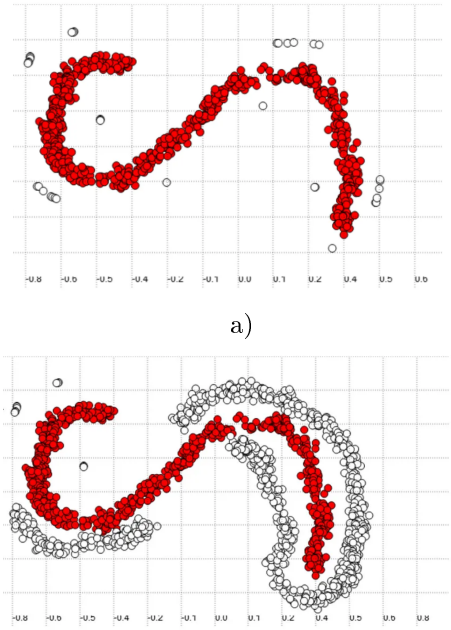
Мотивисани задатом темом, ставићемо акценат на појаву небалансираности која се среће у класификацији. Том приликом разликоваћемо:

1. *међукласну небалансираност* — појаву која се јавља када је расподела зависне променљиве у скупу за обучавање неравномерна;
2. *унутаркласну небалансираност* — појаву која настаје када скуп за обучавање не пати од међукласне небалансираности, али је унутар неке од категорија нерепрезентативан у појединим регионима.

Скица поменутих типова небалансираности дата је на слици 1.1. У овом раду опсежније ћемо се бавити проблемом међукласне небалансираности,

са надом да ћемо поставити темељ и за будућа истраживања на пољу небалансираности другог типа.

Уколико не буде наглашено другачије, под појмом небалансираности подразумеваћемо искључиво међукласну небалансираност.



а)

б)

Слика 1.1 : а) међукласна и б) унутаркласна небалансираност

1.3 Порекло проблема

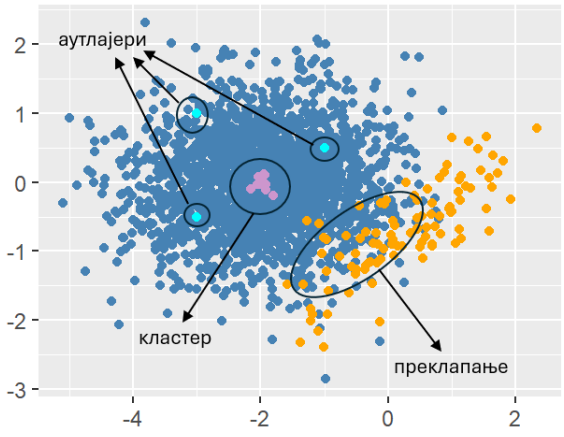
Још од самог настанка и примене првих класификационих модела примећен је пад њихових перформанси онда када у скупу за обучавање постоје значајне разлике у учесталости категорија. Као што смо видели у претходном одељку, у оваквим ситуацијама били бисмо суочени са проблемом класификације небалансираних података.

Деградирајући перформанс модела огледа се у тежњи да већини нових тачака доделе ону категорију која је приликом њиховог обучавања била најбројнија. Стога би оне тачке које одговарају најмање заступљеним категоријама биле неретко погрешно класификоване.

Из потраге за узроком загонетне пристрасности модела ка најзаступљенијој категорији, коју ћемо краће називати *већинском* категоријом, родило се уверење да су за описано понашање модела одговорне како специфичности скупа за тренирање тако и начин на који се модел обучава. Не само да

мали број инстанци најмање заступљене, односно *мањинске* категорије није довољан да модел научи репрезентацију целокупног скупа тачака којима ова категорија одговара, већ и расподела њихових атрибута у карактеристичним ситуацијама не утиче на модел на жељени начин. Чести примери оваквих сценарија су следећи (слика 1.2):

- *присуство малих кластера мањинске категорије*
  наиме, како већина модела партиционише простор атрибута на скупове тачака који одговарају свакој од категорија, овакви кластери могу бити препознати као аутлајери, што имплицира њихово сврставање у скуп који одговара категорији околних тачака. Стога ће свака тачка мањинске категорије која се налази у близини поменутог кластера бити погрешно класификована;
- *присуство изолованих тачака мањинске категорије*
  слично као у претходном случају, изоловане тачке често добијају статус аутлајера и погрешну категорију;
- *преклапање категорија*
  овај сценарио настаје када се инстанце различитих категорија нађу у заједничком простору. У таквим регијама модели често имају половичну успешност у давању предикција.



Слика 1.2 : Пример небалансираног скупа са 4 категорије.

Са друге стране, уколико се приликом обучавања модела подједнако вреднују информације које потичу од свих категорија, утицај тачака мањинске категорије бива *замаскиран* утицајем преосталих тачака, те се на тај начин додатно отежава њихово препознавање.

Јасно је да уколико пронађемо начин за превазилажење оваквих препрека, можемо значајно унапредити предикциону моћ модела који се обучавају на небалансираном скупу.

Међутим, проблем се утолико компликује када се уз небалансираност јаве и друге непогодности, услед којих нам може понестати решења. Рецимо, модел који се обучава на скупу мале кардиналности често има потешкоћа са учењем репрезентације и других категорија. Још екстремнији сценарио од поменутог јесте онај када је узорак тренажних тачака потпуно нерепрезентативан, јер се о њиховој расподели могу извући погрешни закључци. Дакле, некада ћемо се наћи и у ситуацији када неочекивани резултати потичу од узрочника независних од небалансираности, што нам говори да ћемо на путу до циља бити изложени бројним изазовима.

1.4 Евалуација модела

Следеће питање које се поставља јесте како вредновати модел који је обучаван на небалансираном скупу.

Како желимо да се уверимо да ће модел у будућности бити довољно поуздан у давању прогноза, провера његовог квалитета на скупу за тренирање није од великог значаја. Таквим поступком бисмо добили информацију само о томе колико модел добро разликује тачке на којима је обучаван, али не и колика је његова генерализациона моћ. Стога је идеја да испитивање модела извршимо на скупу нових тачака које модел није видео приликом свог обучавања. Такав скуп тачака ћемо звати скуп за *тестирање* (модела).

Када новим тачкама доделимо категорије предвиђене моделом и упоредимо их са правим категоријама, можемо формирати следећу матрицу.

Дефиниција 1.4.1. *Нека је K број категорија које може имати зависна променљива. Матрица $M = [m_{ij}]$ из $\mathcal{M}_K(\mathbb{R})$ таква да m_{ij} представља број инстанци $i-$те категорије класификованих као j, назива се матрица конфузије.*

Матрица конфузије је један од централних појмова у класификацији јер се из њеног изгледа може много закључити о квалитету модела. Наиме, што је матрица ближа дијагоналној, односно што су вредности m_{ij} за различите i и j мање, то је модел успешније обавио класификацију. Конкретно, успешност модела у препознавању i-те категорије огледа се у што мањим вредностима матрице конфузије у i-тој врсти и колони, са изузетком поља m_{ii}.

Имајући наведено у виду, из M можемо извести разне метрике за евалуацију модела. Пример најшире коришћене евалуационе метрике јесте *тачност*:

$$Accuracy \coloneqq \frac{tr(M)}{\sigma(M)},$$

где је са $tr(\cdot)$ означен траг матрице, а са $\sigma(\cdot)$ збир свих њених елемената. Дакле, ова метрика представља удео тачно класификованих тачака у целом скупу.

Међутим, тачност није пожељно користити када се модел обучава на небалансираном скупу, а нарочито не уколико је такав и скуп за тестирање[4]. Разлог је тај што инстанци мањинске категорије има знатно мање од осталих, те грешка класификације на њима не може значајно допринети смањењу ове метрике. Због тога се може догодити да вредност *Accuracy* буде блиска јединици, а да са друге стране ниједна инстанца мањинске категорије не буде тачно класификована.

Дакле, избор одговарајуће метрике умногоме ће зависити од саме природе проблема. Адекватна метрика биће она која може установити ону грешку која је у посматраној проблематици од највеће важности. У присуству небалансираности података, такве метрике би биле оне које дају на значају грешкама класификације мањинске категорије. Да бисмо навели примере истих, уведимо најпре следеће ознаке.

Нека је i мањинска категорија. Назовимо је за тренутак позитивном[5] категоријом, а све остале категорије негативним. Тада је:

- TP_i – број инстанци i-те категорије класификованих као i.
  Важи:
  $$TP_i = m_{ii}.$$

- FP_i – број инстанци негативних категорија класификованих као i.
  Важи:
  $$FP_i = \sum_{\substack{j=1 \\ j \neq i}}^{K} m_{ji}.$$

[4] Како узорак тренажних и тестних тачака обично долази из исте расподеле, природно је очекивати да ће скуп за тестирање имати сличне карактеристике као и скуп за тренирање.

[5] Термини позитивна и негативна категорија настали су када се у бинарној класификацији усталио запис категорија као 1 и -1 или 1 и 0.

- FN_i – број инстанци i-те категорије које нису класификоване као i. Важи:
$$FN_i = \sum_{\substack{j=1 \\ j \neq i}}^{K} m_{ij}.$$

- TN_i – број инстанци негативних категорија које нису класификоване као i. Важи:
$$TN_i = \sigma(M) - TP_i - FP_i - FN_i.$$

Уколико је јасно на коју категорију се односе наведене ознаке, надаље ћемо због једноставности записа изостављати њен индекс. Када је реч о бинарној класификацији, матрица конфузије се може представити на следећи начин (слика 1.3):

		Prediction	
		Positive	**Negative**
Actual	**Positive**	**TP**	**FN**
	Negative	**FP**	**TN**

Слика 1.3 : Матрица конфузије у бинарној класификацији.

На основу досадашњег излагања закључујемо да је важно обратити посебну пажњу на тачност класификације међу инстанцама позитивне категорије, коју ћемо звати *одзив* (енг. *Recall*, *Sensitivity*) и означавати са TPR:
$$TPR \coloneqq \frac{TP}{TP + FN}.$$

Ништа мање важна је и метрика која показује тачност класиикације међу инстанцама које су класификоване као i, познатија као *прецизност* (енг. *Precision*) и означавати са PPV:
$$PPV \coloneqq \frac{TP}{FP + TP}.$$

У бинарној класификацији можемо аналогно посматрати и тачност међу инстанцама негативне категорије (енг. $Specificity$):

$$TNR \coloneqq \frac{TN}{TN + FP}.$$

Како за $K > 2$ ова метрика не говори много о класификацији преосталих категорија понаособ, ова метрика се у тим ситуацијама ређе користи.

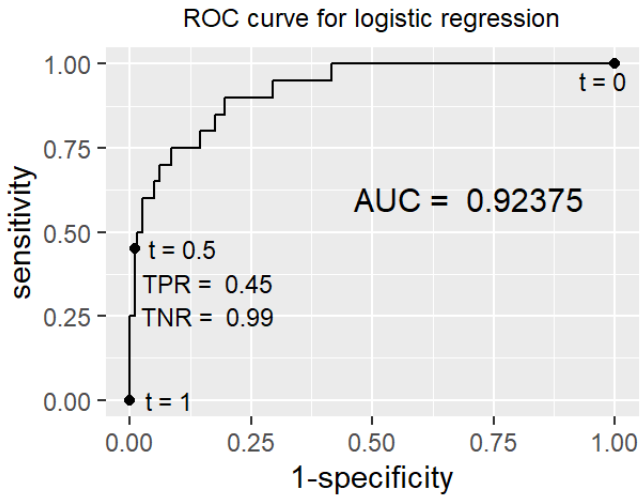
Јасно је да бисмо желели да све до сад изложене метрике буду што ближе јединици, односно да FP и FN буду што мањи. С тим циљем можемо дефинисати и метрике које настају комбинацијом других метрика, као што је F *мера*:

$$F \coloneqq \frac{(1 + \beta^2)(PPV \cdot TPR)}{\beta^2 PPV + TPR}.$$

Подешавањем параметра β^2 можемо утицати на пристрасност F мере ка PPV, односно TPR. За $\beta^2 = 1$ F је исто што и хармонијска средина PPV и TPR и придаје подједнаку важност истима. Када је β^2 блиско 0, $F \approx PPV$, док за велике вредности β^2 важи $F \approx TPR$.

За крај овог поглавља представићемо и визуелни метод евалуације коришћен у бинарној класификацији, који показује како се с променом класификационог правила мења и тачност класификације међу обема категоријама. Овај метод је заснован на графичком приказу парова тачака $(1 - TNR(t),\ TPR(t))$ за све могуће вредности поменутог параметра t, који се назива ROC крива (енг. *receiver operating characteristic curve*).

Почетак и крај криве су, редом, у тачкама $(0, 0)$ и $(1, 1)$, што одговара избору оног t за које се све тачке класификују као негативна, односно као позитивна категорија. Даље, претпоставимо без умањења општости да с опадањем t расте број инстанци класификованих позитивно. Жељени сценарио је онај у коме $TPR(t)$ све више расте, а $TNR(t)$ не опада пребрзо, чему одговара конкавни облик криве, са што већим 'испупчењем' ка тачки $(0, 1)$. Стога је корисно дефинисати метрику која ће евалуирати модел кроз облик ROC криве, а то је управо метрика AUC, која представља површину испод графика криве. Јасно је да је модел утолико бољи што је вредност AUC ближа јединици. Међутим, AUC не говори о перформансу модела при класификацији за фиксно t, због чега може бити непоуздан критеријум евалуације (слика 1.4).



Слика 1.4 : *ROC* крива за модел логистичке регресије обучаван на небалансираном скупу (однос категорија је 10:1).

Поглавље 2

Модели осетљиви на небалансираност података

У одељку 1.3 говорили смо о могућим узроцима проблема небалансираности, са напоменом да се до изнетих закључака дошло емпиријским путем. Сада је тренутак да причу формализујемо на примерима конкретних модела, где ћемо се најпре упознати са архитектуром модела, а потом на основу исте оправдати зашто би наш проблем могао настати.

2.1 Логистичка регресија

Иако је у општем случају тешко минимизовати стварни ризик из раније поменутих разлога, постоје ситуације када се облик решења оптимизационог проблема $\min_f R(f)$ може донекле прецизирати. Узмимо за пример следећу функцију грешке:

$$L(f(\boldsymbol{X}), Y) \coloneqq (Y - f(\boldsymbol{X}))^2.$$

Теорема 2.1.1. *За сваку функцију f за коју постоји коначно очекивање на десној страни неједнакости, важи:*

$$E(Y - E(Y|\boldsymbol{X}))^2 \leq E(Y - f(\boldsymbol{X}))^2.$$

Доказ:

$$E(Y - f(\boldsymbol{X}))^2 = E(Y - E(Y|\boldsymbol{X}) + E(Y|\boldsymbol{X}) - f(\boldsymbol{X}))^2$$

$$= E(Y-E(Y|\boldsymbol{X}))^2+2E\big[(Y-E(Y|\boldsymbol{X}))\cdot(E(Y|\boldsymbol{X})-f(\boldsymbol{X})\big]+E(E(Y|\boldsymbol{X})-f(\boldsymbol{X}))^2.$$

Посматрајмо средњи члан претходног израза:

$$E\big[(Y-E(Y|\boldsymbol{X}))\cdot(E(Y|\boldsymbol{X})-f(\boldsymbol{X})\big] = E(E\big[(Y-E(Y|\boldsymbol{X}))\cdot(E(Y|\boldsymbol{X})-f(\boldsymbol{X})\big]|X)$$

$$= E((E(Y|\boldsymbol{X}) - f(\boldsymbol{X})) \cdot E(Y - E(Y|\boldsymbol{X})|\boldsymbol{X})).$$

Како је

$$E(Y - E(Y|\boldsymbol{X})|X) = E(Y|\boldsymbol{X}) - E(E(Y|\boldsymbol{X})|\boldsymbol{X}) = E(Y|\boldsymbol{X}) - E(Y|\boldsymbol{X}) = 0,$$

следи

$$E(Y - f(\boldsymbol{X}))^2 = E(Y - E(Y|\boldsymbol{X}))^2 + E(E(Y|\boldsymbol{X}) - f(\boldsymbol{X}))^2 \geq E(Y - E(Y|\boldsymbol{X}))^2.$$

■

Дакле, модел који минимизује ризик задат овом функцијом грешке јесте условно очкивање $E(Y|\boldsymbol{X})$ и назива се *регресиона функција*. Уколико би Y била бинарна категоричка променљива, чије бисмо категорије означили са 0 и 1, модел би гласио

$$E(Y|\boldsymbol{X}) = \mathcal{P}\{Y = 1|\boldsymbol{X}\}.$$

Модели којима се моделује вероватноћа $\pi(\boldsymbol{X}) \coloneqq \mathcal{P}\{Y = 1|\boldsymbol{X}\}$ чине класу тзв. *пробабилистичких дискриминативних* модела. Пример таквог модела на који ћемо се фокусирати у овом раду јесте логистички регресиони модел.

Дефиниција 2.1.1. *Модел π задат са*

$$\pi(\boldsymbol{X}) \coloneqq \frac{e^{\beta_0+\boldsymbol{\beta}^T\boldsymbol{X}}}{1 + e^{\beta_0+\boldsymbol{\beta}^T\boldsymbol{X}}}$$

назива се (бинарни) логистички регресиони модел.

Оцене параметара $\beta_0 \in \mathbb{R}$ и $\boldsymbol{\beta} \in \mathbb{R}^p$ стандардно се добијају методом максималне веродостојности.

Нека се скуп за тренирање састоји из парова тачака $(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n)$. Ако претпоставимо независност условних расподела $Y_i|\boldsymbol{X}_i$, функција веродостојности гласи

$$L(\beta_0, \boldsymbol{\beta}) = \prod_{i=1}^{n} \pi(\boldsymbol{x}_i)^{y_i} \cdot (1 - \pi(\boldsymbol{x}_i))^{1-y_i},$$

док је њен логаритам једнак

$$\begin{aligned} l(\beta_0, \boldsymbol{\beta}) &= \log(L(\beta_0, \boldsymbol{\beta})) \\ &= \sum_{i=1}^{n} \left[y_i \log \pi(\boldsymbol{x}_i) + (1 - y_i) \log(1 - \pi(\boldsymbol{x}_i)) \right] \\ &= \sum_{i=1}^{n} \left[y_i (\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i - \log(1 + e^{\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i})) - (1 - y_i) \log(1 + e^{\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i}) \right] \\ &= \sum_{i=1}^{n} y_i (\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i) - \sum_{i=1}^{n} \log(1 + e^{\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i}). \end{aligned}$$

Дакле, оцене параметара логистичког модела гласе:

$$(\widehat{\beta_0}, \widehat{\boldsymbol{\beta}}) = \arg\max_{\beta_0, \boldsymbol{\beta}} l(\beta_0, \boldsymbol{\beta}).$$

Проблемом егзистенције и јединствености решења овог оптимизационог проблема детаљно се бавио M. J. Силвапул, о чему се више може видети у [3].

Након креираног модела π можемо извршити класификацију нове тачке $\boldsymbol{x}$ у зависности од следећег правила:

$$r(\boldsymbol{x}) = 1 \iff \pi(\boldsymbol{x}) > t, \ t \in (0, 1)$$

За вредност прага t најчешће се узима $\frac{1}{2}$, мада су могуће и варијације.

Кључна претпоставка модела логистичке регресије јесте линеарна раздвојивост категорија, односно постојање хиперравни која раздваја тачке које не припадају истој категорији. Наиме, како је

$$\pi(\boldsymbol{x}) > t \iff \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x} > \log\left(\frac{t}{1-t}\right),$$

класификација тачака зависи од њиховог положаја у односу на хиперраван $\alpha: \ \beta_0 - \log\left(\frac{t}{1-t}\right) + \boldsymbol{\beta}^T \boldsymbol{x} = 0$.

Што расподела по категоријама више одступа од овакве поставке, а управо такве су ситуације наведене у одељку 1.3, квалитет логистичке регресије постаје упитан.

2.1.1 Вишекласна класификација

До сада смо анализирали проблематичност логистичке регресије искључиво у случају бинарне класификације. Како се не бисмо ограничили искључиво на случај када је $K = 2$, описаћемо како се на сличан начин може дефинисати и *мултиномни* логистички регресиони модел, који се користи када зависна променљива узима више од 2 вредности.
Како је вероватноћа припадности тачке $\boldsymbol{x}$ негативној категорији у случају бинарне класификације једнозначно одређена са $1 - \pi(\boldsymbol{x})$, није тешко показати да важи

$$\log \frac{\mathcal{P}\{Y = 1|\boldsymbol{X}\}}{\mathcal{P}\{Y = 0|\boldsymbol{X}\}} = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{X}.$$

Дата релација нам говори да се логистичком регресијом успоставља линеарна веза између независне променљиве и логаритма *квота*, где квота представља количник вероватноћа два дисјунктна догађаја. На сличан начин вршимо уопштење логистичке регресије у случају вишекласне класификације.

Дефиниција 2.1.2. *Нека је $K > 2$ и нека је C једна од могућих категорија из скупа $\{1, \ldots, K\}$. Модел који описује вероватноћу $\mathcal{P}\{Y = k|\boldsymbol{X}\}$ тако да важи*

$$\log \frac{\mathcal{P}\{Y = k|\boldsymbol{X}\}}{\mathcal{P}\{Y = C|\boldsymbol{X}\}} = \beta_0^{(k)} + \boldsymbol{\beta}^{(k)T} \boldsymbol{X}, \quad \forall k \in \{1, \ldots, K\} \setminus C,$$

назива се (мултиномни) логистички регресиони модел.
Категорија C назива се референтна категорија.

Како се вероватноће припадности свакој од категорија морају сумирати у 1, изводимо следеће релације:

$$\mathcal{P}\{Y = C|\boldsymbol{X}\} = \frac{1}{1 + \sum_{\substack{j=1 \\ j \neq C}}^{K} e^{\beta_0^{(j)} + \boldsymbol{\beta}^{(j)T} \boldsymbol{X}}},$$

$$\mathcal{P}\{Y = k|\boldsymbol{X}\} = \frac{e^{\beta_0^{(k)} + \boldsymbol{\beta}^{(k)T} \boldsymbol{X}}}{1 + \sum_{\substack{j=1 \\ j \neq C}}^{K} e^{\beta_0^{(j)} + \boldsymbol{\beta}^{(j)T} \boldsymbol{X}}}.$$

Параметри мултиномне логистичке регресије се такође могу добити методом максималне веродостојности, а класификација нове тачке $\boldsymbol{x}$ одвија

се по принципу највеће оцењене вероватноће:

$$r(\boldsymbol{x}) = k \iff \mathcal{P}\{Y = k|\boldsymbol{x}\} = \max\{\mathcal{P}\{Y = j|\boldsymbol{x}\},\ j \in \{1, \ldots, K\}\}.$$

Дакле, мултиномна логистичка регресија такође партиционише простор атрибута на линеарно раздвојиве подскупове. Рецимо, скупу тачака класификованих као C одговара скуп $\{\boldsymbol{x} \ :\ \beta_0^{(j)} + \boldsymbol{\beta}^{(j)T}\boldsymbol{x} < 0, \quad \forall j \neq C\}$, док је скуп тачака класификованих као k дат са

$$\{\boldsymbol{x} \ :\ \beta_0^{(k)} + \boldsymbol{\beta}^{(k)T}\boldsymbol{x} > 0, \quad \beta_0^{(k)} - \beta_0^{(j)} + (\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^{(j)})^T\boldsymbol{x} > 0, \quad \forall j \neq k\}.$$

Стога као и до сада можемо наслутити да ће препознатљивост мањинске категорије бити утолико лошија што су израженија преклапања са тачкама других категорија.

2.2 Модел потпорних вектора

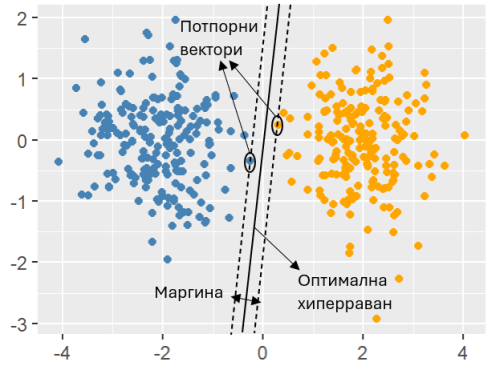
У овом одељку представићемо још један модел који почива на претпоставци о линеарној раздвојивости категорија, али који применом напредних техника може бити прилагођен и другачијим сценаријима. Реч је о моделу потпорних вектора, који ћемо краће ословљавати са SVM[1].

За разлику од модела логистичке регресије, SVM је алгебарске природе и не сврстава се у пробабилистичке моделе. Тачније, овај модел припада класи модела заснованих на широком појасу, који су дизајнирани са циљем постављања најпоузданијих граница међу категоријама.

Као и у претходном примеру описаћемо архитектуру модела када је пред нама задатак бинарне класификације, а потом извршити и уопштење на вишекласни случај.
Нека су $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ тачке тренажног скупа и нека је са y_i означена категорија $i-$те тачке, при чему ће негативна категорија бити означена са -1, а позитивна са 1. Премда димензија независних променљивих може бити произвољна, ради лакше интерпретације дешавања која следе, све илустрације даћемо у простору мале димензије.

Како претпостављамо линеарну раздвојивост категорија, наш тренутни циљ је да категорије раздвојимо хиперравни која је на највећој могућој удаљености од њој најближе тачке обеју категорија.

[1] енг. *support vector machine*



Слика 2.1 : Скица линеарно раздвојивог скупа.

Поменуте тачке називају се *потпорним векторима*, а растојање између хиперравни одређене њима, а које су паралелне оптималној хиперравни, назива се *маргина* (слика 2.1).

Нека је оптимална хиперраван дата једначином $\omega^T \boldsymbol{x} + b = 0$. Тада су њој паралелне хиперравни које пролазе кроз потпорне векторе позитивне и негативне категорије, редом, дате са $\omega^T \boldsymbol{x} + b = c$ и $\omega^T \boldsymbol{x} + b = -c$. Зарад рачунских олакшица подразумеваћемо да је $c = 1$. При оваквој конфигурацији испуњено је:

- за све тачке ван маргине важи $\omega^T \boldsymbol{x} + b > 1$ (позитивна категорија) или $\omega^T \boldsymbol{x} + b < -1$ (негативна категорија); $\omega^T \boldsymbol{x} + b = \pm 1$ ако и само ако је $\boldsymbol{x}$ потпорни вектор;
- за сваку тачку из скупа за тренирање и њену категорију важи $y_i(\omega^T \boldsymbol{x}_i + b) \geq 1$.

Уз наведене ознаке можемо и формално дефинисати посматрани модел.

Дефиниција 2.2.1. *Модел $f_{\omega,b}$ дефинисан са*

$$f_{\omega,b}(\boldsymbol{X}) = \omega^T \boldsymbol{X} + b$$

назива се модел потпорних вектора.

Такође, видимо и да се иза поменуте дефиниције оптималне хиперравни крије минимизација емпиријског ризика задатог функцијом грешке:

$$L(u, v) = \max(0, 1 - uv).$$

Наиме, за све вредности $\boldsymbol{X}$ ван маргине и у делу простора одговарајуће категорије грешка је једнака 0. Слично, за вредности $\boldsymbol{X}$ у делу простора одговарајуће категорије, али унутар маргине, грешка је у распону $(0, 1)$, док је за сваку вредност $\boldsymbol{X}$ у делу простора супротне категорије грешка већа од 1. Иако постоји више хиперравни за које је емпиријски ризик једнак 0, одлучили смо се за ону са најширом маргином, ослањајући се на претпоставку да је тренажни узорак репрезентативан, те да смо тако разграничили категорије на најпоузданији начин.

Одредимо сада ширину маргине коју желимо максимизовати. Нека је $\boldsymbol{p}$ произвољна тачка за коју важи $\omega^T \boldsymbol{p} + b = -1$. Њој најближа тачка $\boldsymbol{q}$ таква да је $\omega^T \boldsymbol{q} + b = 1$ је облика $\boldsymbol{q} = \boldsymbol{p} + \lambda\omega$, па је растојање између $\boldsymbol{p}$ и $\boldsymbol{q}$ једнако $|\lambda|\|\omega\|$. Уз мало рачуна добијамо да је $\omega^T \boldsymbol{q} + b = \omega^T \boldsymbol{p} + \lambda\|\omega\|^2 + b$, односно $1 = \lambda\|\omega\|^2 - 1$, из чега следи $\lambda\|\omega\| = \frac{2}{\|\omega\|}$. Дакле, параметри b и ω који одређују оптималну хиперраван добијају се као решења следећег оптимизационог проблема:

$$\min_{\omega, b} \frac{\|\omega\|}{2},$$

при услову $y_i(\omega^T \boldsymbol{x}_i + b) \geq 1, \quad \forall i \in \{1, \ldots, n\}$.
Додатни услов обезбеђује одсуство тренажних тачака унутар маргине, а разлог због ког уместо максимизације радимо минимизацију реципрочне вредности је тај што су многи оптимизациони алгоритми конструисани управо за проблеме минимизације.

Дакле, класификација нове тачке $\boldsymbol{x}$ гласи:

$$r(\boldsymbol{x}) = \text{sign}(\omega^T \boldsymbol{x} + b).$$

2.2.1 Регуларизација

Поред чињенице да се поменута хиперраван не може увек конструисати, још један недостатак овог метода јесте велика осетљивост на аутлајере. Наиме, поједине тачке могу драстично утицати на положај хиперравни и тако раздвојити категорије на нерепрезентативан начин. Ова појава је позната под називом *преприлагођавање* подацима из скупа за тренирање.

Да бисмо надоместили овај недостатак, допустићемо неким тачкама да се нађу са погрешне стране хиперравни одређене потпорним векторима своје класе. У ту сврху увешћемо нову променљиву $\xi_i \geq 0,\ i \in \{1, \ldots n\}$ која ће

контролисати одступање тачке $\boldsymbol{x}_i$ од поменуте хиперравни. Да бисмо одредили и оптимално $\xi = (\xi_i, \ldots, \xi_n)$, ослонићемо се на технику регуларизације. Под регуларизацијом подразумевамо модификацију оптимизационог проблема додавањем новог *регуларизационог* параметра, који има улогу у ограничавању грешака које допуштамо. У нашем случају то ће бити параметар $C > 0$, а нови оптимизациони проблем ће гласити:

$$\min_{\omega,b,\xi} \frac{\|\omega\|^2}{2} + C\sum_{i=1}^{n} \xi_i, \tag{2.1}$$

при услову $y_i(\omega^T \boldsymbol{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \quad \forall i \in \{1, \ldots, n\}$.
Дакле, ако је $\xi_i = 0$, тачка $\boldsymbol{x}_i$ се налази у одговарајућем делу простора, а што је ξ_i веће, то се $\boldsymbol{x}_i$ дозвољава да прави веће одступање. За $\xi_i > 1$ се чак може догодити да се $\boldsymbol{x}_i$ нађе у делу простора коме одговара супротна категорија од y_i. Напоменимо да је норма вектора ω квадрирана ради једноставније оптимизације која ће ускоро уследити.

Вредности параметра C блиске 0 немају велики утицај на ξ_i, те су у том случају могуће велике грешке, а што је C веће, то ће сума $\sum_{i=1}^{n} \xi_i$ у претходном изразу бити доминатнија, због чега ће оптимални ξ_i бити све ближи 0. Видимо да други случај заправо одговара првобитној идеји у којој нису биле допустиве грешке, односно у којој смо се потпуно били прилагодили подацима. Како бисмо проценили жељени праг толеранције грешака, предложено је да се вредност параметра C одреди унакрсном валидацијом, која се одвија у следећим корацима:

1. дефинисање опсега могућих вредности параметра C;
2. подела тренажног скупа на K подскупова;
3. за свако C дефинисано у првом кораку:
   - 3.1. за сваки од K подскупова дефинисаних у другом кораку:
     - 3.1.1. оптимизација параметара модела за изабрано C на основу тачака из уније преосталих $K - 1$ подскупова;
     - 3.1.2. класификација тачака из издвојеног подскупа на основу креираног модела;
   - 3.2. евалуација модела на тренажном скупу на основу предикција добијених у кораку 3.1;
4. одабир оног C за које су добијени најбољи резултати у кораку 3.2.

Запишимо сада лагранжијан који одговара нашем оптимизационом проблему:

$$\mathcal{L}(\omega, b, \xi, \alpha, \beta) = \frac{\|\omega\|^2}{2} + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - \xi_i - y_i(\omega^T\boldsymbol{x}_i + b)) + \sum_{i=1}^{n}\beta_i\xi_i,$$

где су $\alpha_i \geq 0, \beta_i \geq 0, \quad \forall i \in \{1, \ldots, n\}$.
Сада њему еквивалентан оптимизациони проблем гласи:

$$\min_{\omega,b,\xi}\ \max_{\alpha_i\geq 0,\ \beta_i\geq 0} \mathcal{L}(\omega, b, \xi, \alpha, \beta).$$

Захваљујући техникама математичке оптимизације може се показати да функција $\mathcal{L}$ задовољава услове услед којих је могућа замена редоследа минумума и максимума, те да нова формулација оптимизације гласи:

$$\max_{\alpha_i\geq 0,\ \beta_i\geq 0}\ \min_{\omega,b,\xi} \mathcal{L}(\omega, b, \xi, \alpha, \beta).$$

За више детаља о истом консултовати [4].

Даље, за фиксне α и β тачка минимума функције $\mathcal{L}$ задовољава $\frac{\partial\mathcal{L}}{\partial\omega} = 0$, $\frac{\partial\mathcal{L}}{\partial b} = 0$ и $\frac{\partial\mathcal{L}}{\partial\xi} = 0$, што је еквивалентно са

$$\omega - \sum_{i=1}^{n}\alpha_i y_i\boldsymbol{x}_i = 0, \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad C + \beta_i - \alpha_i = 0,\ \forall i \in \{1, \ldots, n\},$$

те експлицитан израз оптималног параметра ω тражене хиперравни гласи:

$$\widehat{\omega} = \sum_{i=1}^{n}\alpha_i y_i\boldsymbol{x}_i,$$

а оптимизациони проблем постаје:

$$\max_{\alpha_i\geq 0}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\boldsymbol{x}_i^T\boldsymbol{x}_j,$$

при услову $\sum_{i=1}^{n}\alpha_i y_i = 0, \quad \forall i \in \{1, \ldots, n\}$.

Коначно, када после извршене оптимизације добијемо $\widehat{\alpha}$ и $\widehat{b}$, спремни смо да извршимо класификацију тачке $\boldsymbol{x}$:

$$r(\boldsymbol{x}) = \operatorname{sign}\Big(\sum_{i=1}^{n}\widehat{\alpha}_i y_i\boldsymbol{x}_i^T\boldsymbol{x} + \widehat{b}\Big).$$

2.2.2 Кернелизовани модел потпорних вектора

Иако смо регуларизацијом спречили настанак преприлагођавања, проблем који представља потенцијална линеарна нераздвојивост и даље остаје нерешен. Премда је регуларизацијом могуће ублажити ефекат линеарне нераздвојивости у случају када до ње долази због свега пар тренажних тачака, у пракси су чешће знатно компликованије ситуације. Како моћ линеарних класификатора не би тада остала неискоришћена, дошло се на идеју да се простор независних променљивих преслика у простор у коме ће променљиве бити линеарно раздвојиве и на тако пресликаним тачкама извршити обучавање модела.

Уколико са ϕ означимо поменуто пресликавање, класификација тачке $\boldsymbol{x}$ дата је са:

$$r(\boldsymbol{x}) = \operatorname{sign}\Big(\sum_{i=1}^{n} \widehat{\alpha}_i y_i \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}) + \widehat{b}\Big).$$

Међутим, одговарајуће пресликавање је генерално тешко наћи. Искуство је показало да у карактеристичним ситуацијама поједина пресликавања могу бити корисна, али је само препознавање тих ситуација у простору велике димензије готово немогуће. Да бисмо то постигли, искористићемо резултат следеће теореме.

Теорема 2.2.1. *Нека је $\mathcal{X}$ непразан скуп. За сваку функцију $k : \mathcal{X}\times\mathcal{X} \mapsto \mathbb{R}$ која је симетрична по својим аргументима и која је таква да је за свако $n \in \mathbb{N}$ и произвољне $x_1, \ldots, x_n \in \mathcal{X}$ матрица са елементима $k(x_i, x_j)$ семидефинитна, постоји пресликавање ϕ_k из $\mathcal{X}$ у простор са скаларним производом такво да важи*

$$k(x, y) = \phi_k(x)^T \phi_k(y), \quad \forall x, y \in \mathcal{X}.$$

Функција k која задовољава услове из теореме, назива се (Мерцеров) кернел или језгро.

Стога је корисно адекватно пресликавање независних променљивих потражити међу пресликавањима индукованим поменутим кернелима. Примери често коришћених кернела дати су у табели 2.1.

линеарни	$k(x,y) = x^T y$
полиномни	$k(x,y) = (\gamma x^T y + c)^d$
радијални	$k(x,y) = e^{(-\gamma\|x-y\|)}$
сигмоидни	$k(x,y) = \tanh(\gamma x^T y + c)$

Табела 2.1

Уколико кернел зависи од додатних параметара, њихове вредности се такође могу изабрати унакрсном валидацијом.

Када говоримо о перформансама модела обучаваног на небалансираним подацима, SVM спада у моделе умерено осетљиве на небалансираност. Један од могућих разлога слабијег препознавања мањинске категорије је већ поменута нерепрезентативност њених тачака у скупу за тренирање.
Ву и Ченг су у [5] показали да небалансираност тренажних тачака повлачи и небалансираност потпорних вектора, те да се у околини граничних тачака налази више потпорних вектора већинске него мањинске категорије. Исти аутори су претпоставили да стога граничне тачке потпадају под утицај већинске категорије и бивају класификоване на исти начин.
Међутим, примећено је да у случају умерене небалансираности SVM показује извесну робусност. Верује се да томе делом доприносе веће[2] вредности параметара $\widehat{\alpha}_i$, за оне i који одговарају мањинској категорији, чиме врше утицај на класификационо правило у корист мањинске категорије.

2.2.3 Вишекласна класификација

Класификација применом модела који су попут SVM спецификовани за проблеме бинарне класификације може се проширити на вишекласни случај на два начина.

Први се састоји у креирању граница између категорија добијених на основу $\frac{K(K-1)}{2}$ модела који се обучавају на основу свих могућих парова категорија. Друга опција јесте раздвајање категорија применом K модела који за сваку од K категорија посматрају унију преосталих категорија као њој супротну.

Како је у другом случају за $K > 3$ потребно обучити мање модела, исти приступ је рачунски једноставнији. Међутим, чињеница да унирањем $K-1$

[2] показали смо да за $\widehat{\alpha}_i$ важи $\sum_{i=1}^{n} \widehat{\alpha}_i y_i = 0$, што имплицира да они параметри који одговарају мањинској категорији имају већу вредност

категорија долази до изражене небалансираности, а полазни модел показује знаке лошијег перформанса у таквим околностима, чини овај приступ изузетно непоузданим.

2.3 Модел k најближих суседа

Модел k најближих суседа, краће само kNN[3], један је од најинтуитивнијих модела машинског учења. Заснива се на претпоставци да тачке сличних вредности атрибута с великом вероватноћом припадају истој категорији, при чему се сличност две тачке може измерити одговарајућом метриком d.

За разлику од претходних модела с којима смо се упознали, kNN се равноправно користи како при бинарној тако и при вишекласној класификацији. Формално, овај модел за унапред задато k моделује вероватноћу припадности тачке $\boldsymbol{x}$ свакој од $K \in \mathbb{N}$ категорија на основу њених k најближих суседа.

Дефиниција 2.3.1. *Нека се скуп за обучавање састоји из n тачака $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ и нека је са $y_i \in \{1, \ldots, K\}$ означена категорија i-те тачке. Означимо са $B_k(\boldsymbol{x})$ скуп k најближих тренажних тачака тачки $\boldsymbol{x}$. Модел дефинисан са*

$$\mathcal{P}\{Y = j|\boldsymbol{X}\} = \frac{\sum_{\boldsymbol{x}_i \in B_k(\boldsymbol{X})} I\{y_i = j\}}{k}, \quad \forall j \in \{1, \ldots, K\}$$

назива се модел k најближих суседа.

Класификација нових тачака обично се одвија по принципу највеће оцењене вероватноће, односно:

$$r(\boldsymbol{x}) = \underset{j \in \{1,\ldots,K\}}{\arg\max} \mathcal{P}\{Y = j|\boldsymbol{x}\}. \tag{2.2}$$

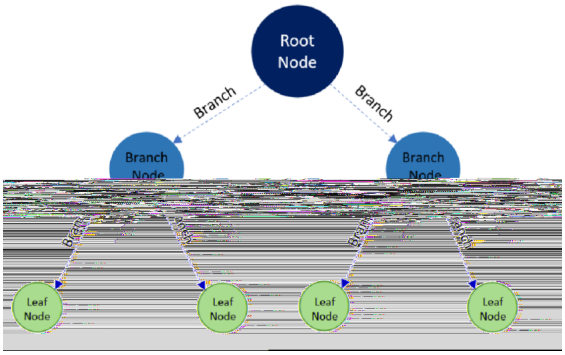
Дакле, обучавање модела kNN своди се на проналажење одговарајуће метрике d и оптималног $k \in \mathbb{N}$.
Уколико се модел обучава на небалансираном скупу, превелике вредности k узрокују додељивање већинске категорије великом броју тачака, што повећава вероватноћу непрепознавања мањинске категорије, нарочито у њеним изолованим кластерима. Са друге стране, премало k отежава моделу

[3] енг. *k nearest neighbors*

да се прилагоди подацима и научи њихову репрезентацију, те лако може доћи до такозваног *потприлагођавања*.

2.4 Стабла одлучивања

За крај овог поглавља дајемо још један пример врло интуитивног и интерпретабилног модела, чија је употреба виђена и ван оквира машинског учења, нарочито у дисциплинама које се ослањају на оптимално доношење одлука. Реч је о стаблима одлучивања[4], моделу представљеним хијерархијском структуром дрвета. Стабла се састоје из једног улазног чвора, односно *корена* стабла, који се грана на један или више чворова, које називамо његовим потомцима. Сваки од тих чворова може се даље гранати на исти начин или садржати резултат, односну одлуку донету овим моделом. Резултујуће чворове стабала називаћемо *листовима* (слика 2.2).



Слика 2.2 : Структура стабла одлучивања.

Кретање кроз стабло одвија се од корена према листовима, при чему се у сваком чвору врши тест којим се одређује даља путања. Сваком исходу теста одговара тачно једна грана која полази из посматраног чвора.

Када је реч о проблемима надгледаног учења, стабла одлучивања се равноправно користе како при регресији тако и при класификацији. Због природе задате теме, у даљем излагању специфичности стабала подразумеваћемо да се иста користе за задатак класификације, што ће рећи да се у њиховим листовима налазе категорије предвиђене датим моделом.

[4] енг. *decision trees*

Једна од главних карактеристика стабала одлучивања јесте постојање великог броја алгоритама на основу којих се она могу изградити. Варијације се махом тичу метрика коришћених при одабиру оптималног критеријума по коме се грана текући чвор. Премда су могућа ситна одступања, псеудокод алгоритма израде модела је следећи [6]:

Алгоритам 1 : Креирање стабла одлучивања (S, A, y)

Улаз: S - скуп за тренирање, A - вектор атрибута, y - вектор циљних променљивих инстанци из S

Излаз: T - стабло одлучивања

$T \leftarrow$ стабло које се састоји из једног чвора - корена стабла;

if Постигнут је један од критеријума заустављања израде стабла:
  Корен стабла T постаје лист са најучесталијом категоријом у S;

else
  Пронаћи дискретно пресликавање f скупа вредности A у скупу S, на основу чијих је слика постигнуто оптимално партиционисање скупа S на $S_1, \ldots, S_m$;
  if Партиционисање скупа S на основу f је адекватно:
    for $i \in \{1, \ldots, m\}$
      $S_i \leftarrow i$ -ти елемент партиције;
      $y_i \leftarrow$ вектор циљних променљивих инстанци из S_i;
      $Subtree_i \leftarrow$ Креирање стабла одлучивања (S_i, A, y_i);
      $Subtree_i$ постаје подстабло стабла T;
    end
  else
    Корен стабла T постаје лист са најучесталијом категоријом у S;
  end
end

Иако једноставна за разумевање, стабла одлучивања су веома склона преприлагођавању. Преприлагођавање је утолико вероватније што је стабло дубље, те се ова појава може спречити пооштравањем критеријума заустављања израде стабла како би се добила стабла једноставније структуре. Међутим, на овај начин се модел лако може потприлагодити, што је довело до настанка технике *орезивања* стабала (енг. *pruning*). Орезивање подразумева уклањање подстабала која настају из појединих грана, а која испуњавају одређене критеријуме. Може се спровести по добијању коначног модела (енг. *post-pruning*) или инкорпорирати у сам процес његове изградње (енг. *pre-pruning*).

Детаљнију анализу горњег псеудокода спровешћемо кроз два примера

најкоришћенијих алгоритама за изградњу стабала одлучивања, *CART* и *C*4.5 алгоритам.

2.4.1 Алгоритам *CART*

Алгоритам *CART* (енг. *classification and regression trees*) је први пут описан 1984. године од стране Л. Брајмана и других аутора [7].
CART представља алгоритам израде бинарних стабала одлучивања, што подразумева да се сваки чвор стабла може гранати на тачно два чвора.

При одабиру оптималног критеријума по коме ће се чвор гранати, односно при проналажењу пресликавања f поменутог у алгоритму 1, алгоритам *CART* узима у обзир меру нечистоће узорка дату следећом дефиницијом.

Дефиниција 2.4.1. *Нека је* $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ *и нека је* $y_i \in \{1, \ldots, K\}$ *вредност категоричке променљиве* Y *тачке* $\boldsymbol{x}_i$. *Џинијев индекс је мера нечистоће скупа* S *у односу на променљиву* Y, *дефинисана са*

$$Gini = 1 - \sum_{i=1}^{K} p_i^2,$$

где је p_i *удео тачака* i*-те категорије у скупу* S.

Да бисмо боље разумели мотивацију која се крије иза дефиниције Џинијевог индекса, доказаћемо следећу лему.

Лема 2.4.1.

а) $\min\limits_{p_1,\ldots,p_K} \left(1 - \sum\limits_{i=1}^{K} p_i^2\right)$ *при условима* $\sum\limits_{i=1}^{K} p_i = 1,\ p_i \geq 0,\ \forall i \in \{1, \ldots, K\},$

се достиже за $p_i = 1,\ p_j = 0,\ j \neq i;$

б) $\max\limits_{p_1,\ldots,p_K} \left(1 - \sum\limits_{i=1}^{K} p_i^2\right)$ *при условима* $\sum\limits_{i=1}^{K} p_i = 1,\ p_i \geq 0,\ \forall i \in \{1, \ldots, K\},$

се достиже за $p_1 = \cdots = p_K = \dfrac{1}{K}.$

Доказ: *а)* Како је

$$1 = \sum_{i=1}^{K} p_i = \Big(\sum_{i=1}^{K} p_i\Big)^2 = \sum_{i=1}^{K} p_i^2 + 2\sum_{i<j} p_i p_j \geq \sum_{i=1}^{K} p_i^2,$$

следи да је при датим условима увек испуњено $1 - \sum_{i=1}^{K} p_i^2 \geq 0$.
Једнакост се достиже акко је за неко i испуњено $p_i = 1$ и $p_j = 0,\ j \neq i$.

б) Нека је g функција дефинисана са

$$g(p_1, \ldots, p_K) = 1 - \sum_{i=1}^{K} p_i^2, \text{ за } p_i > 0, \forall i \in \{1, \ldots, K\}. \tag{2.3}$$

Посматрајмо следећи оптимизациони проблем

$$\max_{p_1,\ldots,p_K} g(p_1, \ldots, p_K) \text{ при услову } \sum_{i=1}^{K} p_i = 1$$

и запишимо њему одговарајућу Лагранжову функцију:

$$\mathcal{L}(p_1, \ldots, p_K, \lambda) = 1 - \sum_{i=1}^{K} p_i^2 - \lambda(\sum_{i=1}^{K} p_i - 1).$$

Како је $\frac{\partial \mathcal{L}}{\partial p_i} = -2p_i - \lambda$ и $\frac{\partial \mathcal{L}}{\partial \lambda} = 1 - \sum_{i=1}^{K} p_i$, следи

$$\frac{\partial \mathcal{L}}{\partial p_i} = 0,\ \frac{\partial \mathcal{L}}{\partial \lambda} = 0 \iff \lambda = -\frac{2}{K},\ p_i = \frac{1}{K},$$

чиме закључујемо да је $p = (\frac{1}{K}, \ldots, \frac{1}{K})$ стационарна тачка функције $\mathcal{L}$. Како је $\frac{\partial^2 \mathcal{L}}{\partial p_i \partial p_j} \equiv 0$ за $i \neq j$ и $\frac{\partial^2 \mathcal{L}}{\partial p_i^2} \equiv -2$, следи

$$\sum_{i,j=1}^{K} \frac{\partial^2 \mathcal{L}}{\partial p_i \partial p_j}(p) h_i h_j = -2h_1 - \cdots - 2h_K,$$

што је мање од 0 кад год је $\sum_{i=1}^{K} h_i^2 > 0$, те је ова квадратна форма негативно дефинитна, а тачка p је решење оптимизационог проблема (2.3).
Преостаје нам још да покажемо да је p условни максимум и када проширимо домен од g тако да важи $p_i \geq 0, \forall i \in \{1, \ldots, K\}$. Нека је онда $p' = (p'_1, \ldots, p'_K)$ тачка таква да је њених $0 < L < K$ координата строго позитивно и за коју важи $\sum_{i=1}^{K} p'_i = 1$. Нека је без умањења општости $p'_{L+1} = \cdots = p'_K = 0$. Како је тада $\sum_{i=1}^{L} p'_i = 1$, према управо показаном следи:

$$g(p') = 1 - \sum_{i=1}^{L} {p'}^2{}_i \leq 1 - \sum_{i=1}^{L} \frac{1}{L^2} < 1 - \sum_{i=1}^{K} \frac{1}{K^2} = g(p).$$

■

Дакле, што се једна категорија више истиче над осталим, то је вредност Џинијевог индекса мања, и обрнуто, што је мања разлика у учесталостима свих категорија, то је вредност Џинијевог индекса већа. Стога је природно за оптимално f изабрати оно пресликавање за које се добијају партиције полазног скупа са што мањим Џинијевим индексима. Како у бинарном стаблу f може индуковати само две партиције, $S_L(f)$ и $S_R(f)$, оптимално f изабраћемо као

$$\arg\min_{f\in Dom}\Big[\frac{|S_L(f)|}{|S_L(f)|+|S_R(f)|}Gini_{S_L(f)}+\frac{|S_R(f)|}{|S_L(f)|+|S_R(f)|}Gini_{S_R(f)}\Big],$$

где је Dom скуп допустивих пресликавања, а $Gini_{S_L(f)}$ и $Gini_{S_R(f)}$ вредности Џинијевог индекса на $S_L(f)$ и $S_R(f)$, редом. Овакво партиционисање је адекватно уколико је вредност горњег израза мања од Џинијевог индекса срачунатог на полазном скупу S.

Алгоритам $CART$ подразумева да се у Dom могу наћи само она пресликавања која на основу непрекидних атрибута X_{con} индукују партиције облика $S_{|X_{con}<t}$ и $S_{|X_{con}\geq t}$, као и пресликавања која на основу категоричких атрибута X_{cat} индукују партиције облика $S_{|X_{cat}=c}$ и $S_{|X_{cat}\neq c}$.

Орезивање стабала се врши након њихове изградње, што не захтева употребу претерано строгих критеријума заустављања. Неретко се са изградњом стабла стаје тек када се у чвору нађу само инстанце исте категорије.

Први корак у орезивању стабла састоји се у креирању низа стабала $T_0,\ldots,T_k$, где је T_0 полазно стабло, а T_k стабло које се састоји само из корена. Стабло T_{i+1} је добијено заменом једног или више подстабала стабла T_i одговарајућим листом, тј. листом са најучесталијом категоријом у корену посматраног подстабла. Подстабла која се орезују су она код којих је дошло до најмањег повећања *грешке по орезаном листу*, односно:

$$\alpha=\frac{\varepsilon(pruned(T,t),S)-\varepsilon(T,S)}{|leaves(T)|-|leaves(pruned(T,t))|},$$

где је $pruned(T,t)$ стабло настало орезивањем подстабла које полази из чвора t, $|leaves(T)|$ број листова у стаблу T и $\varepsilon(T,S)$ удео нетачно класификованих инстанци у скупу за тренирање S на основу модела T.

У следећем кораку свако до стабала $T_0,\ldots,T_k$ се евалуира на скупу за

тестирање, или евентуално унакрсном валидацијом уколико је скуп за тренирање мали, након чега се за коначан модел бира оно стабло са најбољим перформансом.

2.4.2 Алгоритам $C4.5$

Недуго након појаве $CART$ алгоритма Р. Квинлен [8] описује алгоритам $ID3$, који се заснива на концептима теорије информације. Стога ћемо најпре дефинисати следећи појам.

Дефиниција 2.4.2. *Нека важе ознаке из дефиниције* 2.4.1. *Неодређеност или ентропија промењљиве Y у скупу S је величина дефинисана са*

$$H(Y) = -\sum_{i=j}^{K} p_i \log_2(p_i),$$

при чему подразумевамо да је $p_i \log_2(p_i) = 0$ када је $p_i = 0$.

Слично као и Џинијев индекс, ентропија достиже максимум када је Y равномерно расподељена и минимум уколико је дегенерисана.
Како ћемо у наставку подразумевати да је Y увек циљна променљива, ентропију ћемо означавати са $H(S)$, где је S скуп на коме се она рачуна.

За разлику од алгоритма $CART$, Квинлен предлаже да се гранање чворова врши на основу оног атрибута за које се постиже такозвани *највећи добитак информације*. За сваку партицију полазног скупа S на $S_1, \ldots, S_m$ дефинисаћемо добитак информације као разлику ентропије у скупу S и тежинске средине ентропија у скуповима $S_1, \ldots, S_m$:

$$Gain = H(S) - \sum_{i=1}^{m} \frac{|S_i|}{\sum_{j=1}^{m} |S_j|} H(S_i).$$

Овакво партиционисање је адекватно уколико је $Gain > 0$.

У свом оригиналном облику $ID3$ подразумева да се гранање чворова може вршити само на основу категоричких атрибута X_{cat} и да су њима индукована партиционисања облика $S_{|X_{cat}=c_1}, \ldots, S_{|X_{cat}=c_l}$. На тај начин се за разлику од $CART$ алгоритма могу креирати и стабла која нису бинарна. Међутим, на овај начин се фаворизују они атрибути са великим бројем могућих категорија [9]. Из тог разлога исти аутор 1993. године описује алгоритам $C4.5$ [10] који уместо добитка информације користи *нормирани*

добитак како би добио оптимално партиционисање, који се дефинише као:

$$GainRatio = \frac{Gain}{H_{childNodes}},$$

где је $H_{childNodes}$ ентропија расподеле тачака по $S_1, \ldots, S_m$. С увођењем $C4.5$ такође почиње и употреба непрекидних атрибута, који партиционишу S на исти начин као и $CART$ алгоритам.

$C4.5$ такође подразумева орезивање стабала по њиховој изградњи, при чему се ослања на статистичку значајност грешке класификације. Како просечна грешка модела евалуираног на тренинг скупу S није довољно поуздана, $C4.5$ грешку оцењује горњом границом њене интервалне оцене. Другим речима, како $|S| \cdot \varepsilon(T, S)$ има биномну расподелу, посматраћемо

$$\varepsilon_{UB}(T, S) = \varepsilon(T, S) + z_{1-\frac{\alpha}{2}} \sqrt{\frac{\varepsilon(T, S)(1 - \varepsilon(T, S))}{|S|}},$$

где је $z_{1-\frac{\alpha}{2}} = \Phi(1 - \frac{\alpha}{2})$ за жељени праг α.

Поступак орезивања се спроводи од листова ка корену стабла , при чему се за сваки чвор t посматрају следеће вредности:

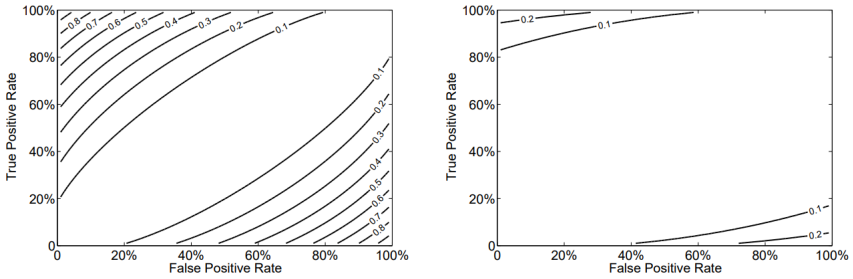
1. $\varepsilon(subtree(T, t), S_t)$
2. $\varepsilon(pruned(subtree(T, t)), S_t)$
3. $\varepsilon(subtree(T, maxchild(T, t)), S_{maxchild(T,t)})$,

где је $maxchild(T, t)$ потомак чвора t у који доспева највише инстанци из S од свих потомака t, а S_t скуп свих инстанци из S које доспевају у чвор t. У зависности од најмање од ових вредности, 1. стабло се не орезује, 2. стабло се орезује у чвору t, 3. чвор t се мења стаблом са кореном у $maxchild(T, t)$.

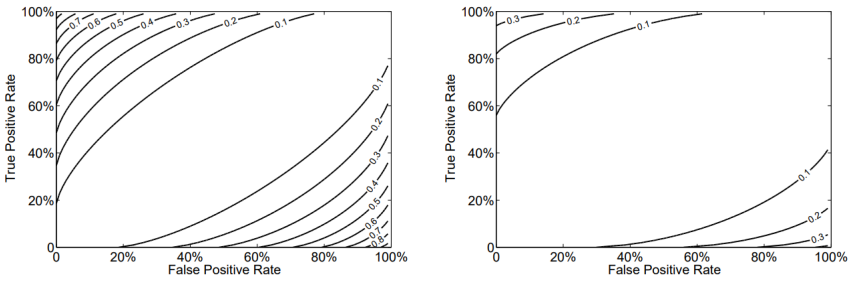
Иако је до данас настао велики број алгоритама за изградњу стабала одлучивања, $CART$ и $C4.5$ су остали водећи конкуренти у практичној примени. Своју популарност дугују врло флексибилној архитектури, захваљујући којој могу бити лако модификовани за конкретне задатке.

Обучавана на небалансираном скупу, стабла одлучивања су такође склона фаворизовању већинске категорије. Иницијална претпоставка о узрочнику ове појаве тицала се неадекватности критеријума гранања чворова, о чему је конкретне резултате изложио П.А. Флах у раду [11].

Флах је посматрао бинарни класификациони модел стабла које се састоји из корена и два листа на које се грана у односу на задати критеријум. Ослањајући се на термине које смо увели у поглављу 1.4, можемо рећи да се таквим моделом у један лист распоређују инстанце означене као TP и FP, а у други инстанце означене као TN и FN. Затим би за унапред задат однос позитивних и негативних категорија у корену стабла представио скуп свих могућих парова (FPR, TPR) који генеришу конкретну вредност критеријума гранања. Поменути скуп ћемо надаље називати изометријском линијом. На сликама 2.3 и 2.4 могу се видети скице изометријских линија за Ђинијев индекс и ентропију на балансираном и небалансираном скупу (однос категорија је $10 : 1$) [12].



Слика 2.3 : Изометријске линије за Ђинијев индекс на балансираном (лево) и небалансираном скупу (десно).



Слика 2.4 : Изометријске линије за ентропију на балансираном (лево) и небалансираном скупу (десно).

Са горњих графика можемо приметити да у случају изражене небалансираности изометријске линије постају све више заравњене, што одговарајући критеријум гранања чини нестабилнијим.

Поглавље 3

Превазилажење проблема

У досадашњем раду упознали смо се са проблемом класификације небалансираних података и описали ситуације услед којих поједини модели могу бити склони непрепознавању мањинске категорије. Ослањајући се на добијене закључке, ово поглавље посвећујемо методама ублажавања ефекта небалансираности, са осврtом на њихове предности и недостатке.

3.1 Селекција предиктора

Незаобилазан корак пре изградње сваког модела јесте припрема података на којима ће се он обучавати. Припрема података се односи на управљање недостајућим вредностима, детекцију и управљење аутлајерима, као и на разне трансформације попут скалирања и слично. Сваки облик припреме података надаље ћемо звати *претпроцесирање*.

По обављеном претпроцесирању неопходно је изабрати одговарајуће атрибуте који ће бити коришћени у моделу, а које ћемо називати *предикторима*. Селекција предиктора је кључан корак у изградњи квалитетног модела јер не само да не носе сви атрибути подједнако важне информације, већ и присуство великог броја предиктора значајно отежава оптимизацију његових параметара.[1]

Један начин на који се предиктори могу изабрати јесте испитивањем квалитета модела изграђених на основу сваке могуће комбинације атрибута. Овакав поступак познат је као метод грубе силе. Међутим, како је тако потребно испитати 2^p комбинација, где је p укупан број атрибута, метод грубе силе постаје исувише рачунски захтеван када је p велико. Стога је препоручено да се скуп полазних атрибута најпре редукује елиминисањем оних атрибута који не испуњавају одређени критеријум, те да се даља селекција врши на скупу преосталих атрибута.

[1] Негативан утицај велике димензије у машинском учењу познат је и као *проклетство димензионалности*.

Елиминисање атрибута се може спровести на разне начине, зависно од специфичности модела. Неретко започиње анализом модела са једним предиктором, којом се значајност сваког атрибута испитује независно од осталих.

Специјално, у моделу логистичке регресије испитивање значајности атрибута се може спровести тестирањем статистичке значајности параметра модела који истом одговара. Како су оцене параметара логистичке регресије добијене методом максималне веродостојности асимптотски непристрасне и нормално расподељене, важи

$$\frac{\widehat{\beta}_i - \beta_i}{SE(\widehat{\beta}_i)} \sim \mathcal{N}(0,1),$$

где је $\widehat{\beta}_i$ оцена параметра β_i у моделу изграђеном на основу i-тог атрибута, а $SE(\widehat{\beta}_i)$ њено стандардно одступање. Сада можемо тестирати хипотезу $H_0 : \beta_i = 0$ против $H_1 : \beta_i \neq 0$ коришћењем Валдове тест статистике:

$$W = \frac{\widehat{\beta}_i}{SE(\widehat{\beta}_i)}$$

и уз чињеницу да $W_{H_0} \sim \mathcal{N}(0,1)$ формирати одговарајуће критичне области. Уколико одбацимо H_0, можемо размотрити укључивање i-тог атрибута у коначни модел.

Поред Валдовог теста на све ширу примену наилазе и непараметарске методе којима се оцењује дискриминантна моћ атрибута у односу на циљну променљиву. На тај начин се атрибути могу рангирати према свом квалитету, након чега ће m најбоље рангираних атрибута проћи у следећи круг селекције. Међутим, поузданост појединих метрика коришћених у методама овог типа умногоме зависи од балансираности зависне променљиве, попут Фишеровог скора и узајамне информације.

Дефиниција 3.1.1. *Нека зависна променљива Y може узети једну од K могућих категорија. Нека је $(X_{i1}, \ldots, X_{in})^T$ вектор вредности атрибута X_i, а $(Y_1, \ldots, Y_n)^T$ вектор вредности променљиве Y на скупу за тренирање. Означимо са $\overline{X_i^k}$ средњу вредност X_i на скупу инстанци обележених k-том категоријом, а са n_k његову кардиналност.*
Фишеров скор атрибута X_i је величина дефинисана са

$$F(X_i) = \frac{S_{Between}(i)}{S_{Within}(i)},$$

при чему је

$$\begin{aligned} S_{Between}(i) &= \sum_{k=1}^{K} n_k(\overline{X_i^k} - \overline{X_i})^2, \\ S_{Within}(i) &= \sum_{k=1}^{K} \sum_{l:Y_l=k} (X_{il} - \overline{X_i^k})^2. \end{aligned} \tag{3.1}$$

Може се показати да је укупна дисперзија $S_{Total}(i) = \sum_{k=1}^{n}(X_{ik} - \overline{X_i})^2$ једнака збиру $S_{Between}(i)$ и $S_{Within}(i)$, где је $\overline{X_i}$ средња вредност X_i на целокупном скупу за тренирање, из чега следи да Фишеров скор мери колико укупне дисперзије потиче од варијација између категорија у односу на укупну дисперзију унутар сваке од њих.

Доказ: Како је

$$\begin{aligned} S_{Between}(i) &= \sum_{k=1}^{K} n_k \overline{X_i^k}^2 - 2\overline{X_i} \sum_{k=1}^{K} n_k \overline{X_i^k} + \overline{X_i^k}^2 \sum_{k=1}^{K} n_k \\ &= \sum_{k=1}^{K} n_k \overline{X_i^k}^2 - 2\overline{X_i} \cdot n\overline{X_i} + n\overline{X_i}^2 \\ &= \sum_{k=1}^{K} n_k \overline{X_i^k}^2 - n\overline{X_i}^2, \end{aligned}$$

а

$$\begin{aligned} S_{Within}(i) &= \sum_{k=1}^{K} \Big(\sum_{l:Y_l=k} X_{il}^2 - 2\overline{X_i^k} \sum_{l:Y_l=k} X_{il} + n_k \overline{X_i^k}^2 \Big) \\ &= \sum_{k=1}^{K} \Big(\sum_{l:Y_l=k} X_{il}^2 - 2\overline{X_i^k} \cdot n_k \overline{X_i^k} + n_k \overline{X_i^k}^2 \\ &= \sum_{k=1}^{K} \Big(\sum_{l:Y_l=k} X_{il}^2 - n_k \overline{X_i^k}^2 \Big), \end{aligned}$$

следи:

$$\begin{aligned} S_{Within}(i) + S_{Between}(i) &= \sum_{k=1}^{K} \sum_{l:Y_l=k} X_{il}^2 - n\overline{X_i}^2 \\ &= \sum_{k=1}^{n} (X_{ik} - \overline{X_i})^2 \\ &= S_{Total}(i). \end{aligned}$$

■

Из изложеног закључујемо да у прилог квалитету X_i иде што већа вредност $F(X_i)$, јер је иста показатељ да се вредности X_i могу кластеровати по категоријама тако да се унутар једног кластера налазе међусобно сличне вредности, док се вредности из различитих кластера значајно разликују. Међутим, како малобројне категорије повлаче и малу дисперзију, S_{Within} потпада под утицај дисперзије бројнијих категорија, што значи да у присуству небалансираних података овај критеријум постаје нестабилнији.

Дефиниција 3.1.2. *Нека су X и Y дискретне случајне величине са вредностима из $\mathcal{X}$ и $\mathcal{Y}$, редом. Тада је са*

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} P_{(X,Y)}(x,y) \log\Big(\frac{P_{(X,Y)}(x,y)}{P_X(x)P_Y(y)}\Big),$$

где је P закон расподеле одговарајуће случајне величине, односно случајног вектора, дефинисана узајамна информација случајних величина X и Y.

Како су X и Y независне акко је $P_{(X,Y)}(x,y) = P_X(x)P_Y(y),\ \forall (x,y) \in \mathcal{X}\times\mathcal{Y}$, тј. акко је $I(X;Y) = 0$, узајамну информацију можемо видети као меру зависности двеју величина. Имајући то у виду, рангирање атрибута се може извршити на основу њихове узајамне информације са зависном променљивом, при чему се $P_{(X_i,Y)}(x,y), P_{X_i}(x)$ и $P_Y(y)$ могу оценити релативним фреквенцијама $f_{x,y}, f_x$ и f_y у скупу за обучавање. Ранг атрибута X_i биће утолико бољи што је $I(X_i;Y)$ веће. Када су у питању атрибути непрекидног типа, препоручено је претходно извршити дискретизацију њихових вредности како не би дошло до сумирања великог броја сабирака. Слично као и код Фишеровог скора, допринос малобројних категорија у узајамној информацији бива теже препознат, што је последица нискофреквентних догађаја

који ту категорију укључују:

$$f_{x,y} \log\left(\frac{f_{x,y}}{f_x f_y}\right) < f_{x,y} \log\left(\frac{1}{f_{x,y}}\right) \xrightarrow{f_{x,y}\to 0} 0.$$

Како би надоместили недостатке метода заснованих на поменутим критеријумима, Л. Јин и др. [13] предлажу разбијање већинске категорије на више *'привремених'* категорија, након чега би се атрибут вредновао у односу на новодобијене категорије на исти начин као и раније. Разбијање се врши кластеровањем инстанци већинске категорије према вредностима посматраног атрибута, након чега свака инстанца добија категорију која одговара њеном кластеру. На слици 3.1 може се видети како разбијање атрибута већинске категорије утиче на вредности Фишеровог скора и узајамне информације.

Слика 3.1 : Фишеров скор и узајамна информација пре и након разбијања већинске категорије.

Коначно, представљамо и критеријум који показује извесну робусност на небалансираност категорија, а који је примењив у задацима бинарне класификације. Са тим циљем, вратићемо се употреби термина позитивна и негативна категорија.

Дефиниција 3.1.3. *Нека су X и Y случајне величине са истим скупом вредности и са законом расподеле $P = (p_1, \ldots, p_m)$ и $Q = (q_1, \ldots, q_m)$. Мера дивергенције расподела P и Q дефинисана са*

$$H(P,Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{m} (\sqrt{p_i} - \sqrt{q_i})^2}$$

назива се Хелингерово растојање.

Хелингерово растојање се може искористити као мера разлике расподела позитивне и негативне категорије, X_i^+ и X_i^-, по вредностима категоричког, односно по дискретизованим вредностима непрекидног атрибута X_i. Дакле, ако X_i на скупу за тренирање узима n (дискретизованих) вредности и ако са f_+^j и f_-^j означимо релативне фреквенције позитивне, односно негативне категорије у скупу инстанци са j-ом вредноћ-ћу X_i, а са f_+ и f_- релативну фреквенцију позитивне, одноно негативне категорије у целокупном скупу за тренирање, Хелингерово растојање расподела X_i^+ и X_i^- можемо израчунати као:

$$H(X_i^+, X_i^-) = \sqrt{\sum_{j=1}^{n} \left(\sqrt{\frac{f_+^j}{f_+}} - \sqrt{\frac{f_-^j}{f_-}} \right)^2}.$$

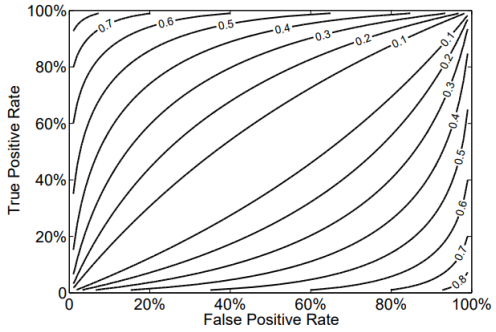
Веће вредности $H(X_i^+, X_i^-)$ су показатељ значајне разлике између X_i^+ и X_i^-, што повлачи и бољи ранг атрибута X_i.

Робусност Хелингеровог растојања на небалансираност категорија се може оправдати Флаховим моделом (слика 3.2), јер је у терминима FPR и TPR оно једнако:

$$\sqrt{\left(\sqrt{\frac{TP}{P}} - \sqrt{\frac{FP}{N}} \right)^2 + \left(\sqrt{\frac{FN}{P}} - \sqrt{\frac{TN}{N}} \right)^2} =$$

$$\sqrt{\left(\sqrt{TPR} - \sqrt{FPR} \right)^2 + \left(\sqrt{1 - TPR} - \sqrt{1 - FPR} \right)^2},$$

што је у потпуности независно од односа позитивне и негативне категорије.



Слика 3.2 : Изометријске линије Хелингеровог растојања [12].

Специјално, како се у моделу стабла одлучивања селекција предиктора врши приликом гранања чворова, у [12] је предложен модификовани алгоритам $C4.5$, код ког се гранање чвора врши тако да Хелингерово растојање зависне променљиве срачунато на чворовима потомцима буде максимално.

3.2 Методе реузорковања

Међу првим предлозима за побољшање квалитета модела нашла се идеја о елиминисању утицаја небалансираности на узорачком нивоу. Овакав приступ подразумева обучавање модела на новом балансираном скупу, који је настао од полазног применом одговарајућих метода реузорковања.
Методе реузорковања могу се грубо поделити на две групе: методе смањивања (енг. *downsampling* или *undersampling*) и методе увећавања (енг. *upsampling* или *oversampling*).

Прву групу чине методе које врше елиминацију инстанци већинске категорије све док се њихов број у преосталом скупу не приближи броју инстанци мањинске категорије. Иако врло интуитивно и лако за имплементацију, овакво реузорковање може резултирати губитком важних информација о већинској категорији, нарочито у случају изражене небалансираности. Примери често коришћених метода смањивања су следећи:

- *метода случајног избора* [14]
  најједноставнија метода којом се инстанце већинске категорије бирају на случајан начин. Како случајно изабране инстанце могу створити погрешну слику о расподели унутар већинске категорије, ова метода ретко даје завидне резултате;

- *метода заснована на кластеровању* [15]
  метода која има за циљ очување расподеле атрибута унутар већинске категорије; у случају бинарне класификације, скуп за тренирање се најпре партиционише на L кластера, а потом се из сваког кластера бира одговарајући број инстанци већинске категорије. Уколико са pos означимо укупан број инстанци мањинске категорије, а са pos_i и neg_i број инстанци мањинске, односно већинске категорије i-тог кластера, препоручени број изабраних инстанци из i-тог кластера је приближно једнак:
  $$pos \cdot \frac{\frac{neg_i}{pos_i}}{\sum_{j=1}^{L} \frac{neg_j}{pos_j}}.$$

Специјално, уколико у i-том кластеру нема инстанци мањинске категорије, подразумевана вредност pos_i је 1. Уопштење ове методе у вишекласном случају се врши на следећи начин:

1. нека је $S = \emptyset$ скуп изабраних инстанци;
2. категорије се сортирају растуће по величини;
3. S постаје скуп свих инстанци најмање категорије по величини;
4. за сваку категорију почевши од друге најмање врши се реузорковање одговарајућом методом, при чему се посматрана категорија сматра већинском, а најмања по величини мањинском. Изабране инстанце већинске категорије се потом додају у S.

У овом раду је, за потребе посматране методе, кластеровање скупа за тренирање извршено помоћу алгоритма $DBSCAN$[2], који проналази регије густо насељених тачака и сврстава их у један кластер. На тај начин број резултујућих кластера није унапред познат, већ се добија као излаз из алгоритма.

Кажемо да је ε околина тачке $\boldsymbol{x}$ густа уколико се у њој налази барем $minPts$ тачака из скупа за тренирање, при чему је одговарајућа околина одређена на основу метрике d. Држећи се наведених ознака, скица $DBSCAN$ алгоритма је следећа [16]:

[2] енг. *density-based spatial clustering of applications with noise*

Алгоритам 2 : $DBSCAN(S, d, \varepsilon, minPts)$

Улаз: S - скуп за тренирање, d - метрика, ε - полупречник околине, $minPts$ - најмањи број суседа у ε околини тачке неопходан да би се околина прогласила густом

Излаз: $labeled(S)$ - скуп инстанци из S са ознаком редног броја кластера

$labeled(S) \leftarrow$ скуп инстанци из S без ознака кластера;
$C \leftarrow 0$;
for $\boldsymbol{x} \in S$:
  if $\boldsymbol{x}$ је означена:
    continue;
  end
  $neighbours \leftarrow \varepsilon$ околина тачке $\boldsymbol{x}$;
  if $|neighbours| < minPts$:
    $label(\boldsymbol{x}) \leftarrow -1$;
    continue;
  end
  $C \leftarrow C + 1$;
  $label(\boldsymbol{x}) \leftarrow C$;
  $S_{tmp} \leftarrow neighbours \setminus \{\boldsymbol{x}\}$;
  for $\boldsymbol{y} \in S_{tmp}$:
    if $label(\boldsymbol{y}) = -1$:
      $label(\boldsymbol{y}) = C$;
    end
    if $\boldsymbol{y}$ је означена:
      continue;
    end
    $label(\boldsymbol{y}) \leftarrow C$;
    $neighbours \leftarrow \varepsilon$ околина тачке $\boldsymbol{y}$;
    if $|neighbours| \geq minPts$:
      $S_{tmp} \leftarrow S_{tmp} \cup neighbours$;
    end
  end
end

Инстанце означене са -1 представљају одударајуће тачке и не припадају ниједном кластеру, због чега алгоритам испољава извесну робусност на присуство аутлајера. Параметри ε и $minPts$ се могу оценити емпиријски или помоћу $k - dist$ графа описаног у [17].

Супротно смањивању, методама увећавања се балансираност постиже понављањем или вештачким генерисањем нових инстанци мањинске категорије по одређеном шаблону. У првом случају се из мањинске категорије извлачи узорак са понављањем, што је неретко рачунски једноставније, али и слабије по перформансама у односу на приступ генерисања нових инстанци.

Међу најшире коришћеним методама увећавања истичу се следеће:

- *метода случајног избора* [14]
  слично попут методе смањивања случајним избором, овом методом се мањинска категорија проширује инстанцама изабраним из исте категорије на случајан начин. Велики недостатак случајног избора поновљених инстанци проистиче из чињенице да се тако у новодобијеном скупу могу више пута наћи мање информативне инстанце, што се негативно одражава на обучавање модела;

- *SMOTE*[3] *метода* [18]
  једна од најпознатијих метода заснованих на генерисању нових инстанци; метода има за циљ да новодобијене инстанце буду блиске стварним инстанцама мањинске категорије у смислу одговарајуће метрике, те се оне генеришу на следећи начин:

  1. за сваку инстанцу мањинске категорије $\boldsymbol{x}_i$ посматрамо њених k најближих суседа из исте категорије;
  2. на основу случајно изабраног суседа $\boldsymbol{x}_i^{neighbour}$ генерише се нова инстанца као конвексна комбинација тог суседа и дате инстанце:

  $$\boldsymbol{x}_i^{new} = \boldsymbol{x}_i + \alpha(\boldsymbol{x}_i^{neighbour} - \boldsymbol{x}_i),$$

  где је α случајно изабран број из интервала $(0, 1)$.

  Други корак се понавља онолико пута колико је потребно да се постигне приближна балансираност класа.

[3] енг. *syntetic minority oversampling technique*

- *метода заснована на кластеровању* [19]
  ова метода подразумева примену методе увећавања случајним избором из кластера тачака присутних у свим категоријама, чиме се елиминише утицај међукластерске и унутаркластерске небалансираности. Узорковање, односно генерисање нових инстанци из сваког кластера врши се док се кардиналност посматраног кластера не изједначи са кардиналношћу највећег кластера у већинској категорији.

3.3 Методе осетљиве на цену грешке

У многим ситуацијама грешка класификације над тачкама одређених категорија носи знатно већи ризик од осталих типова грешака. Како традиционалне методе обучавања модела придају подједнаки значај сваком типу грешке, није за очекивати да ће добијени модел мање грешити над најкритичнијим тачкама. Имајући то у виду, настала је нова подобласт машинског учења - учење осетљиво на цену грешке, чије методе модификацијом оптимизационог израза постижу задати циљ. Таква модификација обично подразумева додавање умножака грешака модела над одабраним тачкама и одговарајућих тежина, при чему је тежина утолико већа што је већа и цена грешке.

Ове методе нашле су примену и у нашој проблематици, како би доделом већих тежина назначиле грешке модела над мањинским категоријама, јер исте теже долазе до изражаја због мање кардиналности унутар скупа за тренирање.

- *Модел тежинске логистичке регресије* [20]
  Оцене параметара овог модела добијају се као тачка максимума модификованог логаритма функције веродостојности, који гласи:

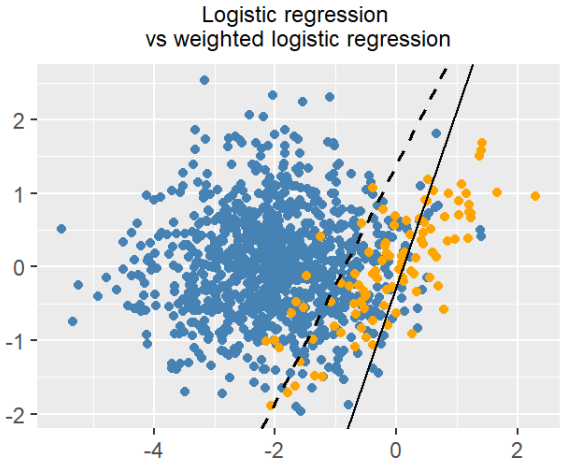
$$\omega_1 \sum_{\{i:\ y_i=1\}} \log \pi(\boldsymbol{x}_i) + \omega_0 \sum_{\{i:\ y_i=0\}} \log(1-\pi(\boldsymbol{x}_i)).$$

  Одговарајуће тежине се могу одредити на исти начин као и сви хиперпараметри које смо до сада описали, док [20] предлажу следећи избор:

$$\omega_i = \frac{\text{величина тренинг скупа}}{\text{број категорија} \cdot \text{број инстанци } i\text{-те категорије}}.$$

  На слици 3.3 приказане су хиперравни модела логистичке и тежинске

логистичке регресије обучаваних на небаланисраном скупу. Можемо приметити да додавање тежина у модел јесте резултирало бољим препознавањем мањинске категорије.



Слика 3.3 : Поређење логистичке регресије (пуна линија) и тежинске логистичке регресије (испрекидана линија).

- *Тежинско гласање у моделу kNN* [21]
  Још један начин на који се може сузбити утицај већинских категорија у моделу kNN, тиче се замене класификатора (2.2) следећим:

$$r(\boldsymbol{x}) = \underset{j\in\{1,\dots,K\}}{\arg\max} \frac{\sum_{\boldsymbol{x}_i\in B_k(\boldsymbol{x})} \omega_i I\{y_i = j\}}{k},$$

  где је $\omega_i = K\left(\frac{d(\boldsymbol{x},\boldsymbol{x}_i)}{d(\boldsymbol{x},\boldsymbol{x}_{k+1})}\right)$, $\boldsymbol{x}_{k+1}$ је најближи сусед тачки $\boldsymbol{x}$ који није у $B_k(\boldsymbol{x})$, а функција $K(\cdot)$ задовољава:

  - $K(t) \geq 0,\ \forall t \in \mathbb{R}$
  - K достиже максимум за $t = 0$
  - K опада на $\mathbb{R}_0^+$

  Дакле, доношење коначне одлуке при оваквој класификацији умногоме зависи и од растојања између суседа и циљне инстанце, јер ће они суседи који су на најмањем растојању од циљне инстанце добити највећу тежину.

- *Модел SVM са тежинским одступањима* [5]
  Контролисање утицаја одређених инстанци на модел SVM може се

постићи додељивањем тежина променљивим ξ_i, што доводи до новог регуларизационог израза:

$$\frac{\|\omega\|^2}{2} + C\sum_{i=1}^{n}\omega_i\xi_i.$$

При оваквом приступу дозвољено одступање тачака од хиперравни одређених потпорним векторима одговарајућих категорија зависи од важности самих тачака, за разлику од првобитног израза (2.1) у коме је одступање униформно контролисано само једним хиперпараметром.

3.4 Методе засноване на ансамблима

Употреба машинског учења у најразличитијим областима до данас је изнедрила велики број модела. Како су сви модели склони грешкама, али не у истој мери и над истим подскуповима простора у коме леже инстанце над којима се модели обучавају, настала је идеја о употреби *ансамбала*, који комбиновањем више тзв. *базних* модела постижу бољи резултат него што би постигли модели обучавани појединачно.

Алгоритми за изградњу ансамбала у надгледаном учењу могу се угрубо разврстати у две велике фамилије: алгоритми просте агрегације (енг. *bagging*) и алгоритми појачавања (енг. *boosting*).
Под простом агрегацијом подразумевамо подучавање више модела који не узимају у обзир грешке оних других, након чега се њиховим усредњавањем (регресија), односно прегласавањем (класификација) добија предикција ансамбла. Појачавањем је сваки следећи модел конструисан тако да надомести недостатке претходног.

Специјално, у комбинацији са претходно описаним методама ансамбли играју значајну улогу и у класификацији небалансираних података.

3.4.1 Проста агрегација

Најпознатији представник алгоритама просте агрегације јесу случајне шуме (енг. *random forest*) [22]. Модели на основу којих се доноси предикција случајних шума су стабла одлучивања конструисана *CART* алгоритмом. Свако стабло је обучавано на бутстреп узорку[4] из целокупног скупа за тренирање и користи случајно изабран подскуп свих доступних атрибута,

[4] прост случајан узорак са понављањем исте величине као и популација из које се извлачи

како би грешке које стабла праве биле што мање корелисане и лакше се поништиле при агрегацији.

Како би се унапредио перформанс случајних шума при небалансираној класификацији, [23] предлажу следећу модификацију:

- *балансиране случајне шуме*
  како бутстреп узорак из небалансираног скупа тежи да задржи то својство, ова модификација подразумева балансирање скупа за тренирање на основу ког се свако стабло обучава: новодобијени тренажни скуп се састоји из уније бутстреп узорка извученог из мањинске категорије и простих случајних узорака са понављањем исте величине извучених из преосталих категорија;

Поред модификованих случајних шума истаћи ћемо и следеће алгоритме:

- *SMOTEBagging алгоритам* [24]
  базни модели се обучавају на балансираним скуповима који су настали комбинацијом $SMOTE$ методе и методе увећавања случајним избором;
- *UnderBagging алгоритам* [25]
  слично, базни модели се обучавају на балансираним скуповима који су настали методом смањивања случајним избором.

3.4.2 Појачавање

За разлику од просте агрегације код које се модели обучавају независно једни од других, алгоритми појачавања настоје да при обучавању сваког следећег модела узму у обзир понашање претходног како би се више усмерили на оне инстанце на којима претходник греши. Најпознатији представник ове фамилије јесте алгоритам бинарне класификације *AdaBoost* (енг. *adaptive boosting*).

AdaBoost је итеративни алгоритам чија предикција зависи од комбинације модела обучаваних у свакој итерацији. Премда то није неопходно, *AdaBoost* у највећем броју случајева такође комбинује стабла одлучивања. У сврху описа алгоритма означићемо могуће категорије циљне променљиве са -1 и 1.

Нека је F_t ансамбл конструисан у кораку t. Како је циљ да F_t греши мање на оним инстанцама на којим греши F_{t-1}, F_t можемо записати као:

$$F_t = F_{t-1} + \alpha_t f_t,$$

где су $\alpha_t > 0$, а $f_t \in \{-1, 1\}$ предикција базног модела изабрани тако да $\alpha_t f_t$ минимизује грешку ансамбла F_{t-1}. Како је предикција ансамбла једнака $sign(F_t(\boldsymbol{x}))$, погодно је користити експоненцијалну функцију грешке $L(u, v) = e^{-uv}$, уз чију употребу α_t и f_t можемо добити решавањем следећег оптимизационог проблема:

$$\begin{aligned}\min_{\alpha,f} \sum_{i=1}^{n} e^{-y_i(F_{t-1}(\boldsymbol{x}_i)+\alpha f(\boldsymbol{x}_i))} &= \min_{\alpha,f} \sum_{i=1}^{n} \omega_i^t e^{-y_i \alpha f(\boldsymbol{x}_i)} \\ &= \min_{\alpha,f} \Big[e^{-\alpha} \sum_{i|f(\boldsymbol{x}_i)=y_i} \omega_i^t + e^{\alpha} \sum_{i|f(\boldsymbol{x}_i)\neq y_i} \omega_i^t\Big],\end{aligned}$$

где је $\{(\boldsymbol{x}_i, y_i) : y_i \in \{-1, 1\}\}_{i=1}^n$ скуп за тренирање, а $\omega_i^t = e^{-y_i F_{t-1}(\boldsymbol{x}_i)}$ тежина i-те инстанце.

Како је горњи израз једнак

$$\min_{\alpha,f} \Big[(e^{\alpha} - e^{-\alpha}) \sum_{i=1}^{n} \omega_i^t I\{f(\boldsymbol{x}_i) \neq y_i\} + e^{-\alpha} \sum_{i=1}^{n} \omega_i^t\Big],$$

за фиксно $\alpha > 0$ решење по f се добија као

$$\arg\min_{f} \sum_{i=1}^{n} \omega_i^t I\{f(\boldsymbol{x}_i) \neq y_i\}.$$

Даље, заменом познатог f_t у полазни проблем добијамо α_t као

$$\arg\min_{\alpha} \Big[e^{-\alpha} \sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t + e^{\alpha} \sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t\Big].$$

Лема 3.4.1.

$$\arg\min_{\alpha} \Big[e^{-\alpha} \sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t + e^{\alpha} \sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t\Big] = \frac{1}{2} \log \frac{\sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t}{\sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t}.$$

Доказ: Нека је $L_{f_t}(\alpha) = e^{-\alpha} \sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t + e^{\alpha} \sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t$.

Важи:

$$\begin{aligned} L'_{f_t}(\alpha) = 0 &\iff e^{-\alpha} \sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t + e^{\alpha} \sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t = 0 \\ &\iff e^{2\alpha} \sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t = \sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t \\ &\iff \alpha = \frac{1}{2} \log \frac{\sum_{i|f_t(\boldsymbol{x}_i)=y_i} \omega_i^t}{\sum_{i|f_t(\boldsymbol{x}_i)\neq y_i} \omega_i^t} \end{aligned}$$

Такође, $L'_{f_t}(\alpha) < 0$ лево, а $L'_{f_t}(\alpha) > 0$ десно од тачке екстремума, чиме је доказ завршен. ■

Специјално, како би се ставио акценат на нетачно класификоване инстанце у претходној итерацији, врши се адаптација тежина:

$$\omega_i^{t+1} = \omega_i^t e^{-\alpha_t y_i f_t(\boldsymbol{x}_i)},$$

односно $\omega_i^{t+1} = \omega_i^t e^{-\alpha_t}$ за тачно, а $\omega_i^{t+1} = \omega_i^t e^{\alpha_t}$ за нетачно класификоване инстанце. Цео алгоритам *AdaBoost*, као и његова вишекласна уопштења *AdaBoost.M*1 и *AdaBoost.M*2, могу се видети у [26].

AdaBoost се такође може надоградити тако да се при изградњи базних модела поред тачности класификације посматра и балансираност класа:

- *SMOTEBoost алгоритам* [27]
  алгоритам који у свакој итерацији за изградњу базних модела користи балансирани скуп настао применом *SMOTE* методе; при одређивању коефицијената који стоје уз добијене моделе користи се оригинални скуп за тренирање.
- *RUSBoost алгоритам* [28]
  слично као и код претходног алгоритма, базни модели се граде на балансираном скупу насталом применом методе смањивања случајним избором.

Поглавље 4

Примери

У претходним поглављима дат је преглед модела осетљивих на небалансираност података као и предлога за њихово унапређење. Изложене хипотезе су експериментално тестиране на примерима бинарне и вишекласне класификације, а сви кодови коришћени за добијање резултата који следе могу се видети на линку https://github.com/Nadja1997/Imbalanced-Classification/tree/master.

4.1 Бинарна класификација

Default of Credit Card Clients [29] је скуп података о неизмирењу обавеза корисника тајванских кредитних картица. Састоји се од 23 атрибута и бинарне циљне променљиве која означава да ли је дошло до неизмирења обавеза наредног месеца (табела 4.1).

Назив променљиве	Тип	Опис
ID		
LIMIT_BAL	нумерички	износ кредита у TWD
SEX	категорички	пол: 1 = мушки; 2 = женски
EDUCATION	категорички	образовање: 1 = постдипломско; 2 = факултетско; 3 = средњошколско; 4 = остало
MARRIAGE	категорички	брачни статус: 1 = у браку; 2 = није у браку; 3 = остало
AGE	нумерички	године старости
PAY_0 - PAY_6	нумерички	месечно кашњење у плаћању у периоду од априла до септембра 2005. године:
BILL_AMT0 - BILL_AMT6	нумерички	извод са рачуна у TWD у периоду од априла до септембра 2005. године:
PAY_AMT1 - PAY_AMT16	нумерички	претходно плаћање у TWD у периоду од априла до септембра 2005. године:
default payment next month	категорички	зависна променљива: 1 = да; 0 = не;

Табела 4.1 : Опис скупа *Default of Credit Card Clients*

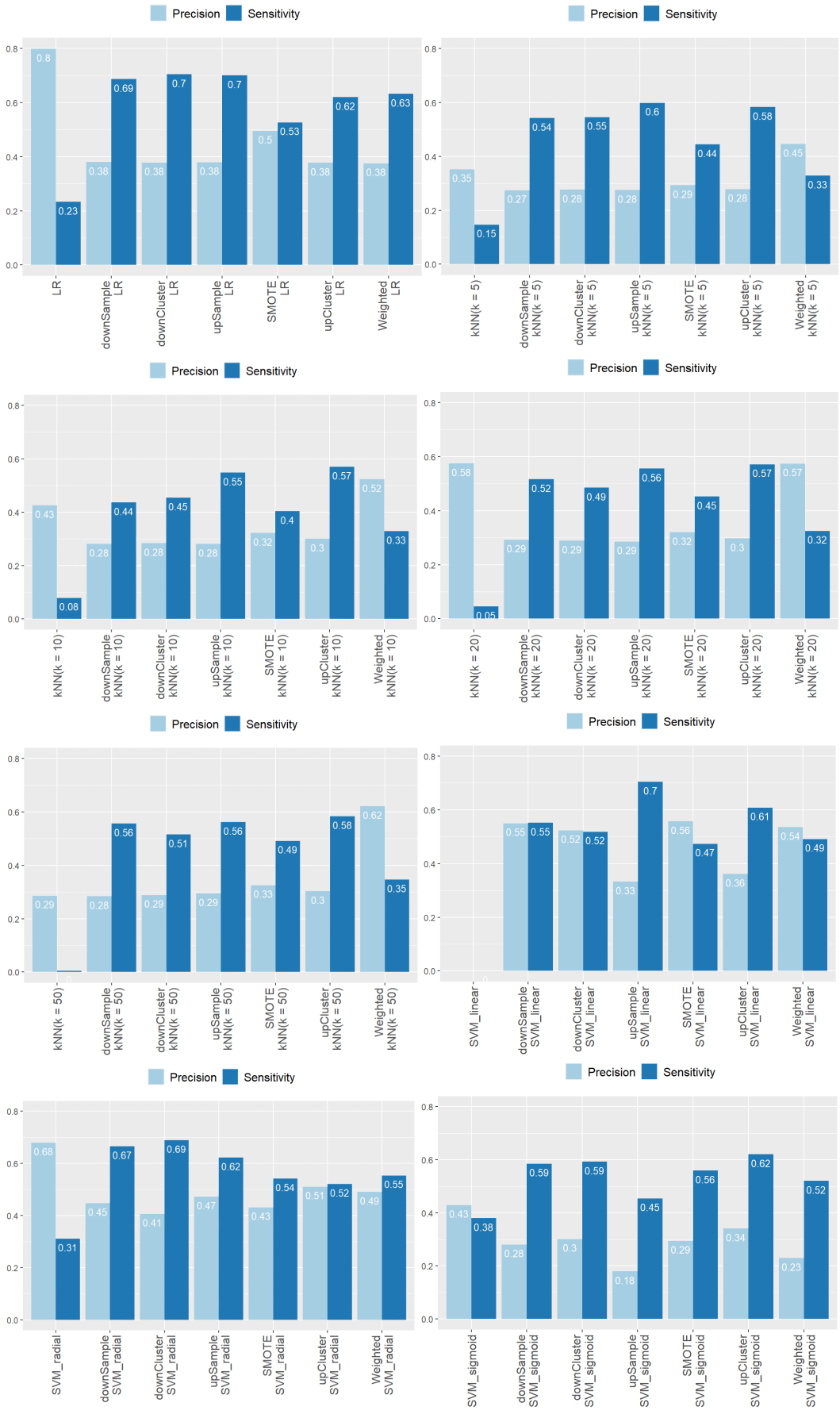
Корисника са статусом неизмирења обавеза има приближно 3.5 пута мање, што овај скуп чини небалансираним. На скупу за тренирање који

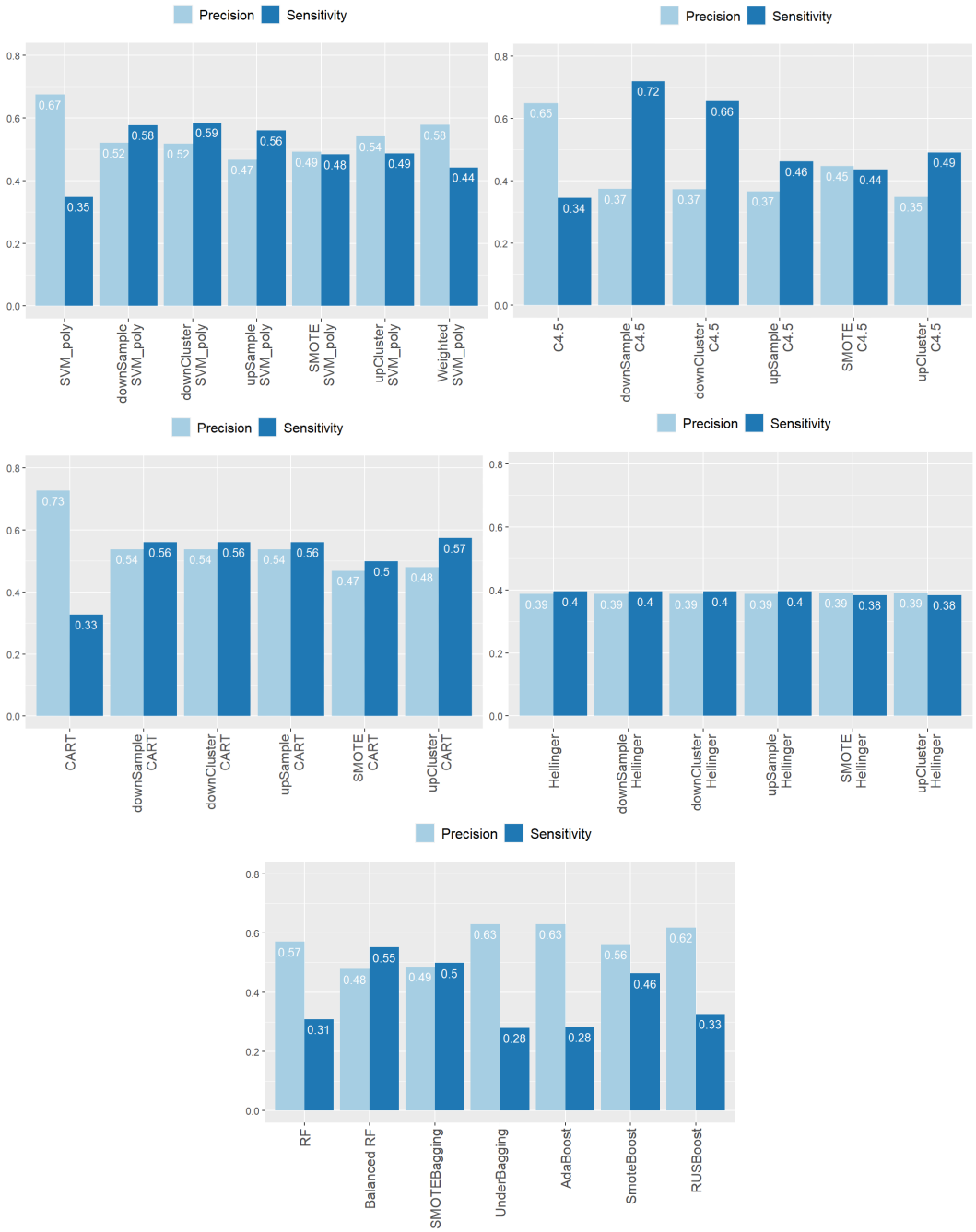
чини око 80% целокупног скупа података утренирани су модели о којима је било речи у овом раду, као и модели на које су примењене методе наведене у трећем поглављу. Тестирање модела спроведено је на скупу за тестирање који чини преосталих 20% полазног скупа.

	Accuracy	AUC	F1	Precision	Sensitivity	Specificity
LR	0.817	0.754	0.361	0.799	0.233	0.983
kNN(k = 5)	0.751	0.605	0.207	0.352	0.147	0.923
kNN(k = 10)	0.772	0.611	0.132	0.426	0.078	0.97
kNN(k = 20)	0.781	0.621	0.083	0.575	0.045	0.991
kNN(k = 50)	0.777	0.624	0.008	0.286	0.004	0.997
SVM_linear	0.778	*NA*	*NA*	*NA*	0	1
SVM_radial	0.815	*NA*	0.427	0.679	0.311	0.958
SVM_sigmoid	0.751	*NA*	0.403	0.429	0.38	0.856
SVM_poly	0.818	*NA*	0.459	0.674	0.348	0.952
CART	0.824	0.652	0.451	0.726	0.327	0.965
C4.5	0.814	0.644	0.45	0.649	0.344	0.947
Hellinger	0.727	*NA*	0.391	0.387	0.395	0.822
downSample_LR	0.683	0.751	0.49	0.38	0.687	0.681
downSample_kNN(k = 5)	0.582	0.605	0.365	0.275	0.542	0.593
downSample_kNN(k = 10)	0.627	0.608	0.342	0.281	0.436	0.682
downSample_kNN(k = 20)	0.616	0.613	0.373	0.292	0.517	0.644
downSample_kNN(k = 50)	0.59	0.618	0.375	0.283	0.556	0.6
downSample_SVM_linear	0.801	*NA*	0.551	0.55	0.552	0.871
downSample_SVM_radial	0.744	*NA*	0.535	0.447	0.665	0.766
downSample_SVM_sigmoid	0.575	*NA*	0.379	0.28	0.585	0.572
downSample_SVM_poly	0.788	*NA*	0.547	0.52	0.577	0.848
downSample_CART	0.796	0.711	0.548	0.538	0.56	0.863
downSample_C4.5	0.67	0.677	0.492	0.373	0.72	0.656
downSample_Hellinger	0.727	*NA*	0.391	0.387	0.395	0.822
downCluster_LR	0.678	0.753	0.492	0.378	0.705	0.67
downCluster_kNN(k = 5)	0.583	0.598	0.367	0.277	0.546	0.593
downCluster_kNN(k = 10)	0.625	0.608	0.349	0.284	0.454	0.674
downCluster_kNN(k = 20)	0.621	0.614	0.362	0.289	0.485	0.66
downCluster_kNN(k = 50)	0.611	0.621	0.37	0.288	0.515	0.638
downCluster_SVM_linear	0.789	*NA*	0.521	0.524	0.519	0.866
downCluster_SVM_radial	0.708	*NA*	0.511	0.406	0.689	0.713
downCluster_SVM_sigmoid	0.605	*NA*	0.399	0.301	0.593	0.608
downCluster_SVM_poly	0.788	*NA*	0.55	0.518	0.585	0.845
downCluster_CART	0.796	0.711	0.548	0.538	0.56	0.863
downCluster_C4.5	0.679	0.662	0.475	0.373	0.656	0.686
downCluster_Hellinger	0.727	*NA*	0.391	0.387	0.395	0.822
upSample_LR	0.68	0.753	0.492	0.38	0.701	0.674
upSample_kNN(k = 5)	0.562	0.605	0.377	0.275	0.599	0.552
upSample_kNN(k = 10)	0.589	0.614	0.371	0.281	0.548	0.601
upSample_kNN(k = 20)	0.594	0.617	0.377	0.286	0.556	0.604
upSample_kNN(k = 50)	0.605	0.632	0.386	0.294	0.562	0.617
upSample_SVM_linear	0.621	*NA*	0.452	0.332	0.705	0.597
upSample_SVM_radial	0.762	*NA*	0.537	0.473	0.622	0.802
upSample_SVM_sigmoid	0.419	*NA*	0.257	0.179	0.454	0.409
upSample_SVM_poly	0.761	*NA*	0.509	0.467	0.56	0.818
upSample_CART	0.796	0.711	0.548	0.538	0.56	0.863
upSample_C4.5	0.703	0.614	0.408	0.366	0.462	0.772
upSample_Hellinger	0.727	*NA*	0.391	0.387	0.395	0.822

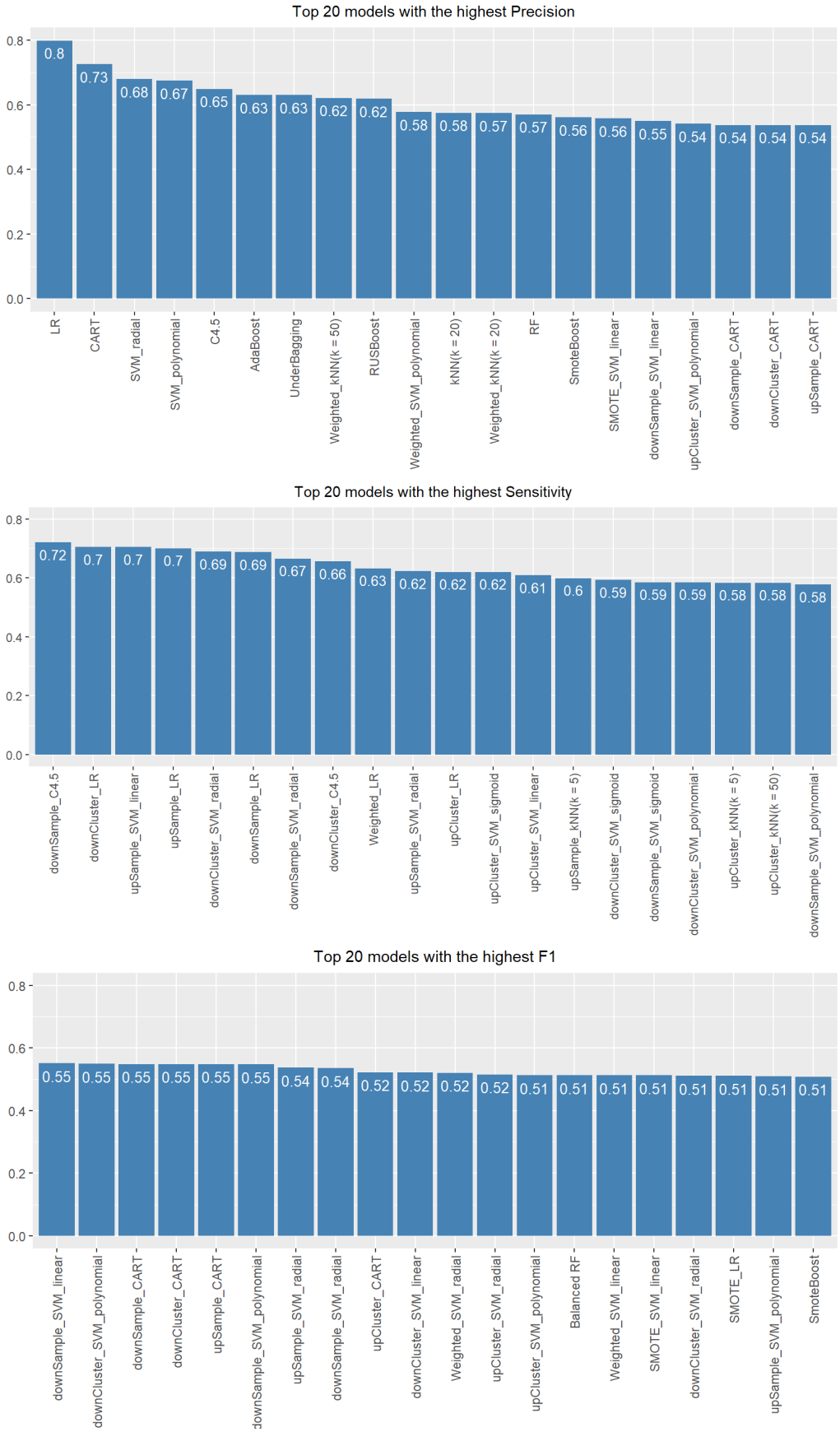
	Accuracy	AUC	F1	Precision	Sensitivity	Specificity
SMOTE_LR	0.776	0.702	0.51	0.495	0.526	0.847
SMOTE_kNN(k = 5)	0.639	0.606	0.353	0.293	0.444	0.695
SMOTE_kNN(k = 10)	0.679	0.624	0.358	0.321	0.403	0.758
SMOTE_kNN(k = 20)	0.666	0.633	0.375	0.32	0.452	0.727
SMOTE_kNN(k = 50)	0.661	0.642	0.391	0.325	0.491	0.71
SMOTE_SVM_linear	0.8	*NA*	0.512	0.558	0.474	0.893
SMOTE_SVM_radial	0.74	*NA*	0.48	0.431	0.542	0.797
SMOTE_SVM_sigmoid	0.605	*NA*	0.385	0.294	0.56	0.617
SMOTE_SVM_poly	0.775	*NA*	0.488	0.492	0.483	0.858
SMOTE_CART	0.763	0.72	0.483	0.468	0.499	0.838
SMOTE_C4.5	0.755	0.699	0.442	0.447	0.436	0.846
SMOTE_Hellinger	0.731	*NA*	0.387	0.39	0.384	0.83
upCluster_LR	0.689	0.71	0.469	0.377	0.62	0.709
upCluster_kNN(k = 5)	0.573	0.608	0.377	0.279	0.583	0.57
upCluster_kNN(k = 10)	0.61	0.628	0.393	0.3	0.569	0.621
upCluster_kNN(k = 20)	0.605	0.64	0.391	0.297	0.571	0.614
upCluster_kNN(k = 50)	0.609	0.634	0.398	0.302	0.583	0.617
upCluster_SVM_linear	0.675	*NA*	0.453	0.361	0.609	0.694
upCluster_SVM_radial	0.783	*NA*	0.516	0.511	0.521	0.858
upCluster_SVM_sigmoid	0.65	*NA*	0.44	0.34	0.62	0.658
upCluster_SVM_poly	0.795	*NA*	0.513	0.541	0.487	0.882
upCluster_CART	0.768	0.699	0.522	0.48	0.573	0.823
upCluster_C4.5	0.683	0.607	0.407	0.348	0.491	0.738
upCluster_Hellinger	0.731	*NA*	0.387	0.39	0.384	0.83
Weighted_LR	0.686	0.711	0.471	0.376	0.632	0.701
Weighted_SVM_linear	0.793	*NA*	0.512	0.535	0.491	0.879
Weighted_SVM_radial	0.774	*NA*	0.52	0.49	0.554	0.836
Weighted_SVM_sigmoid	0.506	*NA*	0.318	0.229	0.521	0.502
Weighted_SVM_poly	0.805	*NA*	0.501	0.578	0.442	0.908
Weighted_kNN(k = 5)	0.761	*NA*	0.378	0.446	0.329	0.884
Weighted_kNN(k = 10)	0.785	*NA*	0.404	0.523	0.329	0.915
Weighted_kNN(k = 20)	0.797	*NA*	0.415	0.574	0.325	0.931
Weighted_kNN(k = 50)	0.808	*NA*	0.445	0.621	0.346	0.94
RF	0.795	0.755	0.401	0.57	0.309	0.934
Balanced RF	0.768	0.766	0.513	0.479	0.552	0.829
SMOTEBagging	0.772	0.733	0.493	0.487	0.499	0.85
UnderBagging	0.804	0.712	0.388	0.63	0.28	0.953
AdaBoost	0.804	0.756	0.391	0.63	0.284	0.953
SmoteBoost	0.801	0.767	0.508	0.562	0.464	0.897
RUSBoost	0.806	0.753	0.428	0.619	0.327	0.943

Табела 4.2 : Евалуација модела.


Precision Sensitivity
0.8 0.23 0.38 0.69 0.38 0.7 0.38 0.7 0.5 0.53 0.38 0.62 0.38 0.63
LR downSample LR downCluster LR upSample LR SMOTE LR upCluster LR Weighted LR
Precision Sensitivity
0.35 0.15 0.27 0.54 0.28 0.55 0.28 0.6 0.29 0.44 0.28 0.58 0.45 0.33
kNN(k = 5) downSample kNN(k = 5) downCluster kNN(k = 5) upSample kNN(k = 5) SMOTE kNN(k = 5) upCluster kNN(k = 5) Weighted kNN(k = 5)
Precision Sensitivity
0.43 0.08 0.28 0.44 0.28 0.45 0.28 0.55 0.32 0.4 0.3 0.57 0.52 0.33
kNN(k = 10) downSample kNN(k = 10) downCluster kNN(k = 10) upSample kNN(k = 10) SMOTE kNN(k = 10) upCluster kNN(k = 10) Weighted kNN(k = 10)
Precision Sensitivity
0.58 0.05 0.29 0.52 0.29 0.49 0.29 0.56 0.32 0.45 0.3 0.57 0.57 0.32
kNN(k = 20) downSample kNN(k = 20) downCluster kNN(k = 20) upSample kNN(k = 20) SMOTE kNN(k = 20) upCluster kNN(k = 20) Weighted kNN(k = 20)
Precision Sensitivity
0.29 0.28 0.56 0.29 0.51 0.29 0.56 0.33 0.49 0.3 0.58 0.62 0.35
kNN(k = 50) downSample kNN(k = 50) downCluster kNN(k = 50) upSample kNN(k = 50) SMOTE kNN(k = 50) upCluster kNN(k = 50) Weighted kNN(k = 50)
Precision Sensitivity
0.55 0.55 0.52 0.52 0.33 0.7 0.56 0.47 0.36 0.61 0.54 0.49
SVM_linear downSample SVM_linear downCluster SVM_linear upSample SVM_linear SMOTE SVM_linear upCluster SVM_linear Weighted SVM_linear
Precision Sensitivity
0.68 0.31 0.45 0.67 0.41 0.69 0.47 0.62 0.43 0.54 0.51 0.52 0.49 0.55
SVM_radial downSample SVM_radial downCluster SVM_radial upSample SVM_radial SMOTE SVM_radial upCluster SVM_radial Weighted SVM_radial
Precision Sensitivity
0.43 0.38 0.28 0.59 0.3 0.59 0.18 0.45 0.29 0.56 0.34 0.62 0.23 0.52
SVM_sigmoid downSample SVM_sigmoid downCluster SVM_sigmoid upSample SVM_sigmoid SMOTE SVM_sigmoid upCluster SVM_sigmoid Weighted SVM_sigmoid



Слика 4.1 : Приказ евалуационих метрика по моделима.



Слика 4.2 : Приказ модела са највећим вредностима одговарајуће метрике.

Евалуација модела приказана је у табели 4.2. Можемо приметити да је пре примене метода одзив мањинске категорије био знатно лошији, а да је тачност класификације међу инстанцама већинске категорије (*specificity*) била велика, што потврђује проблематику изложену у овом раду. Велике вредности тачности и *AUC* потврђују да ове метрике нису поуздани показатељи квалитета модела када је присутна небалансираност, о чему је и раније било речи.

На слици 4.1 дата је визуелна интерпретација резултата за сваки модел и његова унапређења. Видимо да су скоро све методе довеле до повећања одзива, али и до благог смањења прецизности. Изузетак представљају стабло одлучивања са Хелингеровим растојањем као критеријумом гранања, код ког промена величине скупа за тренирање није значајно утицала на препознавање обеју категорија, као и алгоритми *UnderBagging* и *RUSBoost*.

На слици 4.2 приказано је првих 20 модела са највећим вредностима прецизности, одзива и $F1$ мере. Највећу прецизност имали су модели пре примене метода, што може бити последица давања малог броја предикција мањинске категорије. Највећи одзив је примећен након примене метода реузорковања, а највећа $F1$ мера код модела који су половично препознали обе категорије.

4.2 Вишекласна класификација

Glass [30] је скуп података о саставу стакла и његовом типу. Састоји се искључиво од нумеричких атрибута и циљне променљиве која може узети једну од 6 могућих категорија (табела 4.3).

Назив променљиве	Тип	Опис
RI	нумерички	индекс преламања
Na	нумерички	проценат натријума
Mg	нумерички	проценат магнезијума
Al	нумерички	проценат алуминијума
Si	нумерички	проценат силицијума
K	нумерички	проценат калијума
Ca	нумерички	проценат калцијума
Ba	нумерички	проценат баријума
Fe	нумерички	проценат гвожђа
Type	категорички	зависна променљива: 1 = building windows float processed; 2 = building windows float processed; 3 = building windows non float processed; 5 = containers; 6 = tableware; 7 = headlamps;

Табела 4.3 : Опис скупа *Glass*

Расподела категорија зависне променљиве је следећа:

тип 1	тип 2	тип 3	тип 5	тип 6	тип 7
33%	36%	8%	6%	4%	14%

па ћемо категорије 1 и 2 сматрати већинским, а остале мањинским. Као и у претходном примеру целокупни скуп података је подељен на скуп за тренирање (80%) и скуп за тестирање (20%). Евалуација модела је извршена за сваку категорију спрам осталих. Резултати се могу видети у табели 4.4.

Због мале заступљености категорија $3, 5$ и 6 у скупу за тестирање не можемо поуздано тумачити њихове резултате. Категорија 7 је успешно препозната пре примене метода, чак успешније и од већинских катеогрија 1 и 2 које модели логистичке регресије и kNN нису препознали, док су је поједине методе реузорковања и тежинско галсање у моделу kNN у потпуности замаскирале. Овакви резултати могу указивати на груписање категорије 7 у простору предиктора, као и на то да је за испољавање утицаја небалансираности у вишекласном случају потребан скуп података великог обима.

	Type 1 vs others			Type 2 vs others			Type 3 vs others		
	F1	Precision	Sensitivity	F1	Precision	Sensitivity	F1	Precision	Sensitivity
LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SVM_linear	0.56	0.636	0.5	0.5	0.471	0.533	0.25	0.2	0.333
SVM_radial	0.714	0.714	0.714	0.706	0.632	0.8	*NA*	0	0
SVM_sigmoid	0.621	0.6	0.643	0.581	0.562	0.6	*NA*	*NA*	0
SVM_poly	0.5	0.6	0.429	0.485	0.444	0.533	0.25	0.2	0.333
CART	0.72	0.818	0.643	0.688	0.647	0.733	0.286	0.25	0.333
C4.5	0.815	0.846	0.786	0.667	0.611	0.733	*NA*	*NA*	0
downSample_LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_SVM_linear	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_SVM_radial	*NA*	*NA*	0	0.545	0.375	1	*NA*	*NA*	0
downSample_SVM_sigmoid	*NA*	0	0	*NA*	0	0	*NA*	*NA*	0
downSample_SVM_poly	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_CART	*NA*	*NA*	0	*NA*	*NA*	0	0.158	0.086	1
downSample_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_SVM_linear	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_SVM_radial	*NA*	*NA*	0	0.545	0.375	1	*NA*	*NA*	0
downClusterMulti_SVM_sigmoid	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_SVM_poly	0.519	0.35	1	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_CART	*NA*	*NA*	0	*NA*	*NA*	0	0.158	0.086	1
downClusterMulti_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_SVM_linear	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_SVM_radial	*NA*	*NA*	0	0.545	0.375	1	*NA*	*NA*	0
upSample_SVM_sigmoid	*NA*	*NA*	0	0.541	0.455	0.667	*NA*	*NA*	0
upSample_SVM_poly	*NA*	*NA*	0	0.558	0.429	0.8	0.133	0.083	0.333
upSample_CART	0.609	0.438	1	*NA*	*NA*	0	*NA*	*NA*	0
upSample_C4.5	0.24	0.273	0.214	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_SVM_linear	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_SVM_radial	*NA*	*NA*	0	0.545	0.375	1	*NA*	*NA*	0
SMOTEMulti_SVM_sigmoid	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_SVM_poly	*NA*	*NA*	0	0.549	0.389	0.933	0.286	0.25	0.333
SMOTEMulti_CART	0.609	0.438	1	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_C4.5	0.622	0.452	1	0.167	0.222	0.133	*NA*	*NA*	0
upClusterMulti_LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_SVM_linear	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_SVM_radial	*NA*	*NA*	0	0.545	0.375	1	*NA*	*NA*	0
upClusterMulti_SVM_sigmoid	0.5	0.385	0.714	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_SVM_poly	0.562	0.5	0.643	*NA*	*NA*	0	0.08	0.045	0.333
upClusterMulti_CART	*NA*	*NA*	0	*NA*	*NA*	0	0.14	0.075	1
upClusterMulti_C4.5	*NA*	*NA*	0	0.545	0.375	1	*NA*	*NA*	0
Weighted_LR	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
Weighted_SVM_linear	0.64	0.727	0.571	0.552	0.571	0.533	0.2	0.143	0.333
Weighted_SVM_radial	0.769	0.833	0.714	0.722	0.619	0.867	*NA*	0	0
Weighted_SVM_sigmoid	0.444	0.462	0.429	0.148	0.167	0.133	0.167	0.111	0.333
Weighted_SVM_poly	0.759	0.733	0.786	0.615	0.727	0.533	*NA*	0	0
Weighted_kNN(k = 5)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
Weighted_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
Weighted_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
RF	*NA*	*NA*	0	0.56	0.4	0.933	*NA*	*NA*	0
Balanced RF	0.622	0.452	1	*NA*	*NA*	0	*NA*	*NA*	0
AdaBoost	0.774	0.706	0.857	0.593	0.667	0.533	*NA*	0	0

	Type 5 vs others			Type 6 vs others			Type 7 vs others		
	F1	Precision	Sensitivity	F1	Precision	Sensitivity	F1	Precision	Sensitivity
LR	*NA*	*NA*	0	0.059	0.03	1	0.667	0.571	0.8
kNN(k = 5)	*NA*	*NA*	0	*NA*	0	0	0.24	0.15	0.6
kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
SVM_linear	0.4	0.333	0.5	*NA*	*NA*	0	0.889	1	0.8
SVM_radial	0.667	1	0.5	*NA*	*NA*	0	0.889	1	0.8
SVM_sigmoid	*NA*	*NA*	0	*NA*	*NA*	0	0.714	0.556	1
SVM_poly	0.4	0.333	0.5	*NA*	*NA*	0	0.889	1	0.8
CART	0.8	0.667	1	*NA*	*NA*	0	0.8	0.8	0.8
C4.5	0.8	0.667	1	1	1	1	0.8	0.8	0.8
downSample_LR	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
downSample_kNN(k = 5)	*NA*	0	0	*NA*	0	0	*NA*	*NA*	0
downSample_kNN(k = 10)	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
downSample_kNN(k = 20)	*NA*	*NA*	0	0.286	0.167	1	0.051	0.029	0.2
downSample_SVM_linear	*NA*	0	0	0.061	0.031	1	0.667	0.571	0.8
downSample_SVM_radial	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_SVM_sigmoid	*NA*	0	0	*NA*	*NA*	0	*NA*	*NA*	0
downSample_SVM_poly	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
downSample_CART	*NA*	*NA*	0	*NA*	*NA*	0	0.8	0.8	0.8
downSample_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
downClusterMulti_LR	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
downClusterMulti_kNN(k = 5)	*NA*	0	0	*NA*	0	0	*NA*	*NA*	0
downClusterMulti_kNN(k = 10)	*NA*	0	0	*NA*	*NA*	0	0.227	0.128	1
downClusterMulti_kNN(k = 20)	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
downClusterMulti_SVM_linear	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
downClusterMulti_SVM_radial	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_SVM_sigmoid	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
downClusterMulti_SVM_poly	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
downClusterMulti_CART	*NA*	*NA*	0	*NA*	*NA*	0	0.8	0.8	0.8
downClusterMulti_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
upSample_LR	*NA*	*NA*	0	0.061	0.031	1	0.615	0.5	0.8
upSample_kNN(k = 5)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
upSample_kNN(k = 10)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
upSample_kNN(k = 20)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
upSample_SVM_linear	*NA*	*NA*	0	0.05	0.026	1	*NA*	0	0
upSample_SVM_radial	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_SVM_sigmoid	*NA*	*NA*	0	0.105	0.056	1	*NA*	*NA*	0
upSample_SVM_poly	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upSample_CART	0.2	0.125	0.5	*NA*	*NA*	0	*NA*	*NA*	0
upSample_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	0.059	0.034	0.2
SMOTEMulti_LR	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
SMOTEMulti_kNN(k = 5)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
SMOTEMulti_kNN(k = 10)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
SMOTEMulti_kNN(k = 20)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
SMOTEMulti_SVM_linear	*NA*	*NA*	0	0.056	0.029	1	0.8	0.8	0.8
SMOTEMulti_SVM_radial	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_SVM_sigmoid	*NA*	0	0	0.4	0.25	1	0.205	0.118	0.8
SMOTEMulti_SVM_poly	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_CART	0.2	0.125	0.5	*NA*	*NA*	0	*NA*	*NA*	0
SMOTEMulti_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_LR	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
upClusterMulti_kNN(k = 5)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
upClusterMulti_kNN(k = 10)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
upClusterMulti_kNN(k = 20)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
upClusterMulti_SVM_linear	*NA*	0	0	0.051	0.026	1	*NA*	*NA*	0
upClusterMulti_SVM_radial	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_SVM_sigmoid	0.25	0.167	0.5	0.222	0.125	1	*NA*	*NA*	0
upClusterMulti_SVM_poly	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_CART	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
upClusterMulti_C4.5	*NA*	*NA*	0	*NA*	*NA*	0	*NA*	*NA*	0
Weighted_LR	*NA*	*NA*	0	*NA*	*NA*	0	0.222	0.125	1
Weighted_SVM_linear	0.4	0.333	0.5	1	1	1	0.889	1	0.8
Weighted_SVM_radial	0.667	1	0.5	*NA*	*NA*	0	0.889	1	0.8
Weighted_SVM_sigmoid	*NA*	0	0	0.667	0.5	1	0.333	1	0.2
Weighted_SVM_poly	0.667	0.5	1	1	1	1	0.889	1	0.8
Weighted_kNN(k = 5)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
Weighted_kNN(k = 10)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
Weighted_kNN(k = 20)	*NA*	*NA*	0	0.049	0.025	1	*NA*	*NA*	0
RF	*NA*	*NA*	0	*NA*	*NA*	0	0.8	0.8	0.8
Balanced RF	0.667	0.5	1	*NA*	*NA*	0	0.8	0.8	0.8
AdaBoost	0.333	0.25	0.5	1	1	1	0.8	0.8	0.8

Табела 4.4 : Евалуација модела.

Поглавље 5

Закључак

У овом раду представљен је проблем класификације небалансираних података. Под небалансираношћу података подразумевали смо међукласну небалансираност, која се односи на неравномерну расподелу зависне категоричке променљиве на скупу за тренирање.

Изложене су хипотезе о узроцима слабијег распознавања најмање заступљених, односно мањинских категорија и предложено је како приступити евалуацији модела обучаваног у присуству небалансираности.

У другом поглављу дат је преглед модела који показују осетљивост на небалансираност података. Описана је њихова архитектура, процес обучавања и класификационо правило које се користи у давању предикција. Истакли смо да модели при чијем обучавању функција губитака придаје подједнак значај свим категоријама слабије уче репрезентацију оних најмање заступљених, што јесте један од узрочника њиховог нераспознавања. У такве моделе спадају логистичка регресија и модел потпорних вектора. Даље, истакли смо како превелико k у моделу k најближих суседа има за последицу фаворизовање најзаступљенијих категорија и показали како се Флаховим моделом може проценити утицај небалансираности на поузданост критеријума гранања чвора у стаблу одлучивања.

У трећем поглављу показали смо како поједине методе за селекцију предиктора такође могу бити под утицајем небалансираности и дали пример методе која показује робусност у таквој ситуацији, а то је Хелингерово растојање. Потом су изложене 3 велике групе метода за превазилажење проблема: методе реузорковања, методе осетљиве на цену грешке и методе засноване на ансамблима. За сваку групу метода описане су њихове предности и недостаци.

Предложене методе тестиране су на примерима бинарне и вишекласне

класификације. У бинарној класификацији већином је дошло до побољшања одзива у односу на почетни модел, али и до благог смањења прецизности. Такође видимо и да на стабло одлучивања с Хелингеровим растојањем као критеријумом гранања није значајно утицао однос категорија.

Даље унапређење метода подразумевало би спречавање погоршања прецизности, односно нераспознавања других категорија зарад побољшања одзива мањинских. Један начин да се то постигне јесте другачији одабир тежина у методама осетљивим на цену грешке и обима новодобијеног скупа за тренирање у методама реузорковања. [31] предлажу класификациони алгоритам којим се независно за сваку од категорија конструише простор вредности предиктора који јој одговара, чиме се побољшава учење репрезентације свих категорија у односу на моделе у чијем обучавању категорије учествују симултано. Још један правац даљег истраживања може бити развој метода за рад са подацима који нису представљени у векторском запису, попут слика, звука, текста и слично. Иако би се методе реузорковања могле применити и у тој ситуацији, поставља се питање њихове учинковитости узевши у обзир недостатке описане у овом раду.

Библиографија

[1] М. Достанић Д. Јоцић, М. Арсеновић. Теорија мере, функционална анализа, теорија оператора. 2012.

[2] М. Николић А. Зечевић. Машинско учење. 2019.

[3] M. J. Silvapulle. On the existence of maximum likelihood estimators for the binomial response models. *Journal of the royal statistical society series b-methodological*, 43:310–313, 1981.

[4] Chuong B. Do. Convex optimization overview (cnt'd). 2009.

[5] D. Cheng and M. Wu. A novel classifier - weighted features cost-sensitive svm. pages 598–603, 2016.

[6] L. Rokach and O. Maimon. *Decision Trees*, volume 6, pages 165–192. 01 2005.

[7] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[8] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[9] S. L. Salzberg. Book review: C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16:235–240, 1994.

[10] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[11] P. Flach. The geometry of roc space: Understanding machine learning metrics through roc isometrics. volume 1, pages 194–201, 01 2003.

[12] D. Cieslak and N. Chawla. Learning decision trees for unbalanced data. pages 241–256, 09 2008.

[13] L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan. Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105:3–11, 2013.

[14] H. He and Y. Ma. Imbalanced learning: Foundations, algorithms, and applications. *Imbalanced Learning: Foundations, Algorithms, and Applications*, 06 2013.

[15] S. J. Yen and Y. S. Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36:5718–5727, 01 2006.

[16] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, 1996.

[17] L.Pradeep and M. Sowjanya. Multi-density based incremental clustering. *International Journal of Computer Applications*, 116:6–9, 04 2015.

[18] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 06 2002.

[19] D. T. Jo and N. Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explorations*, 6:40–49, 06 2004.

[20] G. King and L. Zeng. Logistic regression in rare events data. *Political Analysis*, 9, 09 2002.

[21] K. Hechenbichler and K. Schliep. Weighted k-nearest-neighbor techniques and ordinal classification. *discussion paper*, 399, 01 2004.

[22] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001.

[23] C. Chen and L. Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 01 2004.

[24] F. Hanifah, H. Wijayanto, and A. Kurnia. Smote bagging algorithm for imbalanced dataset in logistic regression analysis (case: Credit of bank x). 9:6857–6865, 01 2015.

[25] R. Barandela, R. Valdovinos, and J. Sánchez. New applications of ensembles of classifiers. *Pattern Analysis Applications*, 6:245–256, 01 2003.

[26] M. Galar, A. Fernández, E. Barrenechea, H. Sola, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42:463 – 484, 07 2012.

[27] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer. Smoteboost: Improving prediction of the minority class in boosting. volume 2838, pages 107–119, 01 2003.

[28] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, and A. Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40:185 – 197, 02 2010.

[29] I. C. Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2016. DOI: https://doi.org/10.24432/C55S3H.

[30] B. German. Glass Identification. UCI Machine Learning Repository, 1987. DOI: https://doi.org/10.24432/C5WW2P.

[31] A. Manukyan and E. Ceyhan. Classification of imbalanced data with a geometric digraph family. *Journal of Machine Learning Research*, 17, 10 2016.

Биографија

Нађа Обреновић је рођена 30. октобра 1997. године у Краљеву. У свом родном граду завршила је основну школу и Гимназију. Основне студије уписала је 2016. године на Математичком факултету Универзитета у Београду, смер Статистика, актуарска и финансијска математика, које је завршила 2020. године са просечном оценом 8.95. Од 2023. године запослена је као статистички програмер у компанији Парексел. Интересује се за математичку теорију која је темељ машинског и статистичког учења.