

UNIVERZITET U BEOGRADU  
MATEMATIČKI FAKULTET



Vladimir Luković

PRIMENA METODA POTPORNIH VEKTORA  
ZA VIŠEKLASNU KLASIFIKACIJU  
BANKARSKIH TRANSAKCIJA

master rad

Beograd, 2024.

**Mentor:**

dr Bojana MILOŠEVIĆ, vanredni profesor  
Univerzitet u Beogradu, Matematički fakultet

**Članovi komisije:**

dr Mladen NIKOLIĆ, vanredni profesor  
Univerzitet u Beogradu, Matematički fakultet

dr Milan JOVANOVIĆ, docent  
Univerzitet u Beogradu, Matematički fakultet

**Datum odbrane:** \_\_\_\_\_

*Mami, tati i Milici*

**Naslov master rada:** Primena metoda potpornih vektora za višeklasnu klasifikaciju bankarskih transakcija

**Ključne reči:** metod potpornih vektora, statističko učenje, binarna klasifikacija, višeklasna klasifikacija

# Sadržaj

1	Uvod	1
2	Metod potpornih vektora u slučaju binarne klasifikacije	2
3	Metod potpornih vektora u višeklasnom slučaju	9
4	Primena	25
5	Zaključak	47
	Bibliografija	48

# Glava 1

## Uvod

Bankarska transakcija je bilo koja promena na stanju računa klijenta. Svako plaćanje karticama, slanje novca ili podizanje na bankomatu spada u bankarsku transakciju. U današnje vreme raste udeo plaćanja karticama u odnosu na gotovinu. Po podacima banke federalnih rezervi San Franciska, 2022. godine, gotovina je predstavljala samo 18% plaćanja [5]. Imajući u vidu sve veći broj bankarskih transakcija nameće se potreba za njihovom ispravnom klasifikacijom iz više razloga. Ispravna klasifikacija bankarskih transakcija omogućava finansijskim institucijama da efikasno otkriju i spreče prevaru, da bolje upoznaju svoje klijente, da osiguraju usklađenost sa regulatornim zahtevima. Pored toga, uspešna klasifikacija transakcija pomaže u analizi ponašanja klijenata, omogućavajući personalizaciju usluga što povećava zadovoljstvo klijenata.

U ovom radu ispitaćemo mogućnost primene metoda potpornih vektora za višeklasnu klasifikaciju bankarskih transakcija, gde se klasa odnosi na kategoriju kojoj transakcija pripada, kao što je plaćanje, transfer novca i slično. Prvo ćemo predstaviti metod potpornih vektora u slučaju kada imamo samo dve klase. Nakon toga ćemo ispitati šest različitih načina da se metod potpornih vektora prenese na slučaj kada imamo više od dve klase. Predstavljeni algoritmi će biti primenjeni na bankarskim transakcijama u poglavlju 4, gde ćemo uporediti performanse modela.

## Glava 2

# Metod potpornih vektora u slučaju binarne klasifikacije

Metod potpornih vektora (eng. SVM) je algoritam nadgledanog statističkog učenja koji se koristi za klasifikaciju i regresiju. Prvobitno je razvijen od strane Vladimira Vapnika i njegovih saradnika u okviru teorije statističkog učenja [3]. SVM je zasnovan na ideji pronalaženja optimalne hiperravni u visokodimenzionalnom prostoru, koja maksimalno razdvaja dve klase podataka. Osnovni cilj SVM-a je da pronađe hiperravan koja razdvaja dve klase uz maksimalnu marginu, tj. razdaljinu između hiperravni i najbližih tačaka iz svake klase, koje nazivamo potporni vektori. (v. [2])

### Metod potpornih vektora sa tvrdom marginom

Za dati skup podataka  $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$ , gde je  $\mathbf{x}_i \in \mathbb{R}^N$  vektor ulaznih podataka, a  $y_i \in \{-1, 1\}$  klase, metod potpornih vektora pokušava da pronađe hiperravan koja razdvaja ove dve klase. Funkcija  $\phi$  predstavlja transformaciju ulaznog vektora  $\mathbf{x}$  u prostor viših dimenzija, a vektor  $\mathbf{w}$  je vektor normalan na hiperravan. Pošto definišemo slučaj sa tvrdom marginom, pretpostavljamo da su tačke trening skupa linearno razdvojive u prostoru  $\phi(x)$ . Hiperravan se može definisati kao:

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b. \quad (2.1)$$

Tražena hiperravan treba da zadovoljava:

$$\mathbf{w}^T \phi(\mathbf{x}_i) + b \geq C, \text{ za } y_i = 1,$$

$$\mathbf{w}^T \phi(\mathbf{x}_i) + b \leq -C, \text{ za } y_i = -1.$$

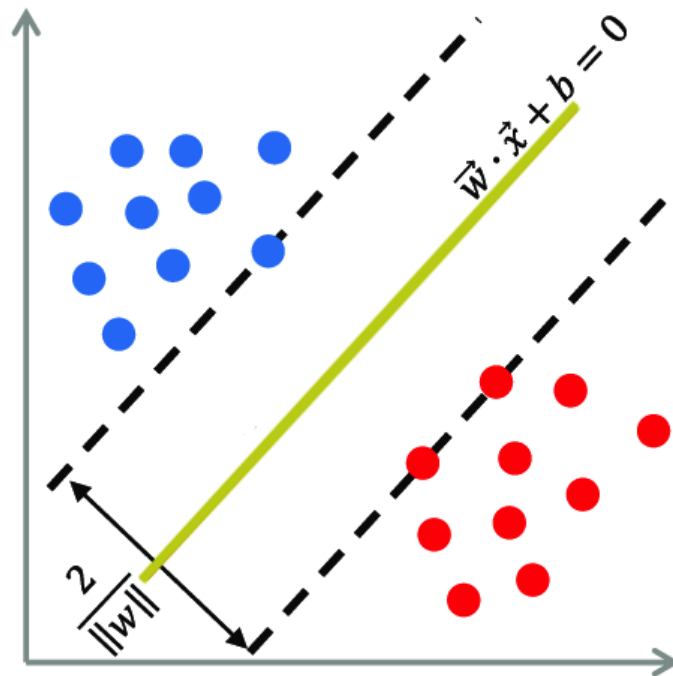
Deljenjem ovih izraza sa konstantom  $C$ , dobija se da je:

$$\mathbf{w}^T \phi(\mathbf{x}_i) + b \geq 1, \text{ za } y_i = 1,$$

$$\mathbf{w}^T \phi(\mathbf{x}_i) + b \leq -1, \text{ za } y_i = -1.$$

Ova dva uslova možemo objediniti

$$y_i[\mathbf{w}^T \phi(\mathbf{x}_i) + b] \geq 1, i = 1 \dots l. \quad (2.2)$$



Slika 2.1: Zelena linija označava optimalnu hiperravan koja razdvaja klase, dok isprekidane linije predstavljaju odgovarajuće maksimalne margine različitih klasa. Tačke na marginama su potporni vektori.

Znamo da je  $y_i[\mathbf{w}^T \phi(\mathbf{x}_i) + b] = 1$ , za tačke najbliže optimalnoj hiperravni.

Tražena hiperravan je ona koja daje maksimalnu marginu, to jest maksimalno rastojanje između potpornih vektora. Tražimo  $\mathbf{w}$  koje daje maksimalno rastojanje  $\frac{2}{\|\mathbf{w}\|^2}$ . Radi boljeg razumevanja, pogledati sliku 2.1. Optimalna hiperravan je ona koja zadovoljava uslov 2.2 i maksimizuje rastojanje:  $\frac{2}{\|\mathbf{w}\|^2}$ . Traženu hiperravan dobijamo rešavanjem sledećeg optimizacionog problema:

$$\arg \max_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|^2} \min_i \left[ y_i (\mathbf{w}^T \phi(x_i) + b) \right],$$



---

uz važenje uslova 2.2. Ideja je da maksimizujemo najkraću distancu tačke od hiperravni. Sa problema maksimizacije  $\frac{2}{\|\mathbf{w}\|^2}$ , možemo preći na problem minimizacije  $\|\mathbf{w}\|^2$ .

Uvedimo Lagranžove množioce  $(\alpha_1, \alpha_2, \dots, \alpha_l)$ , da bismo originalni problem sa strogim uslovima transformisali u problem koji se može lakše rešiti, uz uslov nenegativnosti  $\alpha_i \geq 0$ ,  $i = 1, \dots, l$ . Formiramo Lagranžovu funkciju:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i [y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) - 1]. \quad (2.3)$$

Kako bismo pronašli minimum Lagranžove funkcije, potrebno je izračunati parcijalne izvode u odnosu na  $\mathbf{w}^T$  i  $b$ :

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i) = 0, \\ \frac{\partial L}{\partial b} &= \sum_{i=1}^l \alpha_i y_i = 0. \end{aligned} \quad (2.4)$$

Kada primenimo distributivnost na 2.3, dobijamo:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i y_i \mathbf{w}^T \phi(\mathbf{x}_i) - \sum_{i=1}^l \alpha_i y_i \mathbf{w}^T b + \sum_{i=1}^l \alpha_i.$$

Sada ubacujemo 2.4 i dobijamo:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \|\mathbf{w}\|^2 + \sum_{i=1}^l \alpha_i.$$

Ovo je takozvani dualni Lagranžov problem:

$$L^*(\boldsymbol{\alpha}) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \phi(\mathbf{x}_j), \quad (2.5)$$

$$\text{uz uslove } \alpha_i \geq 0 \text{ i } \sum_{i=1}^l \alpha_i y_i = 0.$$

Ovo je problem kvadratnog programiranja i rešava se sekvencijalnom minimalnom optimizacijom (SMO) [13]. Rešenje treba da zadovoljava Karuš-Kun-Taker uslove:

$$\begin{aligned} \alpha_i &\geq 0, \\ y_i y(x_i) - 1 &\geq 0, \\ \alpha_i (y_i y(x_i) - 1) &= 0. \end{aligned}$$

---

Dakle, za svaku tačku podataka važi da je  $\alpha_i = 0$  ili  $y_i y(x_i) = 1$ . Svaka tačka podataka za koju je  $\alpha_i = 0$  neće se pojaviti u sumi i stoga ne igra nikakvu ulogu u predikcijama za nove podatke. Potporni vektori zadovoljavaju  $y_i y(x_i) = 1$  i samim tim oni leže na maksimalnim marginama odgovarajućih klasa. Ova osobina je ključna za praktičnu primenljivost metoda potpornih vektora. Kada se model obuči, značajan deo tačaka podataka može se odbaciti i zadržati samo potporni vektori.

Klasifikovanje novih tačaka na osnovu 2.1, može da se iskaže u odnosu na Lagranžove parametre, koristeći uslov 2.4:

$$y(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i) \phi(\mathbf{x}) + b. \quad (2.6)$$

Zahvaljujući osobini potpornih vektora  $y_i [\mathbf{w}^T \mathbf{x}_i + b] = 1$  i uslovu 2.4, dobijamo jednakost:

$$y_i \left( \sum_{j \in S} \alpha_j y_j \phi(\mathbf{x}_j) \phi(\mathbf{x}_i) + b \right) = 1, \quad (2.7)$$

gde  $S$  označava skup indeksa potpornih vektora. Rešavanjem problema kvadratnog programiranja 2.5 i na osnovu 2.7, parametar  $b$  možemo dobiti na sledeći način:

$$\hat{b} = \frac{1}{N_S} \sum_{i \in S} \left( y_i - \sum_{j \in S} \alpha_j y_j \phi(\mathbf{x}_j) \phi(\mathbf{x}_i) \right),$$

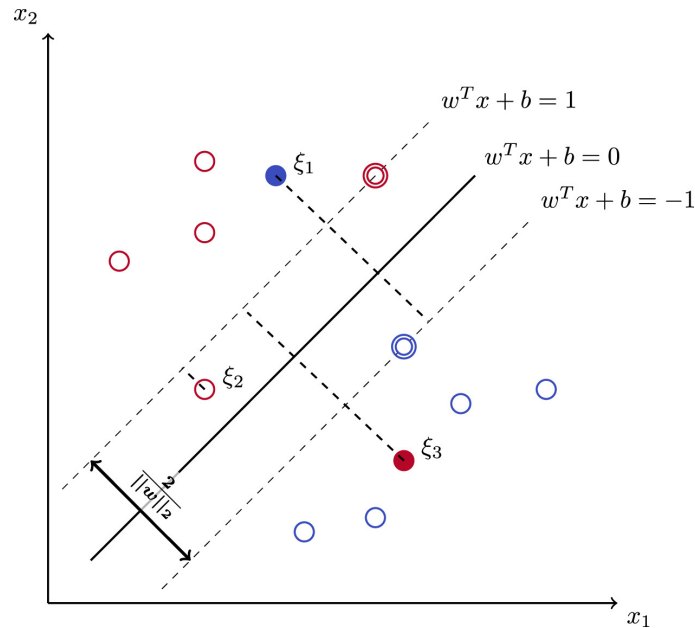
pri čemu  $N_S$  predstavlja ukupan broj potpornih vektora.

## Metod potpornih vektora sa mekom marginom

Do sada smo pretpostavili da su tačke trening skupa linearno razdvojive u prostoru  $\phi(x)$ . Model potpornih vektora će omogućiti tačno razdvajanje trening podataka u originalnom ulaznom prostoru  $x$ , iako će odgovarajuća granica odlučivanja biti nelinearna. Međutim, u praksi se klase mogu preklapati, što znači da tačno razdvajanje trening podataka možda nije uvek moguće. Potrebno je prilagoditi model potpornih vektora tako da omogući da se neki trening podaci klasifikuju netačno. Za naredni problem optimizacije, korisno je da ta kazna bude linearno zavisna od rastojanja. Da bismo to postigli, uvodimo pomoćne promenljive  $\xi_i \geq 0$ , gde je  $i = 1, \dots, l$ , sa jednom promenljivom za svaku tačku trening skupa.

Ove promenljive se definišu sa  $\xi_i = 0$  za podatke koji se nalaze unutar ili na ispravnoj granici margine, dok je  $\xi_i = |y_i - y(x_i)|$  za ostale tačke. Proširujemo uslov 2.2 dodavanjem mekog pojasa. Tada tražena hiperravan treba da zadovoljava:

$$y_i y(x_i) \geq 1 - \xi_i, \quad i = 1, \dots, l. \quad (2.8)$$



Slika 2.2: Podaci za koje je  $\xi_i = 0$  su ispravno klasifikovani i nalaze se na marginama ili na ispravnoj strani margine. Oni za koje važi  $0 < \xi_i < 1$  su unutar margine, ali su pravilno klasifikovani, jer su na ispravnoj strani granice odlučivanja (primer  $\xi_2$ ). Ako se podatak nalazi na granici odlučivanja  $y(x_i) = 0$  onda je  $\xi_i = 1$ . Podaci za koje važi  $\xi_i > 1$  nalaze se na netačnoj strani hiperravni odlučivanja i klasifikovani su netačno (primer  $\xi_1$  i  $\xi_3$ ). Slika preuzeta iz [7].

Ovaj pristup ublažava strogu granicu kako bi se postigla fleksibilnija margina i omogućilo da se neki podaci u trening skupu klasifikuju nepravilno. Važno je napomenuti da, iako pomoćne promenljive omogućavaju preklapanje distribucija klasa, ovaj okvir i dalje je osetljiv na ekstremne vrednosti jer kazna za netačnu klasifikaciju linearno raste sa  $\xi$ .

Naš cilj je sada da maksimiziramo marginu, dok kažnjavamo tačke koje se nalaze na pogrešnoj strani margine. Zbog toga minimiziramo izraz:

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i, \quad (2.9)$$

gde je parametar  $C > 0$  regularizacioni parametar. Parametar  $C$  kontroliše ravnotežu između minimiziranja grešaka u obučavanju i složenosti modela. U granici  $C \rightarrow \infty$ , vraćamo se na raniji slučaj linearno razdvojitih klasa.

Sada želimo da minimiziramo 2.9 pomoću uslova 2.8 zajedno sa  $\xi_i \geq 0$ . Odgovarajući lagranžijan je dat izrazom:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i (y_i y(x_i) - 1 + \xi_i) - \sum_{i=1}^l \mu_i \xi_i,$$

gde su  $\alpha_i \geq 0$  i  $\mu_i \geq 0$  Lagranžovi množiocci. Odgovarajući skup KKT uslova za  $i = 1, \dots, l$  je dat kao:

$$\begin{aligned} \alpha_i &\geq 0, \\ y_i y(x_i) - 1 + \xi_i &\geq 0, \\ \alpha_i (y_i y(x_i) - 1 + \xi_i) &= 0, \\ \mu_i &\geq 0, \\ \xi_i &\geq 0, \\ \mu_i \xi_i &= 0. \end{aligned} \tag{2.10}$$

Sada optimizujemo  $\mathbf{w}$ ,  $b$  i  $\xi_n$  koristeći 2.1 definiciju za  $y(x)$ :

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \phi(x_i), \tag{2.11}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^l \alpha_i y_i = 0, \tag{2.12}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i = C - \mu_i. \tag{2.13}$$

Koristeći ove rezultate da eliminišemo  $\mathbf{w}$ ,  $b$  i  $\xi_i$  iz Lagranžijana, dobijamo dualni Lagranžijan u obliku:

$$L^*(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \phi(\mathbf{x}_j), \tag{2.14}$$

koji je identičan kao u slučaju kada su podaci separabilni, osim što su uslovi nešto drugačiji. Da bismo identifikovali ove uslove, primetimo da je potrebno  $\alpha_i \geq 0$  jer su to Lagranžovi množiocci. Pored toga 2.13 zajedno sa  $\mu_i \geq 0$  implicira da je  $\alpha_i \leq C$ .

---

Prema tome moramo da minimiziramo 2.14 sa obzirom na dualne promenljive  $\alpha_i$  i pod uslovom da:

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, l, \quad (2.15)$$

$$\sum_{i=1}^l \alpha_i y_i = 0.$$

Uslovi 2.15 se nazivaju ograničenjima okvira (eng. box constraints). Ovo predstavlja ponovo problem kvadratnog programiranja. Ako zamenimo 2.11 u 2.1, vidimo da se predikcije za nove podatke vrše ponovo koristeći 2.6.

Sada možemo da tumačimo rešenje. Podaci gde je  $\alpha_i = 0$ , ne doprinose prediktivnom modelu jer se iz 2.6 zaključuje da je odgovarajući član ukupne sume jednak nula. Podaci za koje je  $\alpha_i > 0$  čine potporne vektore. Oni moraju zadovoljiti:

$$y_i y(x_i) = 1 - \xi_i.$$

Ako je  $\alpha_i < C$ , onda 2.13 implicira  $\mu_i > 0$ , što na osnovu 2.10 zahteva da je  $\xi_i = 0$  i znači da takve tačke leže na margini. Tačke gde je  $\alpha_i = C$  mogu biti ispravno ili pogrešno klasifikovane, zavisno od vrednosti  $\xi_i$ .

Za određivanje parametra  $b$ , koristimo podatke gde je  $0 < \alpha_i < C$  za koje važi  $\xi_i = 0$  tako da  $y_i y(x_i) = 1$  i dobijamo:

$$\hat{b} = \frac{1}{N_M} \sum_{i \in M} \left( y_i - \sum_{j \in S} \alpha_j y_j \phi(\mathbf{x}_j) \phi(\mathbf{x}_i) \right),$$

gde  $M$  označava skup indeksa podataka koji imaju  $0 < \alpha_i < C$ ,  $S$  označava skup indeksa potpornih vektora, a  $N_M$  ukupan broj članova u skupu  $M$ .

## Glava 3

# Metod potpornih vektora u višeklasnom slučaju

U prethodnom poglavlju govorili smo o metodu potpornih vektora u slučaju binarne klasifikacije. Metod potpornih vektora inicijalno je razvijen za slučaj dve klase. Kako su primene metoda na samo dve klase ograničene, javila se potreba da se primena metoda proširi na slučaj kada imamo više od dve klase. U ovom poglavlju ćemo da pređemo sa binarnog slučaja na višeklasni slučaj, pomoću različitih pristupa (v. [8] [12]). Postoje dva načina za klasifikaciju sa više klasa pomoću SVM-a. Prvi pristup podrazumeva konstruisanje i kombinovanje više binarnih klasifikatora, a obrađene metode uključuju: jedan protiv svih, jedan na jedan, DAGSVM i kodiranje izlaznih kodova za korekciju grešaka. Drugi pristup direktno uključuje sve podatke u jednoj optimizacionoj formulaciji, pri čemu smo obradili metode Vestona i Votkinsa, kao i Kramera i Singera.

### Jedan protiv svih

Prvi pristup koji je predložio Vapnik ogleda se u tome da se jedna klasa poredi sa svima ostalima (eng. *one against the rest, one against all*). Ovaj pristup konstruiše  $k$  SVM modela, gde je  $k$  broj klasa. Za bilo koju klasu  $m$ , obučavamo SVM model tako što svim elementima klase  $m$  dodeljujemo 1, a svim elementima van klase  $m$  dodeljujemo  $-1$ . Time smo obučili binarni klasifikator koji razdvaja klasu  $m$  od svih ostalih. Postupak ponavljamo za svaku od  $k$  klasa. Konačno rešenje za konkretan ulazni vektor je ona klasa koja daje najveću marginu.

---

Dakle, za datih  $l$  trening podataka  $(x_1, y_1), \dots, (x_l, y_l)$ , gde je  $x_i \in \mathbb{R}^N, i = 1, \dots, l$  i  $y_i \in \{1, \dots, k\}$  je klasa od  $x_i$ ,  $m$ -ti SVM rešava sledeći problem:

$$\min_{w_m, b_m, \xi^m} \frac{1}{2} (w_m)^T w_m + C \sum_{i=1}^l \xi_i^m, \quad (3.1)$$

pod sledećim uslovima:

$$\begin{aligned} (w_m)^T \phi(x_i) + b_m &\geq 1 - \xi_i^m, & \text{ako } y_i = m, \\ (w_m)^T \phi(x_i) + b_m &\leq -1 + \xi_i^m, & \text{ako } y_i \neq m, \\ \xi_i^m &\geq 0, & i = 1, \dots, l, \end{aligned}$$

gde su trening podaci  $x_i$  mapirani u prostor viših dimenzija funkcijom  $\phi$  i  $C$  je regulizacioni parametar.

Želimo da nađemo najveću marginu među klasama. Sa problema maksimizacije  $\frac{2}{\|w_m\|}$  prelazimo na problem minimizacije  $\frac{1}{2} (w_m)^T w_m$ . Kada podaci nisu linearno razdvojni, postoji element  $C \sum_{i=1}^l \xi_i^m$  koji kontroliše koliko značaja se pridaje greškama.

Nakon rešavanja 3.1, dobijamo  $k$  funkcija odlučivanja:

$$\begin{aligned} (w_1)^T \phi(x) + b_1, \\ \vdots \\ (w_k)^T \phi(x) + b_k. \end{aligned}$$

Funkcija odlučivanja za svaku klasu  $m$  je definisana kao  $(w_m)^T \phi(x) + b_m$ , gde veći izlaz označava veću pripadnost toj klasi. Kažemo da je  $x$  u klasi koja ima najveću vrednost funkcije odlučivanja:

$$\text{klasa od } x \equiv \arg \max_{m=1, \dots, k} ((w_m)^T \phi(x) + b_m).$$

Time smo sa višeklasne klasifikacije prešli na binarnu klasifikaciju. Dakle, rešavamo  $k$  binarnih klasifikacija.

Prednost ovog pristupa je što omogućava kreiranje binarnog klasifikatora za svaku klasu, što daje fleksibilnost u razdvajanju klasa. Zahtevi za memorijom su generalno upravljivi sa savremenim računarima jer je potrebna  $O(l)$  memorija, gde je  $l$  broj klasa. Jedan od značajnih nedostataka je nebalansiranost klasa, jer se svaka klasa poredi sa svim ostalima, što često dovodi do ovog problema.

---

## Jedan na jedan

Da bi se prevazišli nedostaci pristupa jedan-protiv-svih, predložen je pristup jedan-na-jedan (eng. one against one). Ako imamo  $k$  klasa, ovaj metod konstruiše  $k(k-1)/2$  klasifikatora, gde je svaki treniran na podacima iz dve klase. Za trening podatke iz  $i$ -te i  $j$ -te klase, gde  $i, j \in \{1, \dots, k\}$ , rešavamo sledeći binarni klasifikacioni problem:

$$\min_{w_{ij}, b_{ij}, \xi^{ij}} \frac{1}{2} (w_{ij})^T w_{ij} + C \sum_t \xi_t^{ij}, \quad (3.2)$$

pod sledećim uslovima:

$$\begin{aligned} (w_{ij})^T \phi(x_t) + b_{ij} &\geq 1 - \xi_t^{ij}, & \text{ako } y_t = i, \\ (w_{ij})^T \phi(x_t) + b_{ij} &\leq -1 + \xi_t^{ij}, & \text{ako } y_t = j, \\ \xi_t^{ij} &\geq 0. \end{aligned}$$

Nakon obučavanja svih  $k(k-1)/2$  klasifikatora, odlučivanje se vrši na sledeći način: ako  $\text{sgn}(w_{ij})^T(\phi(x) + b_{ij})$  ukazuje da je  $x$  u  $i$ -toj klasi, tada se dodaje jedan glas za tu klasu. U suprotnom,  $j$ -ta klasa dobija jedan glas više. Zatim se predviđa da je  $x$  u grupi sa najvećim brojem glasova. Ova strategija glasanja naziva se „pobednik nosi sve” (eng. max wins). U slučaju da dve klase imaju jednake glasove, iako to možda nije optimalna strategija, biramo onu sa manjim indeksom. U praksi, rešavamo dualni problem sa istim brojem promenljivih kao što ima podataka u obe klase. Ako na primer svaka klasa u proseku ima  $l/k$  tačaka podataka, to znači da ćemo morati da rešimo  $k(k-1)/2$  problema kvadratnog programiranja, gde svaki od njih ima oko  $2l/k$  promenljivih.

Ovim pristupom kreira se više binarnih klasifikatora, ali svaki od njih se obučava na manjem skupu podataka nego kod 'jedan protiv svih' pristupa. Glavni nedostatak ovog pristupa je što se broj klasifikatora koji se obučavaju povećava sa brojem klasa. Uprkos potencijalnim izazovima pri radu sa velikim brojem klasa, ovaj metod se ističe kao dobra opcija kada je potrebna visoka tačnost klasifikacije uz neophodno postojanje zadovoljavajućih resursa.

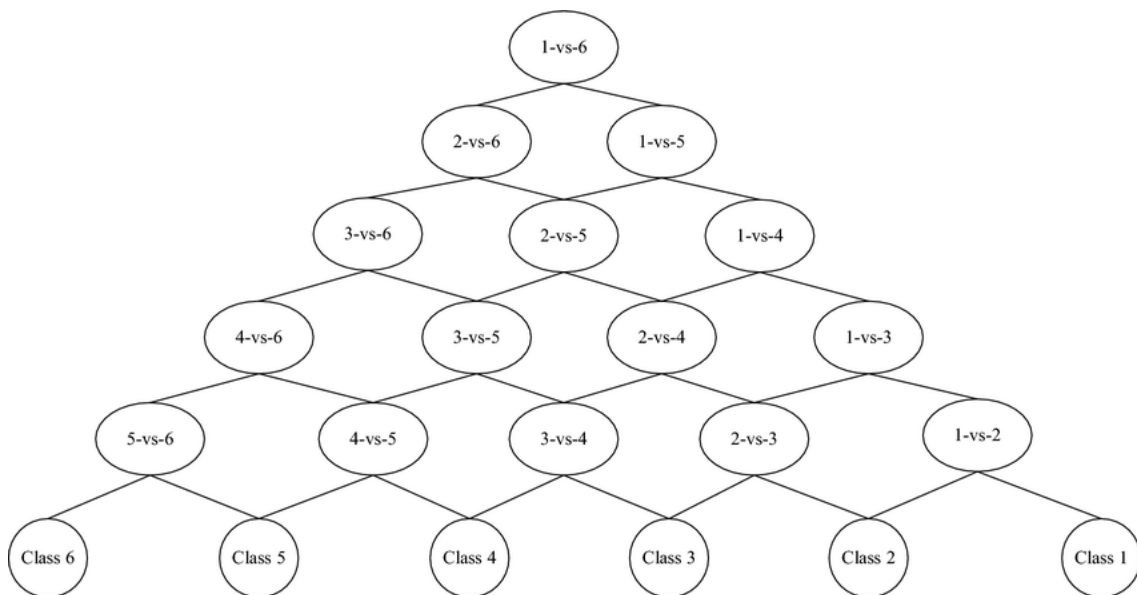


---

## DAGSVM

Treći algoritam koji ćemo razmotriti za rešavanje višeklasnog problema je DAGSVM (eng. Directed Acyclic Graph Support Vector Machines), predložen u [14]. Direktni aciklični graf je skup čvorova i usmerenih grana pri čemu nije moguće vratiti se unazad. Koren, koji je prvi član grafa, može biti bilo koja od  $k$  klasa, jer je eksperimentalno potvrđeno da se svaki izbor svodi na istu finalnu odluku. Isto važi i za izbor klasa za koje formiramo binarne klasifikatore u čvorovima. Zbog toga se obično koristi prirodan redosled klasa (pogledati sliku 3.1), što olakšava i pojednostavljuje proces treniranja.

Broj binarnih klasifikatora ovog algoritma je  $k(k - 1)/2$ , i isti je kao u metodi jedan-na-jedan. Zatim se koristi korenski binarni usmereni aciklični graf koji ima  $k(k - 1)/2$  unutrašnjih čvorova i  $k$  listova. Svaki čvor je binarni SVM za  $i$ -tu i  $j$ -tu klasu. Za dati uzorak  $x$ , počevši od korena, evaluira se binarna funkcija odlučivanja. Zatim se kreće ili levo ili desno u zavisnosti od izlazne vrednosti. U svakom čvoru se odbacuje jedna klasa. Proces se završava kada ostane samo jedna klasa u listu. Opisani postupak predstavlja DDAG (eng. Decision Directed Acyclic Graph) strukturu.



Slika 3.1: Primer DAGSVM algoritma [17]

---

Da bismo ocenili određeni DDAG za ulazni vektor  $x$ , krećemo se od korena i određujemo binarne funkcije na čvorovima. Zavisno od vrednosti binarne funkcije (0 ili 1), iz čvora se krećemo levo ili desno. Zatim se određuje binarna funkcija sledećeg čvora. Vrednost funkcije odlučivanja  $D(x)$  je vrednost povezana sa poslednjim listnim čvorom. Putanja kroz DDAG je poznata kao evaluaciona putanja. Ulazni vektor  $x$  će stići do nekog čvora grafa ako je taj čvor deo evaluacione putanje za  $x$ . Imenovaćemo odlučujući čvor koji razlikuje klase  $i$  i  $j$  kao  $ij$ -čvor. Pretpostavljajući da je broj listova jednak broju klasa, ovaj čvor je  $i$ -ti čvor u  $(k - j + i)$ -tom sloju, pod uslovom da je  $i < j$ .

Kako bismo olakšali razumevanje, povezaćemo DDAG sa strukturom podataka nalik listi. Lista se inicijalizuje sa svim klasama, a svaki odlučujući čvor odgovara prvom i poslednjem elementu liste dok ne ostane samo jedna klasa u listi. Dakle, za problem sa  $k$  klasa, biće procenjeno  $(k - 1)$  binarnih klasifikatora. Trenutno stanje liste je ukupno stanje sistema. Pošto je stanje liste dostižno na više mogućih puteva kroz sistem, graf odluka koji algoritam pretražuje je direktan aciklični graf (DAG), a ne samo stablo. Odlučujući DAGovi prirodno generalizuju klasu stabala odlučivanja (eng. decision tree), omogućavajući efikasnije predstavljanje ponavljajućih situacija koje mogu nastati u različitim granama stabla putem spajanja različitih putanja odlučivanja.

Dakle, DAGSVM predstavlja efikasan pristup rešavanju višeklasnih problema u mašinskom učenju. Iako je proces treniranja isti kao u metodi jedan-na-jedan, ovaj algoritam se izdvaja po svojoj sposobnosti da u procesu testiranja redukuje broj potrebnih binarnih klasifikatora na  $k - 1$ , u poređenju sa  $k(k - 1)/2$  potrebnih klasifikatora u metodi jedan-na-jedan. Ova redukcija ima značajan uticaj, posebno u slučaju velikog broja klasa. Takođe omogućava brzo i efikasno donošenje odluka u realnom vremenu, što je posebno važno u kontekstu bankarskih transakcija, gde brzina procesa igra ključnu ulogu.

---

## Metod Vestona i Votkinsa

Za razliku od prethodnih pristupa, Veston i Votkins su predstavili metod gde se višeklasni slučaj ne svodi na kombinaciju binarnih klasifikatora. Njihov pristup se sastoji u konstrukciji funkcije odlučivanja razmatranjem svih klasa odjednom [16]. Autori predlažu da se optimizacioni problem 3.1 generalizuje na sledeći način:

$$\begin{aligned}\phi(\mathbf{w}, \xi) &= \frac{1}{2} \sum_{m=1}^k \mathbf{w}_m \cdot \mathbf{w}_m + C \sum_{i=1}^l \sum_{m \neq y_i} \xi_i^m, \\ (\mathbf{w}_{y_i} \cdot x_i) + b_{y_i} &\geq (\mathbf{w}_m \cdot x_i) + b_m + 2 - \xi_i^m, \\ \xi_i^m &\geq 0, i = 1, \dots, l; m \in \{1, \dots, k\} \setminus \{y_i\}.\end{aligned}\quad (3.3)$$

Ovaj postupak daje funkciju odlučivanja:

$$f(x) = \arg \max_i [(\mathbf{w}_i \cdot x) + b_i], \quad i = 1, \dots, k. \quad (3.4)$$

Rešenje za ovaj optimizacioni problem možemo pronaći u dualnim promenljivima, traženjem tačke sedla Lagranžijana:

$$\begin{aligned}L(\mathbf{w}, b, \xi, \alpha, \beta) &= \frac{1}{2} \sum_{m=1}^k (\mathbf{w}_m \cdot \mathbf{w}_m) + C \sum_{i=1}^l \sum_{m=1}^k \xi_i^m \\ &- \sum_{i=1}^l \sum_{m=1}^k \alpha_i^m [(\mathbf{w}_{y_i} - \mathbf{w}_m) \cdot x_i] + b_{y_i} - b_m - 2 + \xi_i^m] - \sum_{i=1}^l \sum_{m=1}^k \beta_i^m \xi_i^m,\end{aligned}\quad (3.5)$$

sa pomoćnim promenljivama

$$\alpha_i^{y_i} = 0, \xi_i^{y_i} = 2, \beta_i^{y_i} = 0, i = 1, \dots, l$$

i uslovima

$$\alpha_i^m \geq 0, \beta_i^m \geq 0, \xi_i^m \geq 0, i = 1, \dots, l; m \in \{1, \dots, k\} \setminus \{y_i\}.$$

Ovaj problem mora biti maksimizovan u odnosu na  $\alpha$  i  $\beta$  i minimizovan u odnosu na  $\mathbf{w}$  i  $\xi$ . Uvođenjem oznaka

$$c_i^n = \begin{cases} 1, & \text{ako } y_i = n \\ 0, & \text{ako } y_i \neq n \end{cases}$$

i

$$A_i = \sum_{m=1}^k \alpha_i^m$$

i diferenciranjem po  $\mathbf{w}_n, b_n$  i  $\xi_j^n$ , za  $n \in \{1, \dots, k\}$  dobijamo:

$$\frac{\partial L}{\partial \mathbf{w}_n} = \mathbf{w}_n + \sum_{i=1}^l \alpha_i^n x_i - \sum_{i=1}^l A_i c_i^n x_i,$$

$$\frac{\partial L}{\partial b_n} = - \sum_{i=1}^l A_i c_i^n + \sum_{i=1}^l \alpha_i^n,$$

$$\frac{\partial L}{\partial \xi_j^n} = -\alpha_j^n + C - \beta_j^n.$$

U tački sedla, rešenje treba da zadovolji sledeće uslove:

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \mathbf{w}_n} = 0 \Rightarrow \mathbf{w}_n = \sum_{i=1}^l (c_i^n A_i - \alpha_i^n) x_i, \quad (3.6)$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial b_n} = 0 \Rightarrow \sum_{i=1}^l \alpha_i^n = \sum_{i=1}^l c_i^n A_i, \quad (3.7)$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \alpha, \beta)}{\partial \xi_n} = 0 \Rightarrow \beta_j^n + \alpha_j^n = C \text{ i } 0 \leq \alpha_j^n \leq C. \quad (3.8)$$

Zamenom 3.6 nazad u 3.5, dobijamo:

$$\begin{aligned} L^*(b, \xi, \alpha, \beta) &= \frac{1}{2} \sum_{m=1}^k \sum_{i=1}^l \sum_{j=1}^l (c_i^m A_i - \alpha_i^m)(c_j^m A_j - \alpha_j^m)(x_i \cdot x_j) \\ &- \sum_{m=1}^k \sum_{i=1}^l \alpha_i^m \left[ \sum_{j=1}^l (c_j^{y_i} A_j - \alpha_j^{y_i})(x_i \cdot x_j) - \sum_{j=1}^l (c_j^m A_j - \alpha_j^m)(x_i \cdot x_j) + b_{y_i} - b_m - 2 \right] \\ &- \sum_{m=1}^k \sum_{i=1}^l \alpha_i^m \xi_i^m + C \sum_{m=1}^k \sum_{i=1}^l \xi_i^m - \sum_{i=1}^l \sum_{m=1}^k \beta_i^m \xi_i^m. \end{aligned}$$

Nakon što uključimo 3.8 dobijemo da se poslednja tri elementa u izrazu potiru.

Analiziramo dva člana koja idu uz  $b$ :

$$\begin{aligned} B_1 &= \sum_{m=1}^k \sum_{i=1}^l \alpha_i^m b_{y_i} = \sum_{m=1}^k b_m \left( \sum_{i=1}^l c_i^m A_i \right), \\ B_2 &= - \sum_{m=1}^k \sum_{i=1}^l \alpha_i^m b_m = - \sum_{m=1}^k b_m \left( \sum_{i=1}^l \alpha_i^m \right). \end{aligned}$$

Međutim, iz 3.7 imamo:

$$\sum_{i=1}^l \alpha_i^m = \sum_{i=1}^l c_i^m A_i,$$

dakle  $B_1 = B_2$  tj. ta dva člana se poništavaju, dajući:

$$L^*(\alpha) = 2 \sum_{i,m} \alpha_i^m + \sum_{i,j,m} \left( \frac{1}{2} c_i^m c_j^m A_i A_j - \frac{1}{2} c_i^m A_i \alpha_j^m - \frac{1}{2} c_j^m A_j \alpha_i^m + \frac{1}{2} \alpha_i^m \alpha_j^m - c_j^{y_i} A_j \alpha_i^m + \alpha_i^m \alpha_j^{y_i} + c_j^m A_j \alpha_i^m - \alpha_i^m \alpha_j^m \right) (x_i \cdot x_j). \quad (3.9)$$

Pošto važi  $\sum_m c_i^m A_i \alpha_j^m = \sum_m c_j^m A_j \alpha_i^m$ , 3.9 svodimo na:

$$L^*(\alpha) = 2 \sum_{i,m} \alpha_i^m + \sum_{i,j,m} \left( \frac{1}{2} c_i^m c_j^m A_i A_j - c_j^{y_i} A_i A_j + \alpha_i^m \alpha_j^{y_i} - \frac{1}{2} \alpha_i^m \alpha_j^m \right) (x_i \cdot x_j).$$

Takođe  $\sum_m c_i^m c_j^m = c_i^{y_i} = c_j^{y_j}$ , sledi:

$$L^*(\alpha) = 2 \sum_{i,m} \alpha_i^m + \sum_{i,j,m} \left[ -\frac{1}{2} c_j^{y_i} A_i A_j + \alpha_i^m \alpha_j^{y_i} - \frac{1}{2} \alpha_i^m \alpha_j^m \right] (x_i \cdot x_j), \quad (3.10)$$

što je kvadratna funkcija u odnosu na  $\alpha$  sa linearnim ograničenjima

$$\sum_{i=1}^l \alpha_i^n = \sum_{i=1}^l c_i^n A_i, \quad n = 1, \dots, k,$$

uz uslove:

$$0 \leq \alpha_i^m \leq C, \quad \alpha_i^{y_i} = 0, \\ i = 1, \dots, l, \quad m \in \{1, \dots, k\} \setminus y_i.$$

Ovim se dobija funkcija odlučivanja:

$$f(x; \alpha) = \arg \max_n \left[ \sum_{i=1}^l (c_i^n A_i - \alpha_i^n) (x_i \cdot x) + b_n \right] \quad (3.11)$$

ili ekvivalentno

$$f(x, \alpha) = \arg \max_n \left[ \sum_{i:y_i=n} A_i (x_i \cdot x) - \sum_{i:y_i \neq n} \alpha_i^n (x_i \cdot x) + b_n \right].$$

Kao i obično, skalarni proizvodi  $(x_i \cdot x_j)$  u jednakostima 3.10 i 3.11 se mogu zameniti konvolucijom skalarnog proizvoda  $K(x_i, x_j)$ .

---

Razmotrimo nadalje rešenje optimizacionog problema 3.3 za različite vrednosti broja klasa  $k$ :

1. Najjednostavniji slučaj :  $k = 1$

Tada nema ograničenja, i odluka se implicitno dodeljuje klasi 1.

2. Binarna klasifikacija :  $k = 2$

Ako označimo da  $y_i \in \{0, 1\}$ , tada za svaku tačku trening skupa postoji samo jedno ograničenje:

$$(\mathbf{w}_{y_i} \cdot x_i) + b_{y_i} \geq (\mathbf{w}_{1-y_i} \cdot x_i) + b_{1-y_i} + 1 - \xi_i^{1-y_i}.$$

Razdvajanje dve klase pomoću maksimuma dve linearne funkcije je ekvivalentno razdvajanju pomoću jedinstvene hiperravni. Dobijamo:

$$((\mathbf{w}_{y_i} - \mathbf{w}_{1-y_i}) \cdot x_i) + b_{y_i} - b_{1-y_i} \geq 1 - \xi_i^{1-y_i}.$$

Može se pokazati da je ova hiperravan identična onoj dobijenoj iz binarnog SVM-a sa mekim pojasom 2.8, i da će imati iste koeficijente potpornih vektora. Li i saradnici [9] i Smola i Šulkpof [15] su pokazali da su rezultati opisanog pristupa i jedan-protiv-svih pristupa slični. U ovom pristupu se razmatraju svi potporni vektori odjednom što zahteva više memorije i može dovesti do značajno dužeg vremena izvršavanja algoritma.

---

## Metod Kramera i Singera

Kramer i Singer su predložili sličan pristup prethodno opisanom metodu Vestona i Votkinsa [4]. Umesto svodenja na binarnu klasifikaciju, višeklasna klasifikacija se rešava jednim optimizacionim problemom. U suštini, rešavamo sledeći problem:

$$\begin{aligned} \min_{\mathbf{w}_m, \xi_i} \quad & \frac{1}{2} \sum_{m=1}^k \mathbf{w}_m^T \mathbf{w}_m + C \sum_{i=1}^l \xi_i, \\ \text{uz uslov} \quad & \mathbf{w}_{y_i}^T \phi(\mathbf{x}_i) - \mathbf{w}_m^T \phi(\mathbf{x}_i) \geq e_i^m - \xi_i, \quad i = 1, \dots, l, \end{aligned} \quad (3.12)$$

gde je:

$$\begin{aligned} e_i^m &\equiv 1 - \delta_{y_i, m}, \\ \delta_{y_i, m} &\equiv \begin{cases} 1, & \text{ako } y_i = m, \\ 0, & \text{ako } y_i \neq m. \end{cases} \end{aligned}$$

Tada je funkcija odlučivanja:

$$\arg \max_{m=1, \dots, k} \mathbf{w}_m^T \phi(\mathbf{x}).$$

Glavna razlika u odnosu na 3.3 je ta što 3.12 koristi samo  $l$  promenljivih  $\xi_i, i = 1, \dots, l$ . To jest, umesto da se koristi  $\xi_i^m$  kao razdaljina između svake dve hiperravni, ovde se uzima maksimum od  $k$  takvih brojeva:

$$\xi_i = \left( \max_m (\mathbf{w}_m^T \phi(\mathbf{x}_i) + e_i^m) - \mathbf{w}_{y_i}^T \phi(\mathbf{x}_i) \right)_+,$$

gde je  $(\cdot)_+ \equiv \max(\cdot, 0)$ . Pored toga, 3.12 ne sadrži koeficijente  $b_i, i = 1, \dots, l$ . Primetimo da ne moramo eksplicitno navoditi ograničenja  $\xi_i \geq 0$  jer kada  $y_i = m, e_i^m = 0$ , tada uslov 3.12 postaje:

$$0 \geq 0 - \xi_i,$$

što je zapravo  $\xi_i \geq 0$ .

Dualni problem za 3.12 je:

$$\begin{aligned} \min_{\alpha} f(\alpha) &= \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l K_{i,j} \bar{\alpha}_i^T \bar{\alpha}_j + \sum_{i=1}^l \bar{\alpha}_i^T \bar{e}_i, \\ \sum_{m=1}^k \alpha_i^m &= 0, \quad i = 1, \dots, l, \\ \alpha_i^m &\leq 0, \quad \text{ako } y_i \neq m, \\ \alpha_i^m &\leq C, \quad \text{ako } y_i = m, \quad i = 1, \dots, l, \quad m = 1, \dots, k, \end{aligned} \quad (3.13)$$

---

gde je:

$$\begin{aligned} K_{i,j} &\equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \\ \bar{\alpha}_i &\equiv [\alpha_i^1, \dots, \alpha_i^k]^T, \\ \bar{e}_i &\equiv [e_i^1, \dots, e_i^k]^T. \end{aligned}$$

Imamo,

$$\mathbf{w}_m = \sum_{i=1}^l \alpha_i^m \phi(\mathbf{x}_i).$$

Ako  $\alpha$  i  $e$  izrazimo na sledeci nacin  $\alpha \equiv [\alpha_1^1, \dots, \alpha_1^k, \dots, \alpha_l^1, \dots, \alpha_l^k]^T$  i  $e \equiv [e_1^1, \dots, e_1^k, \dots, e_l^1, \dots, e_l^k]^T$ , tada ciljnu funkciju možemo napisati kao:

$$\frac{1}{2} \alpha^T (K \otimes I) \alpha + e^T \alpha,$$

gde je  $I$  jedinična matrica dimenzija  $k \times k$  i  $\otimes$  je Kronekerov proizvod. Pošto je  $K$  pozitivno semidefinitna matrica,  $K \otimes I$ , Hesijan ciljne funkcije je takođe pozitivno semidefinitan. Ovo je još jedan način da se objasni da je 3.13 konveksni optimizacioni problem.

Funkcija odlučivanja je:

$$\arg \max_{m=1, \dots, k} \sum_{i=1}^l \alpha_i^m K(\mathbf{x}_i, \mathbf{x}).$$

Autori su predložili da svaki put kada optimizujemo model, optimizujemo svih  $k$  varijabli  $\alpha_i^1, \dots, \alpha_i^k$  povezanih sa jednim uzorkom  $x_i$ . Tada je potproblem:

$$\begin{aligned} \min_{\alpha_i} \quad & \frac{1}{2} A \bar{\alpha}_i^T \bar{\alpha}_i + B^T \bar{\alpha}_i, \\ & \sum_{m=1}^k \alpha_i^m = 0, \\ & \alpha_i^m \leq C_{y_i}^m, \quad m = 1, \dots, k, \end{aligned} \tag{3.14}$$

gde su:

$$A = K_{i,i}, \quad B = \bar{e}_i + \sum_{j \neq i} K_{j,i} \bar{\alpha}_j.$$



Dodatno,  $C_{y_i}^m$ ,  $m = 1, \dots, k$  je vektor dimenzije  $k \times 1$ , sa svim elementima nula osim  $i$ -te komponente ( $y_i$ ) koja je  $C$ .

Gradijent dualne ciljne funkcije je:

$$\begin{bmatrix} \sum_{j=1}^l K_{1,j} \alpha_j^1 + e_1^1 \\ \vdots \\ \sum_{j=1}^l K_{1,j} \alpha_j^k + e_1^k \\ \sum_{j=1}^l K_{2,j} \alpha_j^1 + e_2^1 \\ \vdots \\ \sum_{j=1}^l K_{2,j} \alpha_j^k + e_2^k \\ \vdots \end{bmatrix}.$$

Zatim se  $B$ , vektor dimenzije  $k \times 1$ , može izračunati na sledeći način koristeći informacije iz gradijenta:

$$B_m = \frac{\partial f(\alpha)}{\partial \alpha_i^m} - K_{i,i} \alpha_i^m = \frac{\partial f(\alpha)}{\partial \alpha_i^m} - A \alpha_i^m, \quad m = 1, \dots, k.$$

Stoga, tokom iteracija je ključno uvek ažurirati gradijent. Ovo se radi nakon što su dobijene nove  $\alpha_i^1, \dots, \alpha_i^k$  pomoću 3.14 i potrebne su operacije složenosti  $O(kl)$ .

Dalje ćemo se osvrnuti na izbor radnog skupa  $\{\alpha_i^1, \dots, \alpha_i^k\}$  i uslov zaustavljanja dekompozicionog metoda. Radni skup je podskup varijabli odabran za optimizaciju u datom trenutku. KKT uslov zahteva da postoje  $b_1, \dots, b_l$  i  $\lambda_1^1 \geq 0, \dots, \lambda_l^k \geq 0$  tako da za sve  $i = 1, \dots, l$ ,  $m = 1, \dots, k$  važi:

$$\sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m - b_i = -\lambda_i^m,$$

$$\lambda_i^m (C_{y_i}^m - \alpha_i^m) = 0.$$

Ovi uslovi su ekvivalentni sledećem: za sve  $i = 1, \dots, l$ ,  $m = 1, \dots, k$ ,

$$\begin{cases} \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m - b_i = 0 & \text{ako } \alpha_i^m < C_{y_i}^m, \\ \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m - b_i \leq 0 & \text{ako } \alpha_i^m = C_{y_i}^m. \end{cases}$$

Možemo ovo preformulisati kao:

$$\max_{\alpha_i^m \leq C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) \leq b_i \leq \min_{\alpha_i^m < C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right). \quad (3.15)$$

---

Tokom iteracija biramo sledeći radni skup  $\{\alpha_i^1, \dots, \alpha_i^k\}$  sa indeksom  $i$  iz:

$$\arg \max_i \left( \max_{\alpha_i^m \leq C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) - \min_{\alpha_i^m < C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) \right).$$

Drugim rečima, među  $l$  grupa promenljivih sa  $k$  komponenti, biramo onu sa najvećim kršenjem KKT uslova. Kod binarnih SVM-ova, biranje indeksa koji najviše krše KKT uslov je standardna strategija, dok ovde biramo celu grupu odjednom. Tada promenljive  $\{\alpha_i^1, \dots, \alpha_i^k\}$  koje su izabrane, ne zadovoljavaju KKT uslov sub-problema 3.14 tako da rešavanje 3.14 garantuje striktno smanjenje ciljne funkcije dualnog problema.

Na osnovu 3.15, uslov zaustavljanja može biti:

$$\max_i \left( \max_{\alpha_i^m \leq C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) - \min_{\alpha_i^m < C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) \right) < \epsilon, \quad (3.16)$$

gde je  $\epsilon$  tolerancija za zaustavljanje. Konvergencija gornjeg dekompozicionog metoda 3.16 je dokazana u radu [10]. Takođe [10] pokazuje da granica od

$$\max_i \left( \max_{\alpha_i^m \leq C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) - \min_{\alpha_i^m < C_{y_i}^m} \left( \sum_{j=1}^l K_{i,j} \alpha_j^m + e_i^m \right) \right)$$

teži nuli kako broj iteracija teži beskonačnosti. Dakle, nakon konačnog broja iteracija, dekompozicioni metod se zaustavlja kada je 3.16 zadovoljen, što se može videti u radu [8].

---

## Kodiranje izlaznih kodova za korekciju grešaka

Naredni metod (eng. error correcting output code) ECOC su predložili Ditrih i Bakiri [6], a pristup su nadogradili Alvejn, Šapire i Singer [1]. Metod svodi problem razlikovanja  $C$  klasa na  $L$  binarnih klasifikacija. Svakoj klasi dodeljujemo binarnu kodnu reč dužine  $L$ . Kod za korekciju grešaka  $(L, C, d)$  je dug  $L$  bita, ima  $C$  jedinstvenih kodnih reči sa Hamingovim rastojanjem  $d$ . Hamingovo rastojanje između dve kodne reči je broj bitova u kojima se razlikuju. U problemu klasifikacije  $C$  je broj klasa, a  $L$  je broj određen metodom korišćenom za generisanje kodova za korekciju grešaka. Ditrih i Bakiri su predložili korišćenje kodova sa maksimalnim Hamingovim rastojanjem. Dekompozicija problema višeklasne klasifikacije sa  $C$  klasa, koje imaju oznake  $K_1, \dots, K_C$ , generiše skup od  $L$  binarnih klasifikatora koji su predstavljeni sa  $f_1, \dots, f_L$ . Binarni klasifikator deli ulazne šablone u dve komplementarne super klase  $K_i^1$  i  $K_i^{-1}$  grupišući jednu ili više klasa. Neka  $M = [b_{ij}]$  bude matrica dekompozicije dimenzija  $L \times C$ , koja povezuje klase  $K_1, \dots, K_C$  sa super klasama  $K_i^1$  i  $K_i^{-1}$ , gde se element matrice  $M$  može definisati kao:

$$b_{ij} = \begin{cases} 1, & \text{ako } K_j \subset K_i^1 \\ -1, & \text{ako } K_j \subset K_i^{-1} \end{cases}$$

Stoga, za  $C$  klasa, dobija se matrica kodiranja  $D \in \{\pm 1\}^{L \times C}$ . Kada novi podaci treba da budu klasifikovani, obučeni binarni klasifikatori proizvode odluku  $e_i$  da test podaci potiču iz  $i$ -te grupe, proizvedeći vektor procena,  $e = (e_1, e_2, \dots, e_L)^T$  od svih binarnih klasifikatora. Da bi se klasifikovali novi podaci, računa se  $L_1$ -udaljenost za svaku od  $C$  klasa i dodeljuje se klasa sa minimalnom  $L_1$ -udaljenošću.  $L_1$ -udaljenost je definisana kao:

$$L_1 = \sum_{i=1}^L |e_j - o_{ij}|$$

gde je  $o_{ij}$   $j$ -ti bit kodne reči klase  $i$ . [12]

Radi boljeg razumevanja algoritma, razmotrimo sledeću tabelu 3.1 koja prikazuje šestobitne kodne reči za problem prepoznavanja cifara. Kao i u većini primena kodiranja izlaznih kodova, bitovi u kolonama su odabrani da budu smisleni. U ovom slučaju njihova vrednost predstavlja sledeće osobine cifara:

- vl - cifra sadrži vertikalnu liniju,
- hl - cifra sadrži horizontalnu liniju,
- dl - cifra sadrži dijagonalnu liniju,
- cc - cifra sadrži zatvorenu krivu,
- ol - cifra sadrži krivu otvorenu na levo,
- or - cifra sadrži krivu otvorenu na desno.

Tabela 3.1: Primer ECOC

Klasa	vl	hl	dl	cc	ol	or
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	1	0
3	0	0	0	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	0	0	1	0	0
9	0	0	1	1	0	0

Primetimo da je svaki red jedinstven, tako da svaka klasa ima jedinstvenu kodnu reč. Tokom učenja, jedna binarna funkcija će biti naučena za svaku kolonu i pritom je svaka kolona jedinstvena. Svaka binarna funkcija koja treba da se nauči je disjunkcija originalnih klasa, tj.  $f_{vl}(x) = 1$  ako  $f(x) \in \{1, 4, 5\}$  i  $f_{vl}(x) = 0$  ako  $f(x) \in \{0, 2, 3, 6, 7, 8, 9\}$ . Za klasifikaciju novog ručno pisanog broja  $x$ , šest binarnih funkcija  $f_{vl}, f_{hl}, f_{dl}, f_{cc}, f_{ol}, f_{or}$  se evaluiraju da bi se dobio šestobitni niz, npr. 110001. Zatim se računa udaljenost ovog niza od svake od deset kodnih reči. Najbliža kodna reč, po Hamingovom rastojanju koja broji broj bitova koji se razlikuju, je 110000, što odgovara klasi 4. Stoga, ovo predviđa da je  $f(x) = 4$ . (videti [6])

---

Kodovi za korekciju grešaka sugerišu da posmatramo mašinsko učenje kao problem komunikacije u kome se identitet tačne izlazne klase za novi primer „prenosi” preko kanala. Kanal se sastoji od ulaznih karakteristika, trening primera i algoritma učenja. Zbog grešaka prouzrokovanih ograničenim uzorkom treninga, lošim izborom ulaznih karakteristika i greškama u procesu učenja, klasa informacije je oštećena. Korišćenjem kodiranja klase u kodu za korekciju grešaka i „prenoseći” svaki bit zasebno (tj. kroz zasebno pokretanje algoritma učenja), sistem može biti sposoban da se oporavi od grešaka.

Dodatni eksperimenti su pokazali da su rezultati neosetljivi na specifičnu dodelu kodnih reči klasama. Intervali poverenja su pokazali da nema statistički značajnih varijacija u performansama različitih nasumičnih dodela kodnih reči.

Metoda kodiranja za korekciju grešaka kod višeklasnih problema pokazala je značajna poboljšanja u odnosu na tradicionalne pristupe. Prednosti uključuju bolju generalizaciju, otpornost na greške i robusnost u prisustvu buke i grešaka u učenju.

# Glava 4

## Primena

Kod korišćen u radu se nalazi na github linku<sup>1</sup>. U praktičnom delu korišćeni su podaci koji se odnose na transakcije australijskih i novozelandskih banaka. Podaci su originalni, vlasništvo su kompanije koja se bavi finansijskom analitikom, sačuvani su u csv fromatu, i nisu ranije upotrebljavani u naučne svrhe. Skup podataka sadrži 522846 redova i 6 kolona. Cilj je da prepoznamo tip transakcije, tj. da li je u pitanju plaćanje, podizanje novca, transfer itd. U slučaju da je u pitanju plaćanje, cilj je da detaljnije opišemo transakciju prepoznavanjem delatnosti (npr. da li je kupovina obavljenu u supermarketu, u robnoj kući, u kladionici itd.).

Kolone i njihovi opisi su sledeći:

- *request id*: jedinstveni identifikator svake transakcije,
- *transaction description*: opis transakcije koji može sadržavati ime trgovca, lokaciju ili druge relevantne detalje,
- *transaction amount*: iznos transakcije, izražen kao decimalni broj. Negativne vrednosti predstavljaju troškove, dok pozitivne vrednosti predstavljaju prilive sredstava,
- *transaction account type*: tip računa s kojeg je obavljena transakcija, kao što su transakcioni račun ili štedni račun.

Ciljne promenljive su:

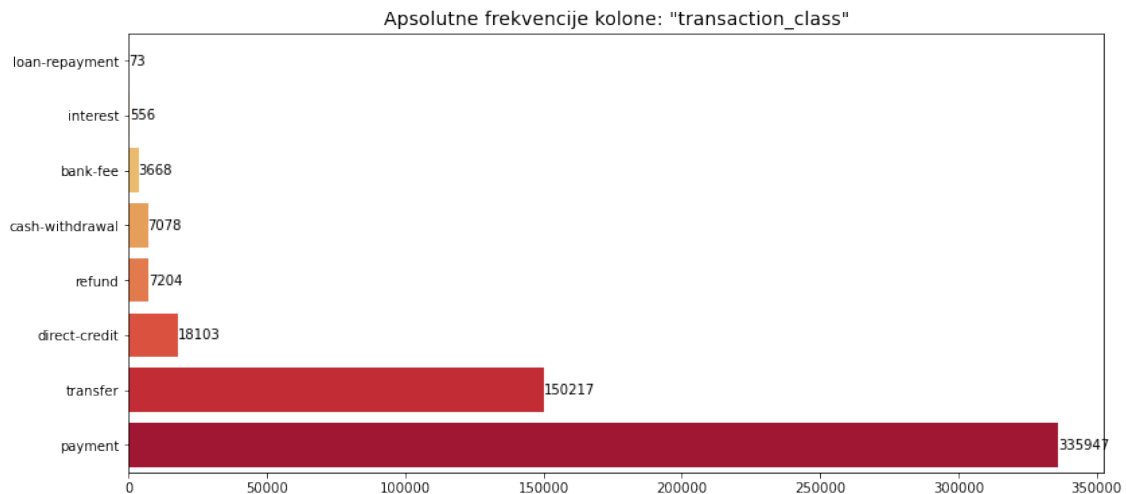
- *transaction class*: klasifikacija transakcije, na primer, plaćanje ili transfer,

---

<sup>1</sup>[https://github.com/VladimirLukovic/Master\\_rad-Klasifikacija\\_bankarskih\\_transakcija/tree/main](https://github.com/VladimirLukovic/Master_rad-Klasifikacija_bankarskih_transakcija/tree/main)

- *anzsic*: predstavlja šifru delatnosti.

Važno je napomenuti da kolona *anzsic* postoji samo za transakcije koje su klasifikovane kao *payment*, jer se u tim slučajevima vrši plaćanje firmama koje su označene odgovarajućim *ANZSIC* kodovima.



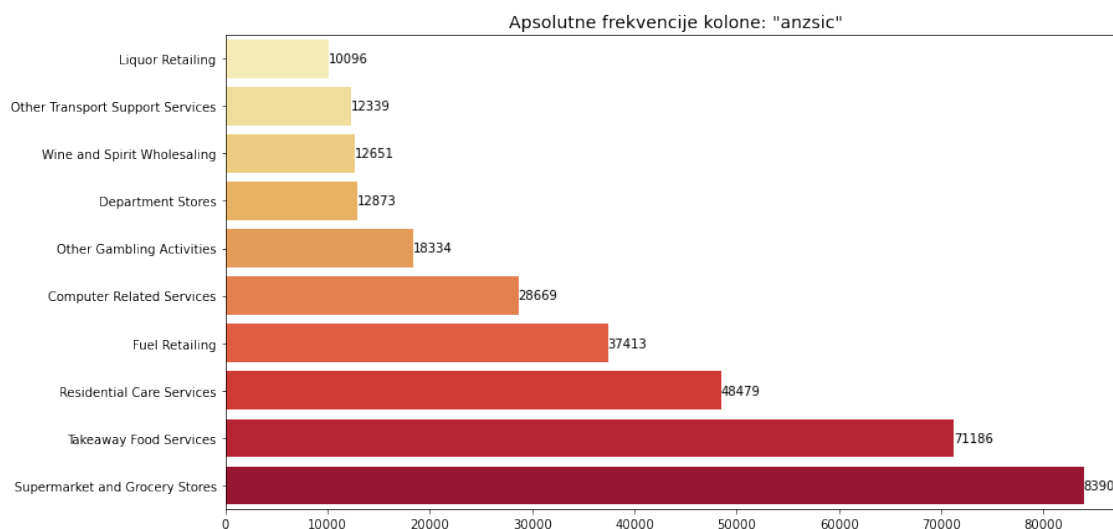
Slika 4.1: Apsolutne frekvencije ciljne promenljive

Podaci o transakcijama sadrže različite vrednosti u koloni **transaction\_class**:

- loan repayment: otplata zajma,
- interest: isplate kamata,
- bank fee: naknade koje banka naplaćuje za različite usluge,
- cash withdrawal: isplata gotovine,
- refund: povraćaj sredstava,
- direct credit: direktne uplate,
- transfer: transfer sredstava između računa,
- payment: plaćanje pružaocima usluga ili trgovcima.

U radu ćemo se prvo baviti klasifikacijom transakcija prema koloni **transaction\_class**. Nakon što utvrdimo da određena transakcija spada u *payment*, predviđaćemo *anzsic* kod te transakcije. Tako ćemo osigurati tačno dodeljivanje delatnosti samo za relevantne transakcije. Primer primene ovog slučaja je grupisanje

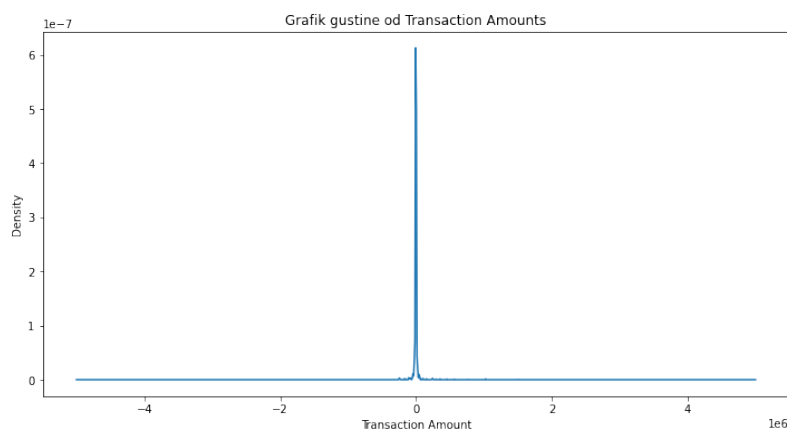
transakcija po kategorijama u bankarskim aplikacijama za fizička lica, kako bi mogli da prate svoju potrošnju po različitim kategorijama poput putovanja, restorana, odeće, hrane...



Slika 4.2: Apsolutna frekvencija delatnosti

Većina informacija o šiframa delatnosti odnosi se na prehrambeni sektor, kao i na česte ljudske aktivnosti poput transporta i stanovanja, što odražava ključne aspekte svakodnevnog života i potrošačkih navika. Posebno se ističu delatnosti kao što su *Supermarket and Grocery Stores* i *Takeaway Food Service*, koje imaju najvišu frekvenciju, dok je *Liquor Retailing* među najmanje zastupljenim.

U nastavku ćemo detaljnije ispitati svaku od kolona kako bismo stekli bolji uvid u podatke.



Slika 4.3: Raspodela promenljive koja sadrži ne-nula iznos transakcije



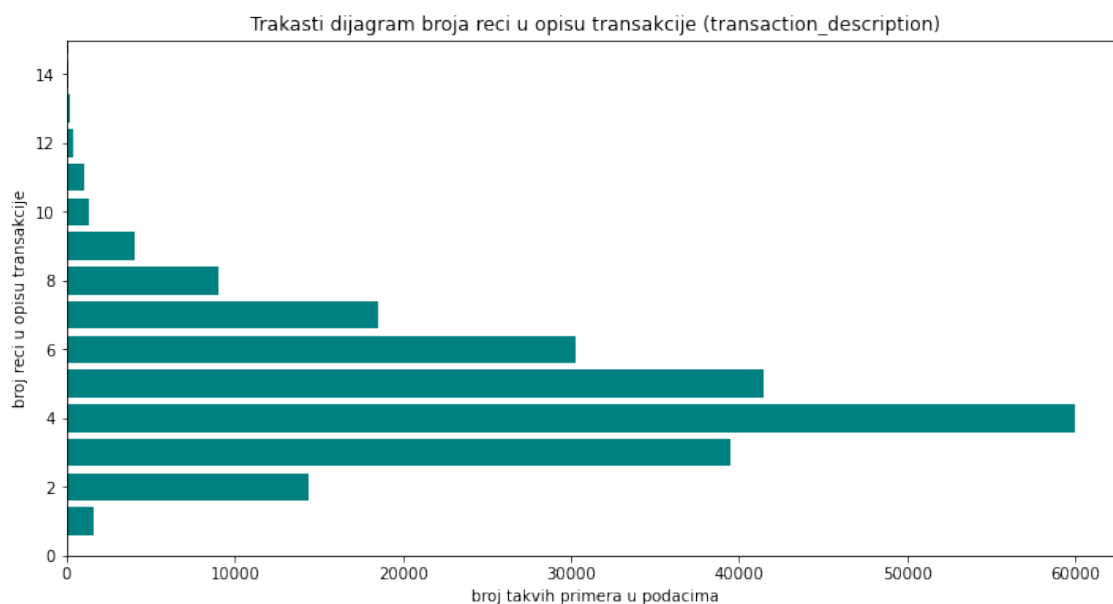


---

Iz prikaza se ističu termini poput *value*, *date*, *card*, i *aus*, što ukazuje na to da se opis transakcija često odnosi na detalje plaćanja i informacija o karticama. Prisutnost reči poput *commbank* i *app* ukazuje na učestalu upotrebu digitalnih platformi za upravljanje finansijama.

Na osnovu domenskog znanja, formirana je lista čestih, a nebitnih reči, koje ne pružaju dodatnu informaciju. Ove reči ćemo eliminisati iz opisa transakcija koristeći regularne izraze. Dodatno čistimo izolovane karaktere (npr. *!*, *?*, *\**, itd.), beline, kao i svako pojavljivanje liste specifičnih karaktera. Ovim postupkom ćemo dobiti pročišćene opise bez gubitka informacije. Nakon uklanjanja nepotrebnih reči, neke transakcije postaju iste, pa je zbog toga neophodno ukloniti duplikate iz uzorka.

Kako bismo promenljivu tipa string preveli u numerički oblik neophodno je da sve reči predstavimo u istoj dužini. Ideja je pronaći optimalan balans između količine podataka koje vektor sadrži i minimizacije gubitka informacija usled skraćivanja dužine opisa. Da bismo odredili maksimalnu dužinu reči, pogledajmo kako izgleda trenutna raspodela broja reči u opisu transakcije u odnosu na broj takvih primer u podacima.



Slika 4.6: Raspodela dužine reči u opisu transakcije

Sa slike 4.6 zaključujemo da najčešći opis transakcije sadrži između 3 i 6 reči, dok više od 90% transakcija ima opis dužine 8 ili manje. Na osnovu toga možemo definisati hiperparametar *max\_len* koji predstavlja maksimalnu dužinu vektora reči, poslatog modelu. Grafik ukazuje da je dovoljno da uzmemo 6, 7 ili 8 reči.

---

U cilju primene različitih SVM pristupa višeklasnoj klasifikaciji, sprovedena je temeljna priprema podataka. U nastavku opisujemo korake koji su urađeni.

- 1) Podeljen je skup podataka na trening i test skup u odnosu 80% prema 20% na slučajan način. Stratifikacija je primenjena u odnosu na ciljnu promenljivu, kako bi se održao proporcionalan udeo klasa u oba skupa.
- 2) Kolona *transaction\_amount* je standardizovana korišćenjem **StandardScaler** algoritma, kako bi se postigli normalizovani podaci sa srednjom vrednošću 0 i standardnom devijacijom 1. Ovaj korak je važan kako bi se osiguralo da raspodela bude usklađena, što poboljšava performanse modela. **StandardScaler** je ocenjen na trening skupu, a zatim primenjen na oba skupa podataka, tj. nijedna informacija iz test podataka nije korišćena tokom procesa treniranja.
- 3) Kategorička kolona *transaction\_account\_type*, koja predstavlja tipove računa, kodirana je u binarni oblik korišćenjem **OneHotEncoding**. Kreirane su indikatorske promenljive koje omogućavaju numeričku reprezentaciju podataka u trening i test skupu.
- 4) Treniran je **Word2Vec** model na osnovu opisa transakcija u koloni *transaction\_description* radi dobijanja vektorske reprezentacije reči. Opisi su najpre podeljeni na pojedinačne reči, a zatim je obučena Word2Vec mreža koja stvara vektore reči na osnovu konteksta. Vektori su zatim dopunjeni do iste dužine korišćenjem funkcije za dodavanje sekvenci, kako bi dimenzija podataka bila ujednačena za analizu i predikciju.
- 5) Zatim, izvršeno je ravnanje ulaznih vektora za model, u kojem su svi delovi ulaznog skupa (opis, iznos, tip računa) spojeni u jedan zajednički vektor.
- 6) Na kraju, ciljne promenljive *transaction\_class* za model koji klasifikuje tip transakcije i *anzsic* za model koji prepoznaje delatnost, su enkodirane korišćenjem **LabelEncoder**, koji transformiše kategoričke klase u numerički format prikladan za treniranje modela. Ovaj postupak primenjen je za trening i test skup, a sve inicijalne klase su sačuvane za kasniju interpretaciju rezultata.

Ovi procesi pripreme podataka su ključni koraci koji omogućavaju pravilnu evaluaciju različitih SVM pristupa u višeklasnoj klasifikaciji i pružaju korisne informacije za dalje analize i interpretaciju rezultata.

---

Sada želimo da se posvetimo načinu evaluacije rezultata pri višeklasnoj SVM klasifikaciji. Kao mere preciznosti koristićemo tačnost i balansiranu tačnost.

1. Tačnost je metrika koja meri koliko je model bio tačan u klasifikacijama s obzirom na ukupan broj predikcija. Formula za tačnost koja je korišćena za binarni slučaj se modifikuje za slučaj više klasa. Izračunava se prema formuli:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Broj tačno predikovanih uzoraka}}{\text{Ukupan broj uzoraka}} \\ &= \frac{\sum_{i=1}^k TP_i}{\sum_{i=1}^k (TP_i + FP_i + FN_i)}. \end{aligned}$$

Gde su:

- $TP_k$  (True Positives) — broj tačno predikovanih uzoraka za klasu  $k$ .
- $FP_k$  (False Positives) — broj pogrešno predikovanih uzoraka za klasu  $k$ .
- $FN_k$  (False Negatives) — broj pogrešno predikovanih uzoraka kao klasa  $k$ .

Tačnost može biti korisna kada su klase u uravnoteženom odnosu (tj. kada postoji sličan broj uzoraka u svakog razredu). Ipak, u slučaju nebalansiranih klasa, može dati lažni utisak uspeha modela.

2. Balansirana tačnost je metrika koja uzima u obzir nesrazmernost klasa tako što meri uspešnost modela na svakoj klasi posebno, i zatim izračunava prosek tih tačnosti [11]. Izračunava se prema formuli:

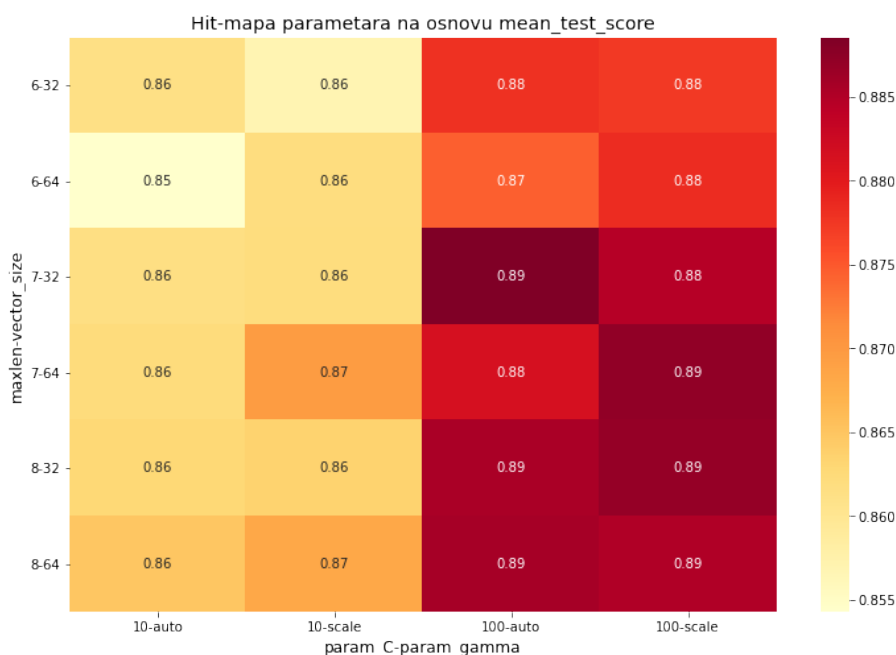
$$\text{Balanced Accuracy} = \frac{1}{k} \sum_{i=1}^k \text{Recall}_i$$

Gde je:

- $k$  — ukupan broj klasa.
- $\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}$  — osetljivost za klasu  $i$ .

Ovaj prosek omogućava da se svakoj klasi da jednaka težina prilikom izračunavanja tačnosti, čime se dobija realističnija slika o performansama modela u scenarijima gde su klase nebalansirane. U praksi, posebno u klasifikacionim problemima sa više klasa ili neizbalansiranim skupovima podataka, balansirana tačnost je često preporučena metrika za evaluaciju.

Radi sistematske pretrage najboljih kombinacija hiperparametara, korišćena je tehnika **GridSearchCV** na trening skupu podataka. Ova metoda omogućava temeljno pretraživanje prostora hiperparametara i kroz unakrsnu validaciju pomaže u odabiru parametara koji daju najbolje rezultate. Na taj način se smanjuje rizik od prekomernog ili nedovoljnog treniranja modela. Takođe se obezbeđuje bolja generalizacija modela na neviđenim podacima.



Slika 4.7: Toplotna mapa hiperparametara SVM modela

Na slici 4.7 je pomoću toplotne mape vizuelno predstavljena tačnost u odnosu na izbor hiperparametara za transakcioni model. GridSearchCV tehnika pretražuje kombinacije vrednosti parametara SVM modela kao što su  $C$  i  $gamma$ , kroz različite veličine Word2Vec vektora ( $vector\_size$ ) i maksimalne dužine sekvenci ( $max\_len$ ). Hiperparametar  $C$  u SVM modelu predstavlja regularizacioni koeficijent koji kontroliše kompromis između maksimizacije margine i minimizacije klasifikacionih grešaka, dok  $gamma$  predstavlja koeficijent kernela i funkcioniše kao kontrola glatkoće granice razdvajanja. Hiperparametar  $vector\_size$  iz Word2Vec modela označava dimenzionalnost vektora koji predstavljaju svaku reč, dok se  $max\_len$  odnosi na maksimalnu dužinu vektora reči, poslatog modelu. Najbolji hiperparametri za model koji prepoznaje transakcionu klasu su:  $C = 100$ ,  $gamma = auto$ ,  $vector\_size = 32$  i  $max\_len = 7$ , dok su za model koji prepoznaje delatnost biznisa najbolji:  $C = 100$ ,  $gamma = scale$ ,  $vector\_size = 64$  i  $max\_len = 6$ .

---

## Analiza rezultata modela za klasifikaciju transakcije

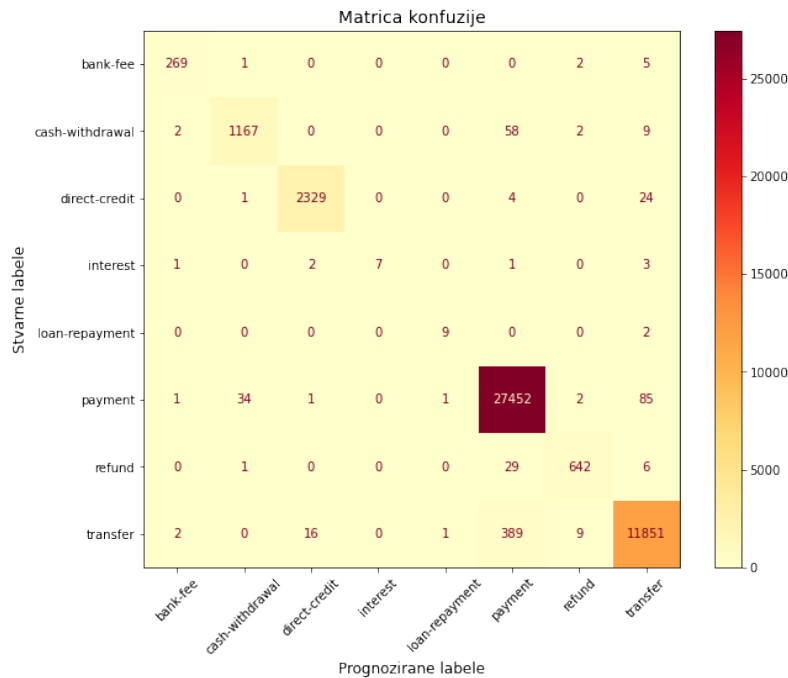
Nakon sprovedenog predprocesiranja podataka, optimizacije hiperparametara i pažljive evaluacije modela, dobili smo konačne rezultate testiranih SVM modela za višeklasnu klasifikaciju tipa transakcije. Koristili smo različite strategije treniranja kako bismo utvrdili koji model pruža najpreciznije i najefikasnije performanse.

Tabela 4.1: Rezultati klasifikacije tipa transakcije na test skupu

Algoritam	Tačnost	Balansirana Tačnost	Vreme obučavanja
Jedan na jedan	98.4%	89.1%	12.52 min
Jedan protiv svih	98.4%	91%	40.11 min
DAGSVM	98.4%	89.9%	9.83 min
Veston i Votkins	94.8%	81.8%	59.75 min
Kramer i Singer	97.3%	88.4%	3.49 min
ECOC	98.4%	87.1%	76.52 min

U tabeli 4.1 vidimo tačnost i balansiranu tačnost za 6 SVM algoritama koji su testirani u ovom radu. Pored metrika koje ukazuju na kvalitet predviđanja, prikazano je i vreme potrebno za obučavanje algoritma kako bismo uporedili i „cenu“ algoritma. Svi algoritmi su postigli prilično dobre rezultate, sa izuzetkom algoritma koji su predstavili Veston i Votkins koji je nekoliko procenata lošiji od ostalih metoda. Metod jedan protiv svih postiže najbolju balansiranu tačnost. Na ovom skupu podataka, najviše vremena je bilo potrebno metodu korekcije grešaka.

Kako bismo detaljnije ispitali razlike među algoritmima, pogledaćemo matrice konfuzije. Na slici 4.8 vidimo da model najčešće greši pri prepoznavanju klasa transfer i plaćanje. U 389 slučaja je klasa transfer bila označena da je plaćanje, a 85 puta je klasa plaćanje bila označena kao transfer. Od ostalih kombinacija, značajna je jedino kombinacija povlačenje novca i plaćanje.



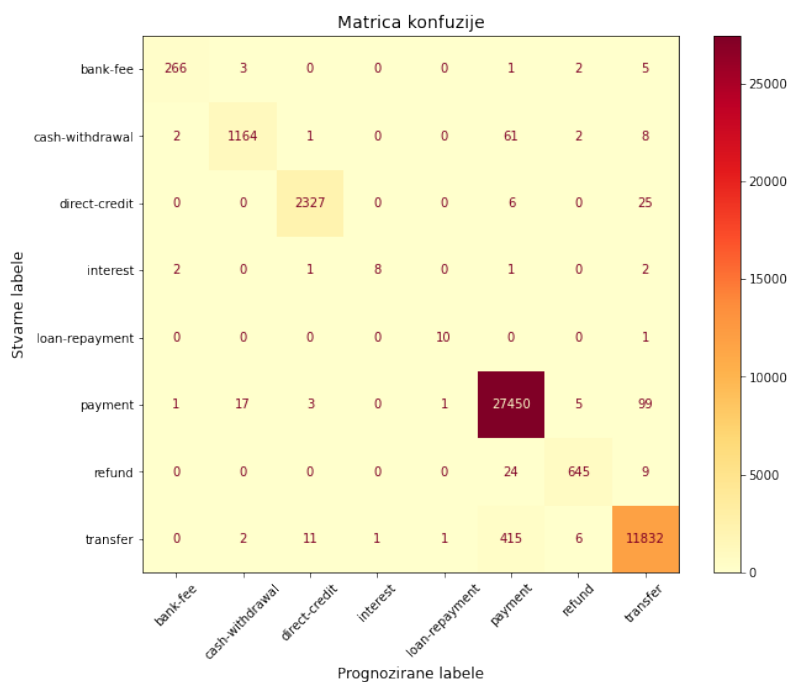
Slika 4.8: Matrica konfuzije jedan na jedan

Ispod vidimo i rezultate klasifikacije po klasama. Odziv je najlošiji pri prepoznavanju kamate zbog malog broja instanci.

	precision	recall	f1-score	support
bank-fee	0.98	0.97	0.97	277
cash-withdrawal	0.97	0.94	0.96	1238
direct-credit	0.99	0.99	0.99	2358
interest	1.00	0.50	0.67	14
loan-repayment	0.82	0.82	0.82	11
payment	0.98	1.00	0.99	27576
refund	0.98	0.95	0.96	678
transfer	0.99	0.97	0.98	12268
accuracy			0.98	44420
macro avg	0.96	0.89	0.92	44420
weighted avg	0.98	0.98	0.98	44420

Slika 4.9: Rezultati klasifikacije po klasama

Kod metoda koji poredi jednu klasu sa svim ostalim, greške se dešavaju na klasama plaćanje i transfer. Ove dve klase su i najbrojnije pa je očekivano da tu vidimo najviše grešaka. Matrica deluje dosta slično matrici kod jedan na jedan algoritma. Broj ispravno klasifikovanih instanci je nešto niži nego kod gorepomenutog algoritma.



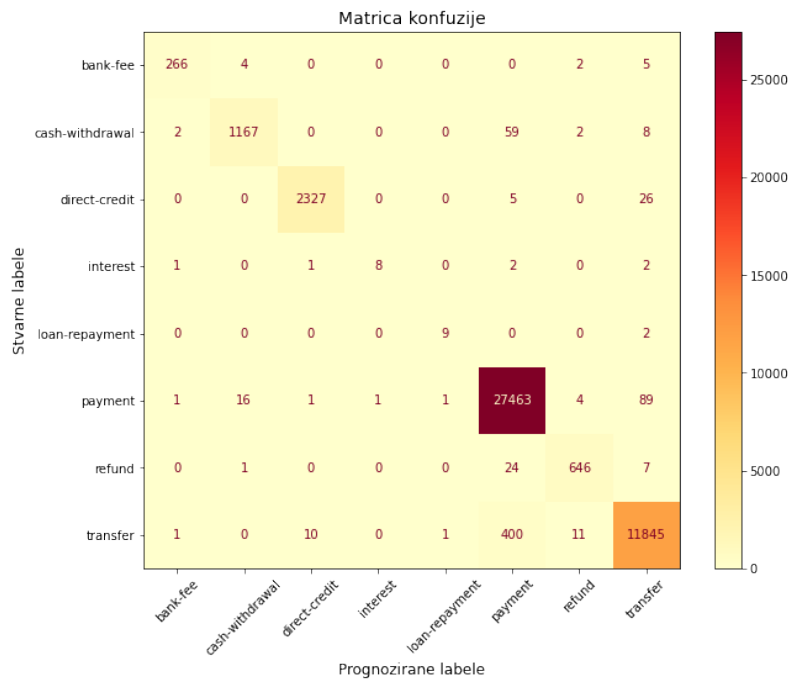
Slika 4.10: Matrica konfuzije jedna protiv svih

	precision	recall	f1-score	support
bank-fee	0.98	0.96	0.97	277
cash-withdrawal	0.98	0.94	0.96	1238
direct-credit	0.99	0.99	0.99	2358
interest	0.89	0.57	0.70	14
loan-repayment	0.83	0.91	0.87	11
payment	0.98	1.00	0.99	27576
refund	0.98	0.95	0.96	678
transfer	0.99	0.96	0.98	12268
accuracy			0.98	44420
macro avg	0.95	0.91	0.93	44420
weighted avg	0.98	0.98	0.98	44420

Slika 4.11: Rezultati klasifikacije po klasama



Na osnovu matrice konfuzije možemo zaključiti da DAGSVM daje malo bolje predikcije od metoda jedna klasa protiv svih ostalih klasa. Ukupan broj tačno prediktovanih instanci je 43731 što je za 5 više od sledećeg najboljeg algoritma jedan na jedan, za 25 bolje od ECOC a za 29 bolje od algoritma jedna klasa protiv svih.

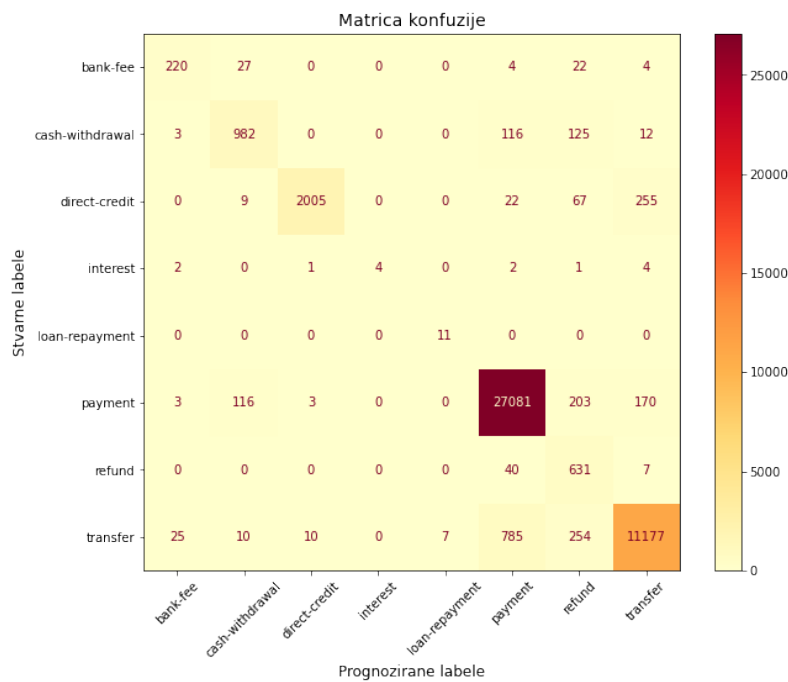


Slika 4.12: Matrica konfuzije DAGSVM

	precision	recall	f1-score	support
bank-fee	0.98	0.96	0.97	277
cash-withdrawal	0.98	0.94	0.96	1238
direct-credit	0.99	0.99	0.99	2358
interest	0.89	0.57	0.70	14
loan-repayment	0.82	0.82	0.82	11
payment	0.98	1.00	0.99	27576
refund	0.97	0.95	0.96	678
transfer	0.99	0.97	0.98	12268
accuracy			0.98	44420
macro avg	0.95	0.90	0.92	44420
weighted avg	0.98	0.98	0.98	44420

Slika 4.13: Rezultati klasifikacije po klasama

Algoritam Vestona i Votkinsa ima nešto drugačije predikcije od prethodna tri modela. Dosta grešaka je napravio pri predikciji klase koja označava podizanje novca, umesto nje prognozirao je klase plaćanje i refundacija novca u najvećem broju slučajeva, redom 116 i 125. S druge strane u 116 slučajeva je za klasu plaćanje prognozirao podizanje novca. Još jedna značajna razlika u odnosu na prethodna tri modela je veći broj ishoda refundacija u slučaju kada je tačna klasa plaćanje ili transfer.

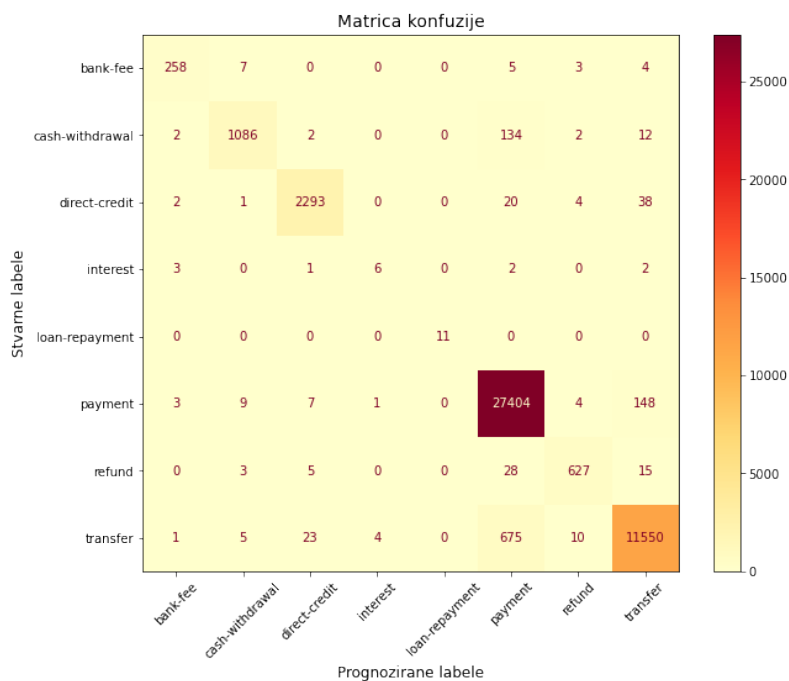


Slika 4.14: Matrica konfuzije Veston Votkins

	precision	recall	f1-score	support
bank-fee	0.87	0.79	0.83	277
cash-withdrawal	0.86	0.79	0.82	1238
direct-credit	0.99	0.85	0.92	2358
interest	1.00	0.29	0.44	14
loan-repayment	0.61	1.00	0.76	11
payment	0.97	0.98	0.97	27576
refund	0.48	0.93	0.64	678
transfer	0.96	0.91	0.94	12268
accuracy			0.95	44420
macro avg	0.84	0.82	0.79	44420
weighted avg	0.95	0.95	0.95	44420

Slika 4.15: Rezultati klasifikacije po klasama

Metod Kramera i Singera na ovom problemu pravi slične greške kao metodi koji višeklasnu klasifikaciju svode na binarnu klasifikaciju. Tačnost je malo lošija od pomenutih metoda.

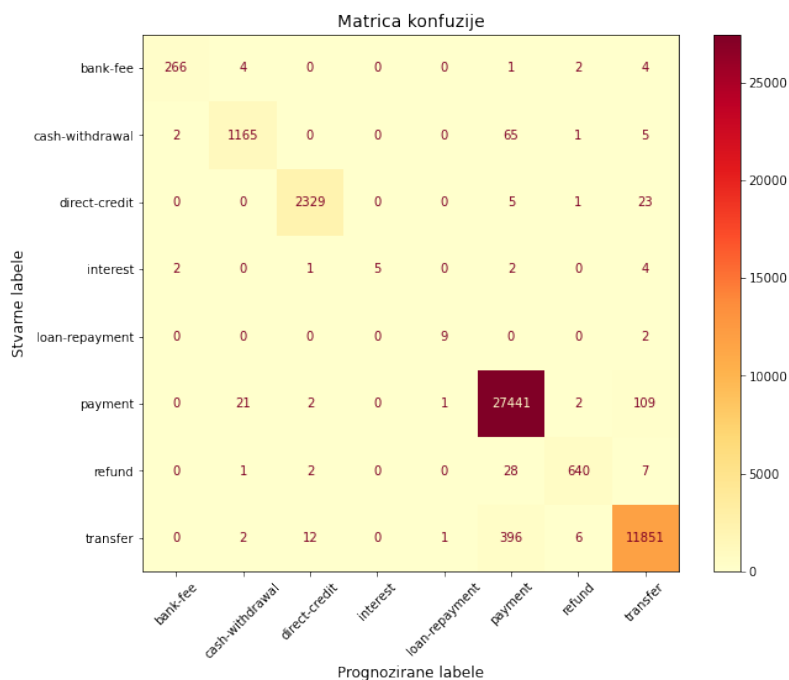


Slika 4.16: Matrica konfuzije Kramer Singer

	precision	recall	f1-score	support
bank-fee	0.96	0.93	0.95	277
cash-withdrawal	0.98	0.88	0.93	1238
direct-credit	0.98	0.97	0.98	2358
interest	0.55	0.43	0.48	14
loan-repayment	1.00	1.00	1.00	11
payment	0.97	0.99	0.98	27576
refund	0.97	0.92	0.94	678
transfer	0.98	0.94	0.96	12268
accuracy			0.97	44420
macro avg	0.92	0.88	0.90	44420
weighted avg	0.97	0.97	0.97	44420

Slika 4.17: Rezultati klasifikacije po klasama

Poslednji metod pokazuje dosta dobre rezultate, pravi slične greške poput zamene plaćanja i transfera ili povlačenja novca i plaćanja.



Slika 4.18: Matrica konfuzije ECOC

	precision	recall	f1-score	support
bank-fee	0.99	0.96	0.97	277
cash-withdrawal	0.98	0.94	0.96	1238
direct-credit	0.99	0.99	0.99	2358
interest	1.00	0.36	0.53	14
loan-repayment	0.82	0.82	0.82	11
payment	0.98	1.00	0.99	27576
refund	0.98	0.94	0.96	678
transfer	0.99	0.97	0.98	12268
accuracy			0.98	44420
macro avg	0.97	0.87	0.90	44420
weighted avg	0.98	0.98	0.98	44420

Slika 4.19: Rezultati klasifikacije po klasama

---

## Analiza rezultata modela za klasifikaciju delatnosti

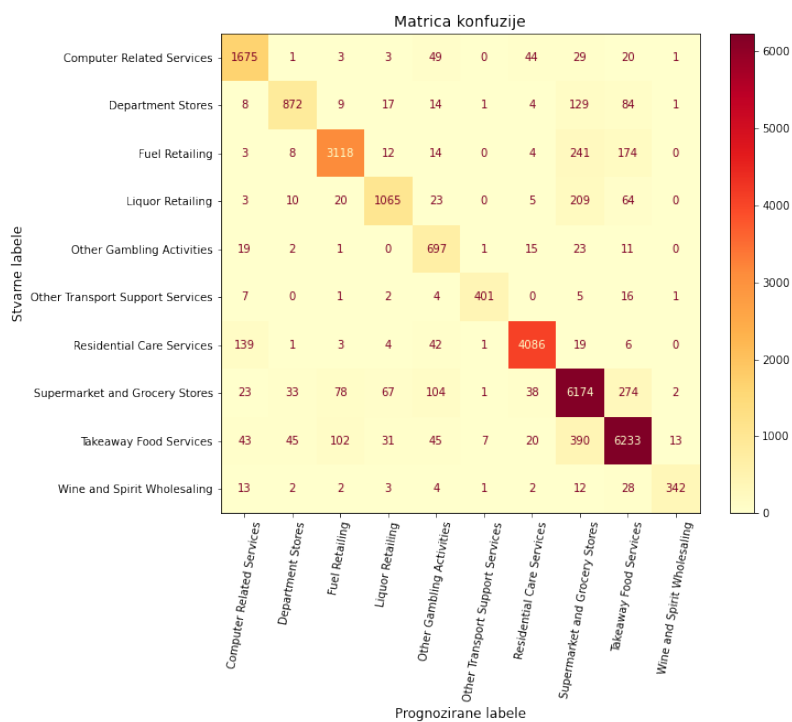
Nakon vizualizacije podataka, pretprocesiranja, optimizacije hiperparametara koristeći GridSearch i evaluacije modela, na test skupu smo dobili rezultate ispitivanih SVM modela za klasifikaciju delatnosti transakcija. Različite metode treniranja omogućile su nam da utvrdimo koji model daje najpreciznije i najefikasnije rezultate.

Tabela 4.2: Rezultati klasifikacije delatnosti na test skupu

Algoritam	Tačnost	Balansirana Tačnost	Vreme obučavanja
Jedan na jedan	89.4%	87.4%	28.2 min
Jedan protiv svih	90.1%	87.7%	116.17 min
DAGSVM	89.3%	87.1%	20.53 min
Veston i Votkins	82%	78.1%	36.06min
Kramer i Singer	83.8%	79.5%	10.78 min
ECOC	89.1%	85.4%	309.06 min

U tabeli 4.2 su prikazani rezultati predviđanja delatnosti. Tačnost je nešto lošija nego kod prethodnog problema zbog povećanja broja klasa i njihove veće izbalansiranosti. Ponovo je jedan protiv svih algoritam pokazao najbolji rezultat, ovaj put se to vidi i iz tačnosti, za razliku od prethodnog problema gde je nekoliko algoritama imalo istu tačnost na 98.4%. Veston i Votkins algoritam ponovo ima najlošiji rezultat. Ponovo najviše vremena za obučavanje treba algoritmu za korekciju grešaka. Primetimo da su kod svih metoda, osim metoda Vestona i Votkinsa, vremena obučavanja duža nego u prethodnom problemu klasifikacije kada smo imali više instanci. Razlog za to je što sada imamo više klasa.

Na slici 4.20 vidimo da su najbrojnije delatnosti prodaja u supermarketu i poručivanje hrane. Posledično najviše grešaka se dešava pri zameni ove dve klase. Dosta grešaka model pravi pri prepoznavanju prodaje goriva i alkohola, klasifikujući to pogrešno kao prodaja u opštim prodavnicama namirnica. Česta greška je i klasifikacija prodaje goriva kao poručivanje hrane.



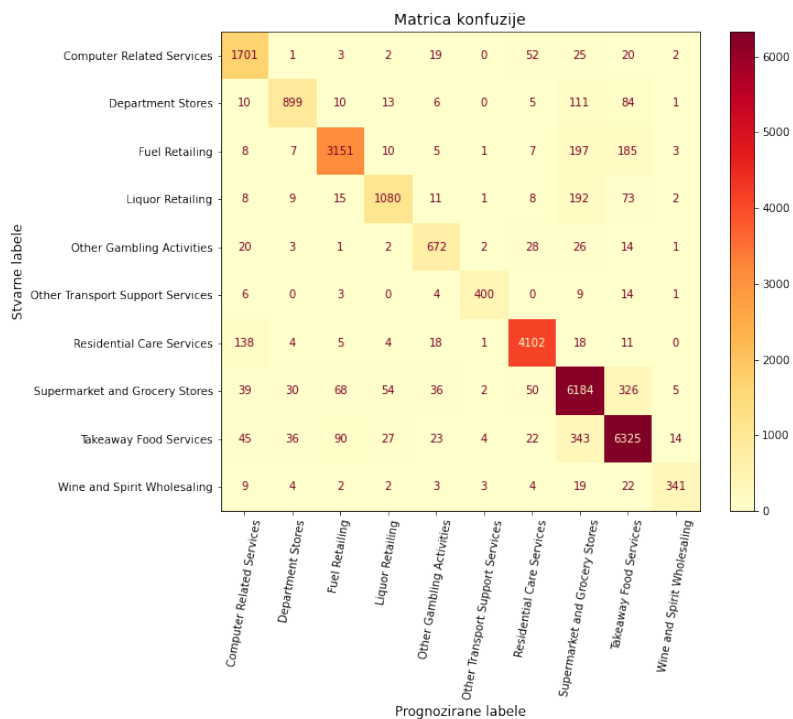
Slika 4.20: Matrica konfuzije jedan na jedan

Iz pregleda klasifikacije po klasama vidimo da model najbolje prepoznaje delatnost rezidencijalni servisi, računarski servisi i transport, a najlošije prodaju alkohola.

	precision	recall	f1-score	support
Computer Related Services	0.87	0.92	0.89	1825
Department Stores	0.90	0.77	0.83	1139
Fuel Retailing	0.93	0.87	0.90	3574
Liquor Retailing	0.88	0.76	0.82	1399
Other Gambling Activities	0.70	0.91	0.79	769
Other Transport Support Services	0.97	0.92	0.94	437
Residential Care Services	0.97	0.95	0.96	4301
Supermarket and Grocery Stores	0.85	0.91	0.88	6794
Takeaway Food Services	0.90	0.90	0.90	6929
Wine and Spirit Wholesaling	0.95	0.84	0.89	409
accuracy			0.89	27576
macro avg	0.89	0.87	0.88	27576
weighted avg	0.90	0.89	0.89	27576

Slika 4.21: Rezultati klasifikacije po klasama

Algoritam jedan protiv svih postiže malo bolje rezultate od prethodnog algoritma. Pravi slične greške, ali na manjem broju slučajeva.

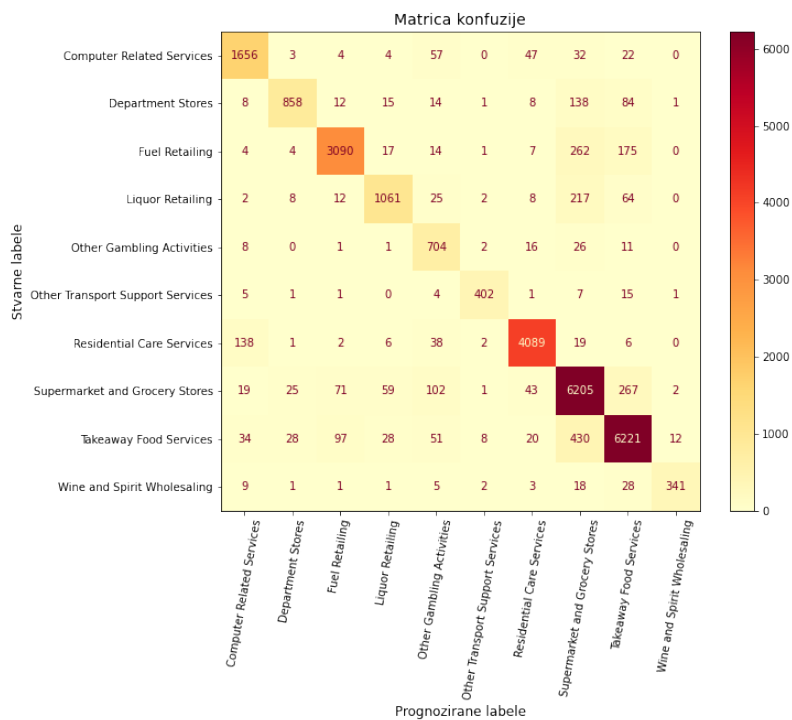


Slika 4.22: Matrica konfuzije jedna protiv svih

	precision	recall	f1-score	support
Computer Related Services	0.86	0.93	0.89	1825
Department Stores	0.91	0.79	0.84	1139
Fuel Retailing	0.94	0.88	0.91	3574
Liquor Retailing	0.90	0.77	0.83	1399
Other Gambling Activities	0.84	0.87	0.86	769
Other Transport Support Services	0.97	0.92	0.94	437
Residential Care Services	0.96	0.95	0.96	4301
Supermarket and Grocery Stores	0.87	0.91	0.89	6794
Takeaway Food Services	0.89	0.91	0.90	6929
Wine and Spirit Wholesaling	0.92	0.83	0.88	409
accuracy			0.90	27576
macro avg	0.91	0.88	0.89	27576
weighted avg	0.90	0.90	0.90	27576

Slika 4.23: Rezultati klasifikacije po klasama

Model dobijen korišćenjem DASVM-a ima slične performanse kao prethodna dva modela.



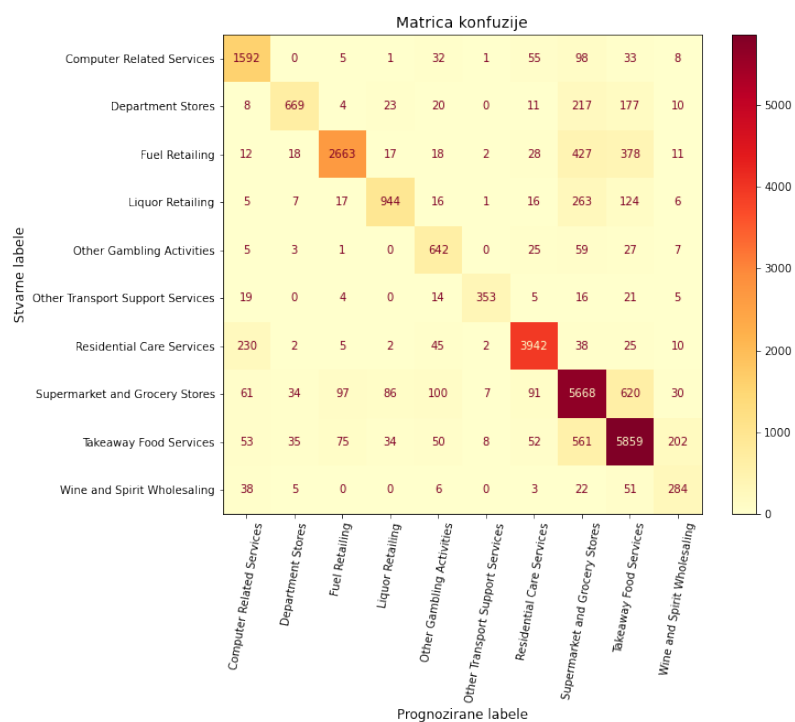
Slika 4.24: Matrica konfuzije DAGSVM

	precision	recall	f1-score	support
Computer Related Services	0.88	0.91	0.89	1825
Department Stores	0.92	0.75	0.83	1139
Fuel Retailing	0.94	0.86	0.90	3574
Liquor Retailing	0.89	0.76	0.82	1399
Other Gambling Activities	0.69	0.92	0.79	769
Other Transport Support Services	0.95	0.92	0.94	437
Residential Care Services	0.96	0.95	0.96	4301
Supermarket and Grocery Stores	0.84	0.91	0.88	6794
Takeaway Food Services	0.90	0.90	0.90	6929
Wine and Spirit Wholesaling	0.96	0.83	0.89	409
accuracy			0.89	27576
macro avg	0.89	0.87	0.88	27576
weighted avg	0.90	0.89	0.89	27576

Slika 4.25: Rezultati klasifikacije po klasama



Algoritam Vestona i Votkinsa i na ovom skupu podataka postiže najlošije rezultate. Najbolji odziv ima klasa rezidencijalni servisi.

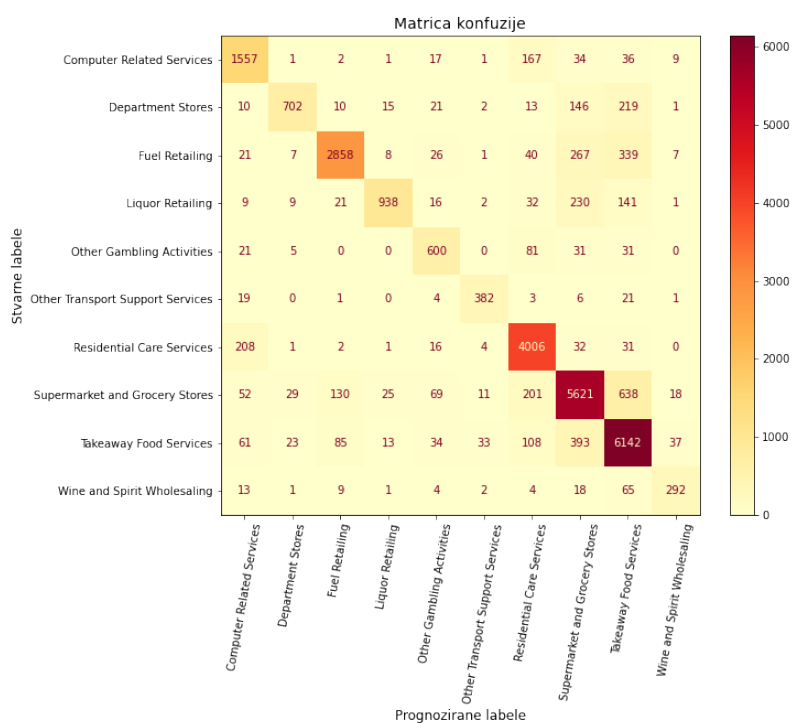


Slika 4.26: Matrica konfuzije Veston Votkins

	precision	recall	f1-score	support
Computer Related Services	0.79	0.87	0.83	1825
Department Stores	0.87	0.59	0.70	1139
Fuel Retailing	0.93	0.75	0.83	3574
Liquor Retailing	0.85	0.67	0.75	1399
Other Gambling Activities	0.68	0.83	0.75	769
Other Transport Support Services	0.94	0.81	0.87	437
Residential Care Services	0.93	0.92	0.92	4301
Supermarket and Grocery Stores	0.77	0.83	0.80	6794
Takeaway Food Services	0.80	0.85	0.82	6929
Wine and Spirit Wholesaling	0.50	0.69	0.58	409
accuracy			0.82	27576
macro avg	0.81	0.78	0.79	27576
weighted avg	0.83	0.82	0.82	27576

Slika 4.27: Rezultati klasifikacije po klasama

Metod Kramera i Singera na ovom problemu ima lošiji performans u odnosu na druge algoritme, za razliku od prethodnog skupa podataka gde je imao približno slične performanse. Najlošiji rezultat postiže pri prepoznavanju klase koja označava prodaju u robnim kućama.

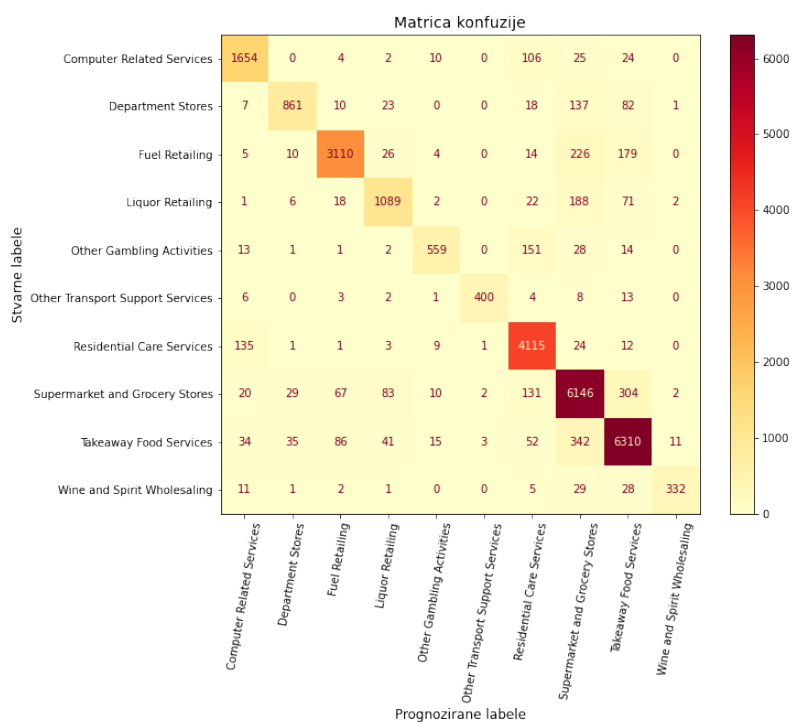


Slika 4.28: Matrica konfuzije Kramer Singer

	precision	recall	f1-score	support
Computer Related Services	0.79	0.85	0.82	1825
Department Stores	0.90	0.62	0.73	1139
Fuel Retailing	0.92	0.80	0.85	3574
Liquor Retailing	0.94	0.67	0.78	1399
Other Gambling Activities	0.74	0.78	0.76	769
Other Transport Support Services	0.87	0.87	0.87	437
Residential Care Services	0.86	0.93	0.89	4301
Supermarket and Grocery Stores	0.83	0.83	0.83	6794
Takeaway Food Services	0.80	0.89	0.84	6929
Wine and Spirit Wholesaling	0.80	0.71	0.75	409
accuracy			0.84	27576
macro avg	0.84	0.80	0.81	27576
weighted avg	0.84	0.84	0.84	27576

Slika 4.29: Rezultati klasifikacije po klasama

Metod korekcije grešaka pokazuje solidne performanse. Najlošiji rezultat postiže pri prepoznavanju klase koja označava kockarske aktivnosti.



Slika 4.30: Matrica konfuzije ECOC

	precision	recall	f1-score	support
Computer Related Services	0.88	0.91	0.89	1825
Department Stores	0.91	0.76	0.83	1139
Fuel Retailing	0.94	0.87	0.90	3574
Liquor Retailing	0.86	0.78	0.82	1399
Other Gambling Activities	0.92	0.73	0.81	769
Other Transport Support Services	0.99	0.92	0.95	437
Residential Care Services	0.89	0.96	0.92	4301
Supermarket and Grocery Stores	0.86	0.90	0.88	6794
Takeaway Food Services	0.90	0.91	0.90	6929
Wine and Spirit Wholesaling	0.95	0.81	0.88	409
accuracy			0.89	27576
macro avg	0.91	0.85	0.88	27576
weighted avg	0.89	0.89	0.89	27576

Slika 4.31: Rezultati klasifikacije po klasama

## Glava 5

# Zaključak

U ovom radu istraženi su različiti načini uopštavanja metoda potpornih vektora na višeklasni slučaj. Predstavljani su pristupi jedan na jedan, jedan protiv svih, DAGSVM, metod Vestona i Votkinsa, metod Kramer Singera i metod korekcije grešaka.

Svaki od pristupa višeklasnog metoda potpornih vektora ima svoje prednosti i mane. Metoda jedan protiv svih je jednostavna za implementaciju i pogodna za manje skupove podataka, ali može biti manje efikasna kada su klase neuravnotežene. Nasuprot tome, jedan na jedan često nudi veću tačnost, iako može biti računski skup kada postoji veliki broj klasa. DAGSVM omogućava brže donošenje odluka smanjenjem broja potrebnih poređenja. Metoda Veston-Votkins je računarski efikasna, ali može imati izazova sa nebalansiranim klasama. Metod Kramera i Singera pruža dobre performanse kroz jedinstveni optimizacijski okvir koji uključuje sve klase, ali može biti dosta spor. Metod korekcije grešaka povećava otpornost i tačnost modela, ali zahteva više resursa za obradu. Ovi pristupi nude različite odnose između tačnosti i računске efikasnosti, stoga se izbor često bazira na specifičnim potrebama primene i skupu podataka.

U ovom radu, svi ovi algoritmi korišćeni su na bankarskim transakcijama za klasifikaciju vrste transakcije, a zatim i na klasifikaciji delatnosti u slučaju kada je klasa bila plaćanje. Metoda jedan protiv svih je pokazala odlične rezultate, dok je Veston Votkins pristup imao najlošije performanse u obe klasifikacije. Najviše vremena za izvršenje je bilo potrebno metodi za korekciju grešaka.

# Bibliografija

- [1] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 1(Dec):113–141, 2000.
- [2] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [3] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.
- [4] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.
- [5] Emily Cubides and Shaun O’Brien. 2023 findings from the diary of consumer payment choice. *Federal Reserve Bank of San Francisco*, 2023.
- [6] Thomas G Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286, 1994.
- [7] Federico D’Onofrio, Giorgio Grani, Marta Monaci, and Laura Palagi. Margin optimal classification trees. *Computers & Operations Research*, 161:106441, 2024.
- [8] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.
- [9] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines (preliminary long abstract). Technical report, Citeseer, 2001.

- [10] Chih-Jen Lin. A formal analysis of stopping criteria of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 13(5):1045–1052, 2002.
- [11] Neptune AI. Balanced accuracy. <https://neptune.ai/blog/balanced-accuracy>, 2024.
- [12] Mahesh Pal. Multiclass approaches for support vector machine based land cover classification. *arXiv preprint arXiv:0802.2411*, 2008.
- [13] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.
- [14] John Platt, Nello Cristianini, and John Shawe-Taylor. Large margin dags for multiclass classification. *Advances in neural information processing systems*, 12, 1999.
- [15] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2018.
- [16] Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.
- [17] Qiuwen Zhang, Yihan Wang, Lixun Huang, Bin Jiang, and Xiao Wang. Fast cu partition decision for h.266/vvc based on the improved dag-svm classifier model. *Multimedia Systems*, 27, 02 2021.

# Biografija autora

Vladimir Luković je rođen 21.11.1995. godine u Beogradu. Završio je osnovnu školu „Jelena Četković“, a zatim upisao Matematičku gimnaziju 2010. godine. Godine 2014. upisuje Matematički fakultet Univerziteta u Beogradu, gde stiče osnovna akademska znanja na modulu „Statistika, aktuarska i finansijska matematika“. Višestruki je prvak države u šahu u svojoj kategoriji, a 2015. godine dobija zvanje Internacionalnog majstora. Od 2019. godine radi u australijskoj kompaniji „Basiq“ na pozicijama vezanim za obradu i modeliranje podataka. U slobodno vreme voli da se bavi kolektivnim sportovima, druži sa prijateljima i provodi vreme u vikendici.