UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



OPTIMALIZÁCIA OBSAHU NA WEBSTRÁNKE POMOCOU KLIENTSKÝCH DÁT A ZÁZNAMOV Z PREHLIADANIA WEBU

DIPLOMOVÁ PRÁCA

2026 Bc. Lukáš KEPAZA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

OPTIMALIZÁCIA OBSAHU NA WEBSTRÁNKE POMOCOU KLIENTSKÝCH DÁT A ZÁZNAMOV Z PREHLIADANIA WEBU

DIPLOMOVÁ PRÁCA

Študijný program:	Ekonomická a finančná matematika a modelovanie
Študijný odbor:	1113 Matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	Mgr. Martin Hurban, PhD.

Bratislava 2026 **Bc. Lukáš KEPAZA**





Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Lukáš Kepaza
Študijný program: ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Optimalizácia obsahu na webstránke pomocou klientskych dát a záznamov z prehliadania webu
Optimization of web content by combining client data and client browsing data

Anotácia: Generovanie relevantného obsahu klientom počas pohybu na webe je úloha, ktorá sa často rieši len pomocou dát, ktoré vznikli priamo v prehliadači. V tejto práci sa budeme zameriavať na kombináciu dát z prehliadača a dát, ktoré sú k dipozícii pre autentifikovaných používateľov. Dizajn takéhoto riešenia bude uvažovať o modeli, ktorý je schopný využiť statické a historické dáta spolu s dátami z konkrétnej návštevy.

Vedúci: Mgr. Martin Hurban, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: doc. Mgr. Igor Melicherčík, PhD.

Dátum zadania: 10.01.2025

Dátum schválenia: 15.01.2025

prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
vedúci práce

Čestne vyhlasujem, že celú diplomovú prácu na tému „Optimalizácia obsahu na webstránke pomocou klientských dát a záznamov z prehliadania webu“, vrátane všetkých jej príloh a obrázkov, som vypracoval samostatne, a to s použitím literatúry uvedenej v priloženom zozname a nástrojov umelej inteligencie. Vyhlasujem, že nástroje umelej inteligencie som použil v súlade s príslušnými právnymi predpismi, akademickými právami a slobodami, etickými a morálnymi zásadami za súčasného dodržania akademickej integrity a že ich použitie je v práci vhodným spôsobom označené.

V Bratislave dňa

Bc. Lukáš Kepaza

Poďakovanie Týmto by som sa chcel poďakovať vedúcemu diplomovej práce Mgr. Martinovi Hurbanovi, PhD., za jeho odborné vedenie, ochotu, cenné rady a konštruktívne pripomienky, ako aj za čas a úsilie venované konzultáciám, ktoré výrazne prispeli k vypracovaniu tejto práce.

Abstrakt v štátnom jazyku

KEPAZA, Lukáš: Optimalizácia obsahu na webstránke pomocou klientských dát a záznamov z prehliadania webu [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: Mgr. Martin Hurban, PhD., Bratislava, 2026, 98s.

Cieľom diplomovej práce je preskúmať prínos použitia webových dát získaných prostredníctvom cookies a interných klientských dát banky pri personalizácii reklamy na webe. V práci kvantifikujeme vplyv dvoch typov dát (cookie dát a klientských dát) a ich interakciu prostredníctvom porovnania modelov založených na jednotlivých typoch dát a ich kombinácii. Zovšeobecnili sme metriky kvantifikujúce kvalitu modelu pre prípad viacnásobných pozorovaní objektu (zákazníka) a pri ich aplikácii sa ukázalo, že rozdiel medzi váženými a neváženými metrikami je minimálny. Kvantifikovali sme efekt cookie dát naprieč viacerými metódami strojového učenia, pričom zlepšenie výkonnosti modelu bolo pozorované najmä pri klasifikačnej metóde LGBM. Pomocou SHAP metódy sa nám podarilo identifikovať relevantné premenné a ich vplyv na predikciu. Súčasťou práce je aj návrh praktického využitia modelov pri personalizácii obsahu na webovej stránke banky. Výsledky práce poukazujú na potenciál kombinácie dátových zdrojov, avšak zároveň zdôrazňujú význam voľby modelu, metódy a charakteru dát pri ich efektívnom využití.

Kľúčové slová: binárna klasifikácia, kombinácia cookie a klientských dát, personalizovaný obsah, vážené metriky predikcie

Abstract

KEPAZA, Lukáš: Optimization of web content by combining client data and client browsing data [Master's thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: Mgr. Martin Hurban, PhD., Bratislava, 2026, 98p.

The aim of this diploma thesis is to examine the contribution of using web data obtained through cookies and the bank's internal client data in the personalization of online advertising. In the thesis, we quantify the impact of two types of data (cookie data and client data) and their interaction by comparing models based on individual data types and their combination. We generalized metrics for evaluating model quality in the case of multiple observations of the same object (customer), and their application showed that the difference between weighted and unweighted metrics is minimal. We quantified the effect of cookie data across several machine learning methods, with performance improvement observed mainly in the LGBM classification method. Using the SHAP method, we were able to identify relevant variables and their influence on predictions. The thesis also includes a proposal for the practical application of these models in personalizing content on the bank's website. The results highlight the potential of combining data sources, while also emphasizing the importance of model selection, methodology, and the nature of the data for their effective use.

Keywords: binary classification, combination of cookie and client data, personalized content, weighted prediction metrics

Obsah

Úvod **10**

1 Použité klasifikačné metódy **12**

1.1 Logistická regresia 13

1.2 Metóda oporných bodov 14

1.3 Metódy založené na rozhodovacích stromoch 16

1.3.1 Rozhodovací strom 16

1.3.2 Náhodny les 19

1.4 Metódy založené na boostingu 21

1.4.1 Boosting stromov 21

1.4.2 Gradient boosting 22

1.4.3 Extreme gradient boosting 24

1.4.4 Light Gradient Boosting Machine 25

2 Kvantifikácia kvality klasifikačných modelov **29**

2.1 Metriky predikcie 29

2.1.1 Accuracy 29

2.1.2 Konfúzna matica 30

2.1.3 ROC, AUC 30

2.1.4 Precision - recall krivka 32

2.1.5 Lift 32

2.2 Meranie príspevkov premenných 34

2.2.1 Základná myšlienka SHAP 35

2.3 Vážené metriky predikcie 36

3 Formulácia problému optimalizácie zobrazovania reklamy **39**

3.1 Cookies 39

3.1.1 Tracking 40

3.1.2 Cookies a tracking z pohľadu banky 42

3.2 Dáta používané v našej práci 43

3.2.1 Charakterizácia cookie dát z bankovej webstránky 44

3.2.2 Deskriptívna štatistika cookie dát z bankovej webstránky 45
3.2.3 Transformácia cookie dát do atribútovej tabuľky 48
3.2.4 Korelačná matica závislých premenných dát attr_web_tracking . . 50
3.2.5 Kódovanie kategorických premenných 51
3.3 Výber dát pre modely strojového učenia 52
3.3.1 Závislé premenné 52
3.3.2 Konštrukcia dátovej vzorky pre modely strojového učenia 52
3.4 Konštrukcia dátových súborov pre modely strojového učenia 53
3.4.1 Cookie model 54
3.4.2 Klientský model 55
3.4.3 Kombinovaný model 55
3.5 Výber finálnych premenných 56

4 Trénovanie klasifikačných modelov 58
4.1 Krosvalidácia a výber finálneho modelu 58
4.2 Výpočtová náročnosť trénovania 59
4.3 Konštrukcia testovacieho dátového súboru 61

5 Kvantifikácia vplyvu cookie dát 62
5.1 Porovnanie výkonnosti modelov 62
5.2 Porovnanie metrík vzhľadom na výsledný počet premenných 67
5.2.1 Porovnanie metrík vzhľadom na výsledný počet pozorovaní 69
5.3 Kvantifikácia cookie premenných v kombinovanom modeli 70
5.3.1 Identifikácia konkrétnych cookie premenných v kombinovanom modeli 72
5.4 Porovnanie vážených a neválených metrík predikcie 74

6 Praktická aplikácia optimalizácie marketingových oslovení v banke 77
6.1 Výber optimálneho klasifikačného modelu 77
6.1.1 Hyperparametre optimálnych modelov 77
6.2 Kalibrácia pravdepodobností modelov 78
6.2.1 Výber cieľovej skupiny klientov 79
6.2.2 Rozhodovacie pravidlo pre výber produktu 80
6.3 Návrh personalizovaného zobrazovania bannerov 81

Záver **85**

Zoznam použitej literatúry **87**

Príloha A **91**

Príloha B **99**

Úvod

V súčasnom digitálnom prostredí zohráva personalizácia obsahu kľúčovú úlohu pri zvyšovaní efektivity online služieb a marketingových aktivít [1]. Vlastníci webových stránok disponujú veľkým množstvom dát o správaní návštevníkov, ktoré vznikajú počas prehliadania obsahu, avšak tieto dáta samy o sebe často neposkytujú úplný obraz o potrebách a zámeroch klienta. Na druhej strane, finančné inštitúcie majú k dispozícii bohaté klientské dáta, ktoré opisujú dlhodobejšie charakteristiky a správanie zákazníkov.

Cieľom tejto diplomovej práce je preskúmať, do akej miery dokáže kombinácia webových dát získaných prostredníctvom cookies a interných klientských dát banky zlepšiť predikciu pravdepodobnosti realizácie vybraného bankového produktu klientom. Práca porovnáva tri prístupy k modelovaniu: model založený výlučne na webových dátach, model využívajúci len bankové klientské dáta a model kombinujúci oba tieto zdroje. Predpokladom je, že prepojenie krátkodobého správania klienta na webe s jeho historickými a statickými charakteristikami povedie k presnejšiemu a prakticky využiteľnejšiemu odhadu nákupného správania.

V praxi sú modely predikcie správania klientov často budované buď výlučne na základe webových dát [2], alebo len na základe interných klientských dát, pričom ich porovnanie a prínos ich kombinácie nie je vždy samozrejmosťou. Podobným smerom ako naša práca sa uberá aj štúdia [3], ktorá analyzuje nákupné správanie zákazníkov v prostredí e-commerce platformy na základe kombinácie transakčných dát (zahŕňajúcich demografické charakteristiky a históriu nákupov) a dát popisujúcich správanie používateľov na webovej stránke. Autori v tejto práci porovnávajú modely využívajúce iba zákaznícke charakteristiky, iba webové správanie a ich kombináciu, pričom ukazujú, že integrácia týchto dvoch zdrojov dát vedie v niektorých prípadoch k zlepšeniu predikcie nákupného správania. V porovnaní s našou prácou však pracujú s výrazne menším počtom premenných (približne päťkrát menej webových a stokrát menej klientských premenných).

Pre kvantifikáciu vplyvu webových dát budeme v našej práci využívať metódy binárnej klasifikácie, v ktorých sa skúma vplyv rôznych množín vysvetľujúcich premenných na kvalitu odhadu pravdepodobnosti sledovanej udalosti, čo predstavuje štandardný prístup v rámci štatistického učenia, podrobne rozpracované v [11, 12]. Výsledky modelov majú potenciálne využitie pri optimalizácii rozhodovacích procesov v oblasti digitálneho

marketingu a personalizácie ponuky bankových produktov na web stránke ČSOB.

Zaoberáme sa problémom, kde máme viac inštancii na jedého klienta podľa toho, koľko rôznych cookies na webovej stránke mal. Uvádzame spôsob ako porovnať výsledky získané pri rôznych úrovniach agregácie. Metódy zaoberajúce sa týmto problémom ako učenie s viacerými inštanciami (angl. multiple instance learning, MIL) [29] však vzhľadom na charakter dostupných dát a definíciu cieľovej premennej neboli v tejto práci aplikovateľné, keďže závislá premenná závisela od klienta a nie od danej cookie.

V kapitole 1 práce sa venujeme teoretickým základom klasifikačných metód a v kapitole 2 spôsobom hodnotenia ich výkonnosti. Predstavujeme vybrané prístupy strojového učenia vhodné pre úlohy binárnej klasifikácie, ktoré budeme následne používať v práci a zároveň uvažujeme metriky používané na ich porovnávanie, s dôrazom na použitie vážených metrík, ktoré zohľadňujú štruktúru dát a význam jednotlivých pozorovaní.

V kapitole 3 formulujeme komerčný problém optimalizácie webového obsahu, predstavíme spôsob fungovania cookies podľa [5, 6] a zároveň popisujeme použité dátové zdroje a konštrukciu dátových súborov, ktoré su vstupom pre modely strojového učenia. Venujeme sa aj spôsobu výberu vysvetľujúcich premenných.

Kapitola 4 sa venuje procesu trénovania modelov, výberu optimálnych modelov, krosvalidácie a konštrukcie testovacích dátových vzoriek určených k hodnoteniu kvality predikčných modelov.

Kapitola 5 predstavuje nosnú časť práce, v ktorej porovnávame výkonnosť jednotlivých modelov a kvantifikujeme prínos cookie dát pre modelovanie preferencií zákazníka.

V záverečnej kapitole 6 sa na problém pozeráme z praktického hľadiska banky, pričom navrhujeme spôsob využitia modelov pre oslovenie cieľovej skupiny klientov marketingovou kampaňou a personalizácii zobrazovaného obsahu na webovej stránke.

1 Použité klasifikačné metódy

Klasifikácia patrí medzi základné metódy strojového učenia. V tejto kapitole uvedieme teoretické poznatky ku klasifikačným metódam, ktoré budeme používať v kapitole 4 pri trénovaní prediktívnych modelov. K získaniu teoretických poznatkov nám pomôžu hlavne knihy [11, 12].

Na rozdiel od regresných problémov, kde je cieľovou premennou kvantitatívna (číselná) hodnota, je pri klasifikácii cieľom predikovať kvalitatívnu (kategorickú alebo binárnu) premennú (napr. farba očí, kúpil/nekúpil), ktorá rozdeluje pozorovania do konečného počtu tried. V tejto práci predikujeme binárnu premennú, ktorá nadobúda hodnoty (1 = kúpil, 0 = nekúpil).

Označme tréningovú množinu pozorovaní

$$(x_1, y_1), \ldots, (x_n, y_n),$$

kde $x_i \in \mathbb{R}^p$ predstavuje vektor hodnôt vysvetľujúcich premenných pre i-te pozorovanie (po prípadnom zakódovaní kategorických premenných[1] do numerickej podoby) a $y_i \in \{0, 1\}$ je identifikátor kategórie, do ktorej i-te pozorovanie patrí. Cieľom klasifikačného modelu je naučiť sa pravidlo, ktoré umožní správne klasifikovať nové, doposiaľ nepozorované dáta.

Mnohé klasifikačné metódy nepriraďujú triedy priamo, ale najskôr odhadujú posteriórne pravdepodobnosti príslušnosti pozorovania k jednotlivým triedam ($k = 0, 1$), teda podmienené pravdepodobnosti vyjadrené ako $P(G = k \mid X = x)$ (pravdepodobnosť, že pozorovanie patrí do triedy k, keď poznáme jeho prediktory $X = x$). G je cieľová (mapujúca) funkcia, ktorá každému vstupu x priradí jeho cieľovú triedu $G(x) \in \{0, 1\}$.

Na základe týchto pravdepodobností je následne pozorovanie klasifikované do triedy s najvyššou odhadnutou pravdepodobnosťou. Tento prístup umožňuje nielen samotnú klasifikáciu, ale aj kvantifikáciu neistoty rozhodnutia.

Priraďovanie pozorovaní do tried na základe posteriórnej pravdepodobnosti vedie k rozdeleniu priestoru vysvetľujúcich premenných na oblasti, v ktorých model priraďuje pozorovaniam rovnakú triedu. Hranice medzi týmito oblasťami sa označujú ako rozhodovacie hranice. Rozhodovacia hranica medzi triedami k a ℓ je definovaná ako množina

[1] Kategorická premenná je premenná nadobúdajúca hodnoty z konečnej množiny kategórií, ktoré reprezentujú kvalitatívne vlastnosti.

bodov $x \in \mathbb{R}^p$, pre ktoré platí $P(G = k \mid X = x) = P(G = \ell \mid X = x)$. Rozhodovacia hranica medzi triedami môže byť lineárna, ak je možné nájsť monotónnu transformáciu posteriórnych pravdepodobností $P(G = k \mid X = x)$, ktorá závisí lineárne od vstupných premenných x. V takom prípade hranica môže byť vyjadrená ako hyperrovina (definovaná v Prílohe A) v priestore $\mathbb{R}^p$, ako je tomu napríklad pri logistickej regresii. Ak taká transformácia neexistuje, hranica je v pôvodnom priestore premenných nelineárna, čoho typickým príkladom sú klasifikačné metódy založené na rozhodovacích stromoch.

V prípade binárnej klasifikácie, mapujúca funkcia G priraďuje každému vstupu x cieľovú triedu $G \in \{0, 1\}$ a sú modelované posteriórne pravdepodobnosti, je často používaným modelom:

$$P(G = 1 \mid X = x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)},$$
$$P(G = 0 \mid X = x) = \frac{1}{1 + \exp(\beta_0 + \beta^T x)}$$

kde $\beta = (\beta_1, \ldots, \beta_p)^T \in \mathbb{R}^p$ je vektor odhadovaných koeficientov modelu. Po aplikácii logitovej transformácie $\text{logit}(p) = \sigma^{-1}(p) = \log\left(\frac{p}{1-p}\right),\ p \in (0, 1)$, kde p predstavuje pravdepodobnosť príslušnosti k jednej z tried, dostávame logaritmus pomeru pravdepodobností (log-odds) v tvare

$$\log \frac{P(G = 1 \mid X = x)}{P(G = 0 \mid X = x)} = \beta_0 + \beta^T x,$$

ktorý je lineárnou funkciou vstupných premenných. Rozhodovacia hranica je potom daná množinou bodov x, pre ktoré je tento výraz rovný nule, teda $\beta_0 + \beta^T x = 0$, čo predstavuje hyperrovinu v priestore vysvetľujúcich premenných.

1.1 Logistická regresia

Medzi najvýznamnejšie metódy, ktoré vedú k lineárnym logaritmom pomeru pravdepodobností patrí logistická regresia.

Ako tvrdí [11, kap. 4.4], pravdepodobnosti nadobúdajú hodnoty v intervale $\langle 0, 1\rangle$ a ich súčet je rovný jednej. Tento prístup je obzvlášť vhodný pre klasifikačné úlohy, v ktorých je cieľovou premennou diskrétna náhodná premenná s konečným počtom tried.

V našej práci sa zaoberáme binárnou logistickou regresiou ($K = 2$), v ktorej nadobúda cieľová premenná len dve hodnoty. V tomto prípade model pre posteriérnu pravdepodob-

nosť príslušnosti k triede 1 má tvar

$$p(x) = P(G = 1 \mid X = x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)} = \frac{1}{1 + \exp(-(\beta_0 + \beta^T x))}. \tag{1}$$

Na porovnanie uvažujeme lineárny regresný model, v ktorom je vzťah medzi vstupnými premennými X a cieľovou premennou modelovaný ako

$$Y = \beta_0 + \beta^T X + \varepsilon.$$

kde $\beta \in \mathbb{R}^p$ je vektor regresných koeficientov a ε je chyba. Použitie lineárneho regresného modelu na odhad takýchto pravdepodobností by mohlo viesť k neplatným predikciám mimo intervalu $\langle 0, 1 \rangle$. Logistická funkcia tento problém eliminuje, keďže jej obor hodnôt je daný ako $H(p) = (0, 1)$ a má charakteristický sigmoidálny tvar (1).

Po úprave modelu dostávame vzťah pre pomer pravdepodobností (angl. odds):

$$\frac{p(x)}{1 - p(x)} = \exp(\beta_0 + \beta^T x), \tag{2}$$

a po aplikácii logaritmu resp. logitovej transformácie

$$\log \frac{p(x)}{1 - p(x)} = \beta_0 + \beta^T x, \tag{3}$$

ktorá je lineárnou funkciou vstupných premenných.

1.2 Metóda oporných bodov

Medzi ďalšie lineárne klasifikačné metódy použité v tejto práci patrí metóda oporných bodov (angl. Support Vector Machine, SVM).

Táto metóda konštruuje rozhodovaciu hranicu medzi triedami tak, aby bola čo najlepšie oddelená a zároveň maximalizovala tzv. *margin*, teda vzdialenosť medzi rozhodovacou hranicou a najbližšími bodmi z oboch tried.

V [11] je SVM definovaná pre binárne triedy označené hodnotami -1 a 1. Keďže v našej práci cieľová premenná nadobúda hodnoty $y_i \in \{0, 1\}$, pre použitie štandardnej formulácie SVM transformujeme naše hodnoty na

$$y_i^* := 2y_i - 1 \in \{-1, 1\}.$$

Lineárna hyperrovina je definovaná ako

$$\{x : f(x) = x^T\beta + \beta_0 = 0\}, \tag{4}$$

a predikcia triedy pre nové pozorovanie x sa vykonáva pomocou sign funkcie (Príloha A):

$$G^*(x) := \text{sign}[f(x)] \in \{-1, 1\}. \tag{5}$$

Predikciu môžeme následne transformovať späť do pôvodnej škály použitím vzťahu

$$G(x) := \frac{G^*(x) + 1}{2} \in \{0, 1\}.$$

Cieľom SVM je nájsť hyperrovinu, ktorá maximalizuje margin medzi triedami. V prípade lineárne separovateľných dát (oddelenie hyperrovinou tak, že všetky body jednej triedy sú na jednej strane a všetky body druhej triedy na druhej strane) sa optimalizácia dá zapísať ako

$$\begin{aligned} &\min_{\beta,\,\beta_0} \|\beta\|_2 \\ &\quad y_i^*(x_i^T\beta + \beta_0) \geq 1, \quad i = 1, \ldots, n, \end{aligned} \tag{6}$$

kde $y_i^* = 2y_i - 1 \in \{-1, 1\}$ je transformovaná cieľová premenná pre účely SVM a šírka marginu je $M = 1/\|\beta\|$.

V praxi sa často stáva, že triedy nie sú úplne separovateľné. V tomto prípade zavádzame tzv. *slack* premenné $\xi_i \geq 0$, ktoré umožňujú niektorým pozorovaniam ležať na nesprávnej strane marginu. Optimalizačný problém sa upraví na

$$\begin{aligned} &\min_{\beta,\,\beta_0,\,\xi} \|\beta\|_2 \\ &\quad y_i^*(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \\ &\quad \xi_i \geq 0, \\ &\quad \sum_{i=1}^{n} \xi_i \leq K, \quad i = 1, \ldots, n, \end{aligned} \tag{7}$$

kde konštanta K obmedzuje celkový počet bodov, ktoré môžu byť nesprávne klasifikované. Body ležiace najbližšie k rozhodovacej hranici (angl. *support vectors*) určujú tvar hranice a majú kľúčový vplyv na model.

Okrem lineárnych rozhodovacích hraníc SVM umožňuje aj nelineárne klasifikácie použitím jadier (angl. *kernels*), ktoré transformujú pôvodný priestor premenných do vyššieho rozmeru, kde môže byť problém lineárne separovateľný.

Takáto transformácia je matematicky ekvivalentná k použitiu rôznych bázových funkcií (napr. polynómy alebo splajny), pričom samotný postup učenia klasifikátora zostáva rovnaký (minimalizuje sa norma vektora koeficientov a maximalizuje margin).

Výsledný klasifikátor stále používa predikciu podľa *sign* funkcie:

$$\hat{G}(x) = \text{sign}(\hat{f}(x)) = \text{sign}\Big(h(x)^T\hat{\beta} + \hat{\beta}_0\Big),$$

kde $h(x)$ predstavuje transformované vstupné premenné.

1.3 Metódy založené na rozhodovacích stromoch

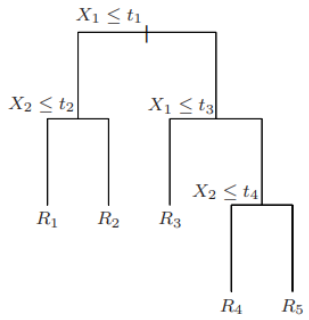
Metódy založené na rozhodovacích stromoch sú ďalšie metódy v našej práci, určené na klasifikačné úlohy a ich princípom je rozdelenie priestoru vysvetľujúcich premenných na viacero jednoduchých oblastí. Podľa [11, kap. 9.2] alebo [12, kap. 8], pre každé pozorovanie sa následne predikcia určuje na základe oblasti, do ktorej patrí, pričom sa typicky používa najčastejšie vyskytujúca sa trieda.

Priestor vysvetľujúcich premenných je rozdeľovaný pomocou rozhodovacích pravidiel, ktoré možno zhrnúť do stromovej štruktúry, odkiaľ pochádza označenie rozhodovacie stromy. Pre každé pozorovanie $x \in \mathbb{R}^p$ sa predikuje cieľová trieda $G(x) \in \{0, 1\}$, do ktorej patrí oblasť, v ktorej sa x nachádza. Rozdelenie priestoru je realizované prostredníctvom rekurzívneho delenia, najčastejšie binárneho, pričom v jednotlivých krokoch sa vyberá premenná a deliaci bod tak, aby sa dosiahla čo najlepšia kvalita modelu.

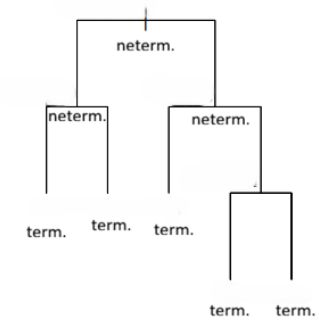
1.3.1 Rozhodovací strom

Výsledkom je rozdelenie priestoru na oblasti $R_1, R_2, \ldots, R_M \subset \mathbb{R}^p$, pričom každému koncovému uzlu rozhodovacieho stromu zodpovedá presne jedna oblasť R_m. Táto oblasť je určená rozhodovacími pravidlami v jednotlivých vnútorných uzloch stromu, ktorými pozorovanie prechádza od koreňového uzla až po príslušný koncový uzol. Rozhodovacie pravidlá majú tvar nerovníc $X_j \leq t$alebo$X_j > t$, kde hodnoty t predstavujú deliace body (konkrétne číselné alebo kategorické hodnoty, ktoré sa vyskytujú v premenných, alebo hodnoty ležiace medzi nimi vysvetľujúcich premenných). Tento vzťah medzi uzlami stromu a oblasťami R_m je ilustrovaný na obr. 1a.

Definícia 1.1 (Uzol)**.** *Uzol v v rozhodovacom strome je základná jednotka stromu, ktorá reprezentuje podmnožinu vstupného priestoru $\mathbb{R}^p$. Každý uzol v obsahuje množinu pozorovaní $S_v \subseteq \mathcal{D}$, kde $\mathcal{D}$ je celý trénovací dátový súbor. Uzol môže byť* rozhodovací (neterminálny) *alebo* koncový (terminálny)*. Každý uzol (okrem koreňového) má práve jedného rodiča.*



(a) Znázornenie oblastí $R_1, R_2, \ldots, R_M$ a uzlov



(b) Znázornenie uzlov (term. = terminálny, neterm. = neterminálny)

Obr. 1: Rozdelenie rozhodovacieho stromu a znázornenie jeho uzlov. Prevzaté z [11]

V regresnom prípade je cieľová premenná v každej oblasti aproximovaná konštantou c_m (predstavuje priemernú hodnotu cieľovej premennej v danej oblasti) a výsledný model má tvar

$$\hat{f}(x) = \sum_{m=1}^{M} c_m \, \mathbb{I}\{x \in R_m\}, \quad x \in \mathbb{R}^p. \tag{8}$$

Kde indikátorová funkcia $\mathbb{I}\{x \in R_m\}$ nadobúda hodnotu 1, ak pozorovanie x patrí do oblasti $R_m \subseteq \mathbb{R}^p$, a hodnotu 0 inak. Hoci sú tieto metódy jednoduché a dobre interpretovateľné, samostatné rozhodovacie stromy často nedosahujú najlepšiu predikčnú presnosť, čo vedie k využívaniu prístupov založených na kombinovaní viacerých stromov.

Pre rozhodovací strom je cieľovou premennou kvalitatívna (kategorická, binárna) premenná $G \in \{0, 1\}$. Pre každé pozorovanie patriace do oblasti R_m predikujeme triedu s najvyššou zastúpenosťou v trénovacej množine:

$$k(m) = \arg\max_k \hat{p}_{mk}, \tag{9}$$

kde $\hat{p}_{mk}$ (používané v klasifikačnom strome namiesto c_m) označuje podiel pozorovaní triedy k v uzle m.

Základnou štruktúrou klasifikačného stromu je štruktúrovaný binárny uzol (delí priestor na dve časti). Rozlišujeme dva typy uzlov (obr. 1b) : *neterminálne uzly*, a *terminálne uzly*, ktoré priraďujú výslednú triedu.

Definícia 1.2 (Neterminálny uzol)**.** *Neterminálny uzol v je uzol, ktorý obsahuje rozhodovacie pravidlo (otázku):* $X_j \leq t alebo X_j > t$*, kde* X_j *je* j*-tá vysvetľujúca premenná a* t *jej deliaci bod. Každý neterminálny uzol má presne dvoch potomkov.*

Definícia 1.3 (Terminálny uzol)**.** *Terminálny uzol v je uzol, ktorý už neobsahuje žiadne ďalšie rozhodovacie pravidlo. Množina pozorovaní* S_v *je priradená k výslednej triede. Uzol sa označí ako terminálny, ak obsahuje pozorovania iba jednej triedy, obsahuje menej než predpísaný minimálny počet trénovacích pozorovaní, alebo bola dosiahnutá maximálna povolená hĺbka stromu. V terminálnom uzle je výsledná trieda určená ako trieda s najvyšším podielom pozorovaní v danom uzle.*

Pri budovaní klasifikačného stromu sa väčšinou používa rekurzívne binárne delenie. V koreňovom uzle sa aplikuje prvé rozhodovacie pravidlo (split) a podľa jeho výsledku sa pozorovanie presúva do jedného z potomkovských uzlov, kde sa proces opakuje, až kým sa nedosiahne terminálny uzol, v ktorom sa určí klasifikácia pozorovania.

Pre každý split sa podľa teórie z [11, kap. 9.2.2] zvažujú všetky prediktory X_j a všetky možné deliace body t. V praxi sa však z dôvodu výpočtovej náročnosti nekontrolujú všetky možné kombinácie t, ale používajú sa heuristiky, ktoré vyberajú deliace body. Pre každý pár (X_j, t) definujeme poloroviny

$$H_1(j,t) = \{X \mid X_j \leq t\}, \qquad H_2(j,t) = \{X \mid X_j > t\}, \tag{10}$$

a vyberieme ten split, ktorý minimalizuje zvolenú mieru impurity uzla (Gini index, entropiu alebo mieru chybovosti) v oboch novovzniknutých uzloch. Tento proces sa opakuje rekurzívne pre všetky vzniknuté oblasti R_m až do splnenia kritéria zastavenia, ako je minimálna veľkosť uzla alebo maximálna povolená hĺbka stromu.

Bežné miery impurity uzla sú:

- **Miera chybovosti:**

$$E_m := 1 - \max_k \hat{p}_{mk}$$

- **Gini index:**

$$G_m := \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- **Entropia:**

$$D_m := -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

Pri dvoch triedach ($K = 2$) tieto miery nadobúdajú tvary

$$1 - \max(p, 1 - p), \qquad 2p(1 - p), \qquad -p \log p - (1 - p) \log(1 - p),$$

pričom p je podiel pozorovaní jednej z tried v danom uzle.

1.3.2 Náhodny les

Používanou metódou založenou na rozhodovacích stromoch v našej práci je náhodny les (angl. Random forest) z [11, kap. 15]. Predstavuje rozšírenie baggingu z [11, kap. 8.7] alebo [12, kap. 8.2] (Bootstrap Aggregating, pri ktorom sa trénuje viac modelov na bootstrapovaných[2] vzorkách dát). Ich predikcie sa spriemerujú, čím sa znižuje rozptyl predikcie[3]. Ak $\hat{f}_b(x)$ je predikcia b-tého modelu z B modelov, priemerná predikcia je $\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x)$. Týmto spôsobom znižuje koreláciu medzi stromami, a tým zlepšuje predikčnú presnosť. Náhodny les podobne vytvára viacero rozhodovacích stromov na bootstrapovaných vzorkách trénovacích dát. Hlavný rozdiel spočíva v tom, že pri každom delení uzla sa vyberie náhodná podmnožina s veľkosťou m z p prediktorov, ktoré môžu byť použité ako kandidáti na split, namiesto toho, aby sa zvažovali všetky p prediktory. Bežnou voľbou je $m \approx \sqrt{p}$.

Tento postup má za cieľ dekorrelovať jednotlivé stromy: ak by existoval jeden veľmi silný prediktor, všetky stromy baggingu by ho používali na hornom delení a stromy by boli veľmi podobné. Náhodný výber podmnožiny prediktorov umožňuje, aby iné premenné dostali šancu, a tým sa zníži korelácia medzi stromami. Nižšia korelácia medzi stromami vedie k výraznejšiemu zníženiu rozptylu priemernej predikcie zo všetkých stromov.

[2]Bootstrapovaná vzorka je náhodne vybraná z pôvodného dátového súboru, pričom každé pozorovanie môže byť vybrané viackrát alebo nemusí byť vybrané vôbec.

[3]Znižovanie rozptylu predikcie nastáva za predpokladu, že jednotlivé modely vykazujú vysoký rozptyl a ich predikcie nie sú dokonale korelované.

Veta 1.1 (Rozptyl predikcie náhodného lesa)**.** *Nech $T(x;\Theta_b)$ označuje predikciu b-tého rozhodovacieho stromu v bode x, pričom jednotlivé stromy sú generované náhodným mechanizmom reprezentovaným náhodnou premennou Θ_b (bootstrapovanie tréningovej množiny).*

Predikcia náhodného lesa pozostávajúceho z B stromov je definovaná ako aritmetický priemer

$$\hat{f}_{\mathrm{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b).$$

Predpokladajme, že:

(i) všetky stromy majú rovnaký rozptyl

$$\mathrm{Var}(T(x;\Theta_b)) = \sigma^2,$$

kde σ^2 vyjadruje variabilitu predikcie jednotlivého stromu,

(ii) pre každé $b \neq b'$ je korelačný koeficient medzi $T(x;\Theta_b)$ a $T(x;\Theta_{b'})$ rovný ρ, t. j.

$$\mathrm{Corr}(T(x;\Theta_b), T(x;\Theta_{b'})) = \frac{\mathrm{Cov}(T(x;\Theta_b), T(x;\Theta_{b'}))}{\sigma^2} = \rho.$$

Potom rozptyl predikcie náhodného lesa je daný vzťahom

$$\mathrm{Var}\left(\hat{f}_{\mathrm{rf}}^{B}(x)\right) = \sigma^2\left(\rho + \frac{1-\rho}{B}\right).$$

Dôkaz. Keďže $\hat{f}_{\mathrm{rf}}^{B}(x)$ je lineárna kombinácia náhodných premenných, z vlastnosti rozptylu lineárnej kombinácie dostávame

$$\mathrm{Var}\left(\hat{f}_{\mathrm{rf}}^{B}(x)\right) = \mathrm{Var}\left(\frac{1}{B}\sum_{b=1}^{B} T(x;\Theta_b)\right) = \frac{1}{B^2}\sum_{b=1}^{B}\sum_{b'=1}^{B} \mathrm{Cov}(T(x;\Theta_b), T(x;\Theta_{b'}))\,.$$

Z predpokladu (i) vyplýva, že pre $b = b'$ platí

$$\mathrm{Cov}(T(x;\Theta_b), T(x;\Theta_b)) = \mathrm{Var}(T(x;\Theta_b)) = \sigma^2.$$

Z definície korelačného koeficientu v (ii) vyplýva, že pre $b \neq b'$ platí

$$\mathrm{Cov}(T(x;\Theta_b), T(x;\Theta_{b'})) = \rho\sigma^2.$$

Dvojitý súčet teda rozdelíme na diagonálne členy (B) a nediagonálne členy ($B(B-1)$). Teda

$$\mathrm{Var}\left(\hat{f}_{\mathrm{rf}}^{B}(x)\right) = \frac{1}{B^2}\left(B\sigma^2 + B(B-1)\rho\sigma^2\right).$$

Po algebraickej úprave dostávame

$$= \sigma^2 \left(\frac{1}{B} + \rho \frac{B-1}{B} \right) = \sigma^2 \left(\rho + \frac{1-\rho}{B} \right).$$

Tým je tvrdenie dokázané. □

Z tohto vzťahu vyplýva, že zvyšovanie počtu stromov B znižuje rozptyl len do tej miery, do akej sú jednotlivé stromy slabo korelované. Práve náhodný výber podmnožiny prediktorov pri každom delení uzla znižuje koreláciu ρ medzi stromami, a tým umožňuje výraznejšiu redukciu rozptylu výslednej predikcie.

1.4 Metódy založené na boostingu

Metódy založené na *boostingu* predstavujú ďalšiu skupinu ensemble[4] metód používaných pri klasifikácii a regresii. Boosting podľa [11, kap. 10] patrí medzi najsilnejšie prístupy v strojovom učení, ktoré boli predstavené v posledných dvadsiatich rokoch. Pôvodne bol navrhnutý pre klasifikačné úlohy, no dá sa úspešne rozšíriť aj na regresiu. Motiváciou boostingu je kombinácia výstupov viacerých slabých klasifikátorov do jedného silného výkonného modelu.

Slabý model je taký model, ktorého predikčná presnosť je len mierne lepšia ako náhodné hádanie. Boosting postupne pridáva tieto slabé modely, pričom každý nasledujúci model sa zameriava na ťažké prípady (pozorovania, kde predchádzajúce modely zlyhali).

1.4.1 Boosting stromov

Samostatné rozhodovacie stromy často neposkytujú dostatočnú predikčnú presnosť, čo viedlo k vývoju metód založených na kombinovaní viacerých stromov. Boosting stromov predstavuje jednu z najvýznamnejších tried týchto metód a tvorí základ viacerých moderných algoritmov.

V rovnici (8) sme rozhodovací strom zapísali pomocou oblastí R_m. Pri vysvetľovaní boostingových metód je však výhodné zapísať strom mierne odlišným spôsobom. Všeobecne vieme zapísať v tvare

$$T(x; \Theta) = \sum_{j=1}^{J} \gamma_j \, I(x \in R_j), \tag{11}$$

[4]Ensemble znamená spájanie viacero slabších modelov do jedného výkonnejšieho.

kde R_j, $j = 1, \ldots, J$, označujú disjunktné[5] oblasti priestoru (výsledné terminálne uzly stromu), γ_j sú konštanty priradené jednotlivým oblastiam a $I(\cdot)$ je indikátorová funkcia. Parameter $\Theta = \{R_j, \gamma_j\}_{j=1}^{J}$ charakterizuje štruktúru stromu aj hodnoty v jeho terminálnych uzloch, pričom počet oblastí J je zvyčajne považovaný za parameter modelu.

Na základe tohto zápisu možno boosting stromov chápať ako postupné budovanie aditívneho modelu[6], v tomto prípade jednotlivých stromov. Boostingový model po M krokoch má tvar

$$f_M(x) = \sum_{m=1}^{M} T(x; \Theta_m), \tag{12}$$

kde $T(x; \Theta_m)$ označuje m-ty klasifikačný strom a Θ_m sú jeho parametre. Každý nový strom je do modelu pridávaný sekvenčne (jeden po druhom) s cieľom zlepšiť predikciu predchádzajúcich stromov.

1.4.2 Gradient boosting

Gradientný boosting možno chápať ako numerickú optimalizačnú metódu na hľadanie funkcie $f(x)$, ktorá minimalizuje zvolenú stratovú funkciu. Nech je celková strata modelu na trénovacích dátach definovaná ako

$$\mathcal{L}(f) := \sum_{i=1}^{N} L\Big(y_i, f(x_i)\Big), \tag{13}$$

kde $L(y, f(x))$ je diferenciovateľná stratová funkcia (pre našu prácu platí $L : \{0, 1\} \times \mathbb{R} \to \mathbb{R}$) a $f(x)$ predstavuje výsledný model.

Definícia 1.4 (Stratová funkcia)**.** *Stratová funkcia $L(y, f(x))$ meria, ako dobre predikcia $f(x)$ zodpovedá skutočnej hodnote y pre každé pozorovanie (x, y). Pre trénovaciu množinu $\{(x_i, y_i)\}_{i=1}^{N}$, kde $x_i \in \mathbb{R}^p$ sú vysvetľujúce premenné a $y_i \in \{0, 1\}$ je cieľová trieda, je celková strata modelu definovaná ako $\mathcal{L}(f) := \sum_{i=1}^{N} L(y_i, f(x_i))$. Používané stratové funkcie pre binárnu klasifikáciu zahŕňajú:*

- ***Chybovosť (misclassification loss):*** $L(y, f(x)) := \mathbf{1}\{sign(f(x)) \neq y\}$,
- ***Binomická deviancia (log-loss):*** $L(y, f(x)) := \log(1 + \exp(-2yf(x)))$,
- ***Kvadratická chyba (squared error):*** $L(y, f(x)) := (y - f(x))^2$.

[5] Disjunktné množiny chápeme ako $R_i \cap R_j = \emptyset$ pre všetky $i \neq j$.

[6] Aditívny znamená súčet viacerých jednoduchších funkcií.

Cieľom učenia je nájsť funkciu $f(x)$ minimalizujúcu túto stratu, pričom $f(x)$ je obmedzená na triedu aditívnych modelov tvorených súčtom stromov. Ak dočasne ignorujeme toto obmedzenie, problém možno chápať ako numerickú optimalizáciu

$$\hat{f} = \arg\min_{f} \mathcal{L}(f), \tag{14}$$

kde optimalizované parametre tvoria hodnoty funkcie $f(x)$ v jednotlivých trénovacích bodoch. Formálne môžeme zapísať

$$\mathbf{f} = \Big(f(x_1), f(x_2), \ldots, f(x_N)\Big)^\top \in \mathbb{R}^N. \tag{15}$$

Praktická implementácia gradientného boostingu prebieha postupne, krok po kroku, pričom každý nový strom sa pridáva do modelu s cieľom zlepšiť predikciu predchádzajúcich stromov. Počiatočný model $f_0(x)$ môže byť jednoduchou konštantou, ktorá minimalizuje zvolenú stratovú funkciu $L(y, f(x))$ na trénovacích dátach:

$$f_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma). \tag{16}$$

V každom kroku $m = 1, \ldots, M$ sa pre každé trénovacie pozorovanie i vyčíslia hodnoty pseudo-reziduí, ktoré predstavujú smer, ktorým by sa predikcia mala upraviť, aby sa minimalizovala strata:

$$r_{im} = - \left. \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right|_{f=f_{m-1}}. \tag{17}$$

Nový strom $T_m(x)$ sa následne učí aproximovať tieto pseudo-reziduá a rozdeľuje priestor vysvetľujúcich premenných na terminálne oblasti R_{jm}, $j = 1, \ldots, J_m$. Každému uzlu sa priradí hodnota γ_{jm}, ktorá optimalizuje lokálnu stratovú funkciu:

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma). \tag{18}$$

Predikcia modelu sa následne aktualizuje pridaním prírastkov z nového stromu:

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}). \tag{19}$$

Pri klasifikačných úlohách s viacerými triedami sa tento postup vykonáva samostatne pre každú triedu, čím vznikajú navzájom závislé stromové reprezentácie, ktoré umožňujú odhad pravdepodobností alebo priamu klasifikáciu pozorovaní.

1.4.3 Extreme gradient boosting

Ďalšou klasifikačnou metódou v našej práci je eXtreme Gradient Boosting (XGBoost). Metóda XGBoost predstavuje modernú a rozšírenú implementáciu gradientného boostingu s rozhodovacími stromami, ktorá je určená pre klasifikačné aj regresné úlohy. Vychádza z princípov gradientného boostingu, pri ktorom je výsledný model konštruovaný ako aditívna kombinácia slabých prediktorov, pričom v prípade XGBoostu sú týmito slabými modelmi rozhodovacie stromy.

Základný tvar modelu v metóde XGBoost je totožný s aditívnym modelom definovaným v rovnici (12).

Hlavným rozdielom podľa [19] oproti klasickému gradientnému boostingu je zavedenie *explicitne regularizovanej cieľovej funkcie*, ktorá okrem chyby predikcie penalizuje aj zložitosť jednotlivých stromov. Regularizovaná cieľová funkcia v metóde XGBoost má tvar

$$\mathcal{L}(f) = \sum_{i=1}^{N} L\Big(y_i, f(x_i)\Big) + \sum_{m=1}^{M} \Omega\Big(T(x;\Theta_m)\Big), \tag{20}$$

kde $L(y_i, f(x_i))$ je diferencovateľná stratová funkcia a $\Omega(T(x;\Theta_m))$ je regularizačný člen penalizujúci zložitosť m-teho stromu. Regularizačný člen má tvar

$$\Omega\Big(T(x;\Theta_m)\Big) = \gamma J_m + \frac{\lambda}{2}\sum_{j=1}^{J_m} \gamma_{jm}^2, \tag{21}$$

kde J_m je počet terminálnych uzlov m-teho stromu, γ_{jm} sú hodnoty priradené jednotlivým uzlom stromu a $\gamma \geq 0$, $\lambda \geq 0$ sú regularizačné parametre modelu. Parameter γ penalizuje počet terminálnych uzlov stromu, zatiaľ čo parameter λ predstavuje regularizáciu váh v jednotlivých listoch stromu, čím obmedzujú ich zložitosť a znižujú riziko preučenia. Pri nulových hodnotách regularizačných parametrov sa cieľová funkcia redukuje na klasický gradientný boosting.

Optimalizácia jednotlivých stromov v XGBoost prebieha aditívne, krok po kroku. Predikcia pre i-tý príklad po $(m-1)$-tom kroku je $f_{m-1}(x_i)$, pričom k tomuto modelu sa pridáva strom $T(x;\Theta_m)$ tak, aby minimalizoval cieľovú funkciu:

$$\mathcal{L}^{(m)} = \sum_{i=1}^{N} L\Big(y_i, f_{m-1}(x_i) + T(x_i;\Theta_m)\Big) + \Omega(T(x;\Theta_m)), \tag{22}$$

Pre výpočet stromu $T(x;\Theta_m)$ sa môže využiť aproximácia, ktorá umožňuje efektívnejšie optimalizovať strom podľa gradientov stratovej funkcie:

$$\mathcal{L}^{(m)} = \sum_{i=1}^{N} \left[g_i T(x_i; \Theta_m) + \frac{1}{2} h_i T(x_i; \Theta_m)^2 \right] + \Omega(T(x; \Theta_m)),$$

kde

$$g_i = \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\Big|_{f(x_i)=f_{m-1}(x_i)}, \quad h_i = \frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\Big|_{f(x_i)=f_{m-1}(x_i)}.$$

Pri učení stromov je kľúčovým problémom nájsť optimálne rozdelenie (split) v každom uzle stromu. XGBoost tento problém rieši kombináciou tzv. presného greedy algoritmu a aproximácií. Presný greedy algoritmus vyhľadáva všetky možné rozdelenia a vyberá to, ktoré v danom kroku najviac znižuje chybu modelu, čo je však výpočtovo náročné. Pre veľké dátové súbory sa preto používajú rôzne aproximácie, viď [19, kap. 3].

Vďaka tejto formulácii je možné efektívne riadiť kompromis medzi presnosťou modelu a jeho komplexnosťou. Tieto vlastnosti robia XGBoost jednou z najvýkonnejších metód založených na boostingu, ktorá dosahuje veľmi dobré výsledky v širokom spektre klasifikačných problémov.

1.4.4 Light Gradient Boosting Machine

Ďalšou metódou založenou na boostingu, ktorú popíšeme po metóde *XGBoost*, je Light Gradient Boosting Machine (LightGBM, LGBM) z [20]. Ide o modernú implementáciu gradientného boostingu rozhodovacích stromov (gradient boosting decision trees, GBDT).

Podobne ako XGBoost, aj LightGBM vychádza z aditívneho modelu definovaného v rovnici (12), kde je výsledný model tvorený súčtom sekvenčne budovaných rozhodovacích stromov optimalizujúcich diferencovateľnú stratovú funkciu. Matematický základ metódy je teda totožný s klasickým gradientným boostingom popísaným v predchádzajúcej časti.

Hlavným cieľom LightGBM je zvýšiť výpočtovú efektívnosť a škálovateľnosť, najmä pri veľkých dátových súboroch s vysokou dimenzionalitou a riedkymi dátami[7]. Na dosiahnutie tohto cieľa LightGBM zavádza dve špecifické techniky: Gradient-based One-Side Sampling (GOSS) a Exclusive Feature Bundling (EFB), ktoré budú podrobnejšie popísané v nasledujúcich podkapitolách.

LightGBM rieši výber optimálnych splitov, čo je výpočtovo najnáročnejším krokom pri trénovaní GBDT. Jednou z často používaných metód je histogram-based algoritmus

[7]Riedke dáta sú dáta obsahujúce veľa núl alebo prazdných hodnôt.

(histogramový algoritmus), ktorý namiesto prehľadávania všetkých hodnôt vysvetľujúcich premenných, rozdeľuje spojité hodnoty do konečného počtu binov (intervalov) a pre tieto biny vytvára histogramy. Pri budovaní stromu sa pre každý uzol a každú premennú spočíta, koľko pozorovaní patrí do jednotlivých binov a aké sú v nich súčty gradientov. Na základe týchto údajov sa určí najlepší split ako hranica medzi dvoma binmi, ktorá najviac zníži stratovú funkciu $L(y, f(x))$.

Tento prístup má výhodu zníženej pamäťovej náročnosti a rýchlosti, keďže počet binov je obyčajne omnoho menší ako počet pozorovaní. Preto je dominantnou časťou výpočtu práve budovanie histogramov, čo ukazuje, že zníženie počtu dát alebo príznakov môže výrazne zrýchliť tréning GBDT.

Gradient-based One-Side Sampling (GOSS)

Keďže dominantnou časťou výpočtu v histogramovom algoritme je budovanie histogramov z veľkého počtu pozorovaní, LightGBM zavádza metódu *Gradient-based One-Side Sampling (GOSS)*, ktorej cieľom je znížiť počet pozorovaní použitých pri výpočte splitov.

GOSS využíva informáciu o veľkosti gradientov každého pozorovania. Pozorovania s väčšími gradientmi predstavujú nie moc dobre natrénované dáta, ktoré majú väčší vplyv na učenie modelu, zatiaľ čo pozorovania s malými gradientmi sú už dobre aproximované. Preto GOSS zachováva všetky pozorovania s veľkými gradientmi a zo zvyšných náhodne vyberá podmnožinu, čím znižuje celkový počet dát potrebných pre výpočet splitov.

Aby sa kompenzoval vplyv redukcie dát na distribúciu, pri výpočte informačného zisku (zníženie stratovej funkcie) sa vybrané pozorovania s malými gradientmi vážia konštantným koeficientom $\frac{1-a}{b}$, kde a je podiel zachovaných pozorovaní s veľkými gradientmi a b je podiel náhodne vybraných malých gradientov. Týmto spôsobom sa zachováva približne pôvodná dátová distribúcia a súčasne sa kladie väčší dôraz na nedobre natrénované vzorky.

Metóda GOSS vychádza z hodnôt pseudo-reziduí definovaných v rovnici (17), teda z negatívnych gradientov stratovej funkcie. Pre jednoduchosť ich budeme označovať g_i. Označme podmnožinu všetkých pozorovaní s veľkými gradientmi A a náhodne vybranú podmnožinu malých gradientov B. Pre každú vysvetľujúcu premennú j a pre možný deliaci bod (split) d sa hodnotí, o koľko sa zníži hodnota stratovej funkcie po vykonaní tohto

splitu. Táto veličina sa označuje ako variačný zisk. Keďže metóda GOSS pracuje iba s vybranou podmnožinou dát, nie je možné vypočítať presný variačný zisk. Namiesto toho sa používa jeho aproximácia, definovaná ako

$$\tilde{V}_j(d) = \frac{\left(\sum_{x_i \in A_l} g_i + \frac{1-a}{b} \sum_{x_i \in B_l} g_i\right)^2}{n_j^l(d)} + \frac{\left(\sum_{x_i \in A_r} g_i + \frac{1-a}{b} \sum_{x_i \in B_r} g_i\right)^2}{n_j^r(d)},$$

kde A_l, A_r, B_l, B_r sú rozdelenia pozorovaní do ľavého (left = l) a pravého (right = r) potomka daného uzla podľa splitu d a n označuje počet pozorovaní v aktuálnom uzle stromu pred vykonaním splitu (n_l, n_r predstavujú počty pozorovaní v ľavom a pravom poduzle pre premennú j po rozdelení). Tento prístup umožňuje určiť split z menšej množiny dát, čím sa výrazne znižuje výpočtová náročnosť.

Exclusive Feature Bundling

Táto časť LightGBM predstavuje techniku redukcie počtu premenných. Využíva sa skutočnosť, že vo väčšine vysoko-dimenzionálnych dátových súborov sú dáta veľmi riedke, čiže mnoho atribútov má hodnotu nula a často sa navzájom nevyskytujú súčasne s nenulovou hodnotou. Takéto atribúty sa nazývajú exkluzívne.

Cieľom EFB je spojiť viacero exkluzívnych atribútov do jedného nového atribútu resp. zhluku (angl. bundle) atribútov bez straty informácií. Namiesto spracovania všetkých jednotlivých atribútov pri tvorbe histogramov stromov, LightGBM teraz spracováva histogram pre každý zhluk atribútov, čím sa znižuje zložitosť z $O(\#\text{dát} \times \#\text{atribútov})$ na $O(\#\text{dát} \times \#\text{zhlukov})$, pričom počet zhlukov je obvykle výrazne menší než počet pôvodných atribútov. Tento prístup významne urýchľuje trénovanie GBDT.

Výber atribútov pre zhlukovanie sa uskutočňuje s cieľom spojiť atribúty do čo najmenšieho počtu zhlukov. LightGBM používa približný postup založený na grafe konfliktov[8]: každý atribút je vrchol a medzi dvoma atribútmi, ktoré sa môžu vyskytovať súčasne, je hrana. Následne sa použije tzv. *greedy algoritmus* (iný ako spomínaný pri XGB), ktorý postupne prechádza atribúty a rozhoduje, či ich priradí do existujúceho zhluku s malým počtom konfliktov, alebo vytvorí nový zhluk. Tento postup efektívne spája atribúty bez výraznej straty presnosti modelu.

[8]Konflikt pri EFB označuje premenné, ktoré majú súčasne veľa spoločných nenulových hodnôt.

Po určení zhlukov je potrebné spojiť pôvodné atribúty tak, aby bolo možné identifikovať ich pôvodné hodnoty pri tvorbe histogramov. Keďže histogramy pracujú s diskrétnymi binmi, EFB pridáva tzv. *offset* k hodnotám jednotlivých atribútov v zhluku, aby sa navzájom neprekrývali.

Príklad na EFB: Na obr. 2a vidíme päť premenných, ktoré upravíme na tri. Vidíme, že P1 a P2 (až na i7) nemajú spoločné nenulové hodnoty. Podobne aj P0 a P3. Tak isto vidno, že napríklad P0 a P1 majú veľa spoločných nenulových hodnôt (konfliktov). Tie premenné, ktoré majú medzi sebou najmenej konfliktov (znázornené grafom konfliktov obr. 2b) označíme ako exkluzívne. Ak sa nastaví algoritmus na maximálny počet konfliktov 1, tak vidno, že P0 ide dokopy s P3 a P1 s P2.

Ďalej treba zistiť, na aké hodnoty sa zmenia vybrané premenné. Ak zoberieme premenné P0 a P3, vidíme, že premenná P0 nadobúda hodnoty od 0 do 3 a P3 nadobúda od 0 do 4. Preto sa pre premennú F3 sa vytvorí offset, kde sa pripočíta 3, teda offset pre P3 bude nadobúdať hodnoty od 3 do 7, a tým pádom nová premenná hodnoty od 0 do 7. Nové hodnoty vytvárame takto:

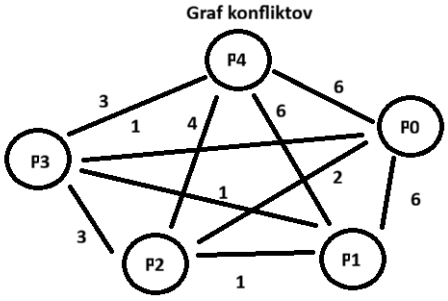
- Ak je prvá hodnota nenulová a druhá nulová, nová premenná nadobúda hodnotu prvej nenulovej.
- Ak je prvá hodnota nulová a druhá nenulová, nová premenná nadobúda hodnotu druhej plus offset.
- Ak sú obe hodnoty nulové, nová premenná nadobúda hodnotu 0.
- Ak sa v dátach vyskytne konflikt (obe hodnoty nenulové), k druhej premennej sa pripočíta offset +3.

	P0	P1	P2	P3	P4
i0	1	1	0	0	1
i1	0	0	1	1	1
i2	1	2	0	0	2
i3	0	0	2	3	1
i4	2	1	0	0	3
i5	3	3	0	0	1
i6	0	0	3	0	2
i7	1	2	3	4	3
i8	1	0	1	0	0
i9	2	3	0	0	2

→

	P0_P3	P1_P2	P4
i0	1	1	1
i1	4	4	1
i2	1	2	2
i3	6	5	1
i4	2	1	3
i5	3	3	1
i6	0	6	2
i7	7	6	3
i8	1	4	0
i9	2	3	2

(a) Bundle premenných



(b) Graf konfliktov

Obr. 2: Príklad na EFB

EFB tak umožňuje výrazné zníženie výpočtovej náročnosti pri zachovaní presnosti modelu.

2 Kvantifikácia kvality klasifikačných modelov

V tejto kapitole uvedieme základne metriky používané na vyhodnocovanie modelov. Hodnotenie výkonu modelov je kľúčové počas tréningu aj validácie modelov, pretože umožňuje kvantifikovať presnosť a spoľahlivosť predikcií. Zameriame sa najmä na klasifikačné metriky, ako sú presnosť (angl. accuracy), konfúzna matica (angl. confusion matrix), ROC (AUC) krivka, krivka presnosti a citlivosti (angl. precision - recall) z [13] a lift. Tieto metriky poskytujú cenné informácie o schopnosti modelu správne rozlišovať medzi triedami a umožňujú optimalizovať rozhodovacie prahy podľa konkrétnych požiadaviek aplikácie.

2.1 Metriky predikcie

Nech máme binárny klasifikačný model, ktorý pre každé pozorovanie x_i v testovacej sade generuje skóre pravdepodobnosti $y_i^{\text{pred}} \in [0, 1]$, že pozorovanie patrí do pozitívnej triedy. Skutočnú triedu označíme $y_i^{\text{true}} \in \{0, 1\}$, kde 1 znamená pozitívny prípad a 0 negatívny.

Aby sme získali binárnu predikciu $\hat{y}_i \in \{0, 1\}$ z pravdepodobnostného odhadu, zvolíme rozhodovací prah (treshold) $\tau \in [0, 1]$ a definujeme predikciu nasledovne:

$$\hat{y}_i = \begin{cases} 1, & \text{ak } y_i^{\text{pred}} > \tau, \\ 0, & \text{ak } y_i^{\text{pred}} \leq \tau. \end{cases}$$

Na základe týchto binárnych predikcií môžeme každú predikciu zaradiť do jednej zo štyroch kategórií:

- Správne pozitívne (angl. true Positives, TP): $\hat{y}_i = 1$ a $y_i^{\text{true}} = 1$,
- Nesprávne pozitívne (angl. false Positives, FP): $\hat{y}_i = 1$ a $y_i^{\text{true}} = 0$,
- Správne negatívne (angl. true Negatives, TN): $\hat{y}_i = 0$ a $y_i^{\text{true}} = 0$,
- Nesprávne negatívne (angl. false Negatives, FN): $\hat{y}_i = 0$ a $y_i^{\text{true}} = 1$.

2.1.1 Accuracy

Accuracy modelu vyjadruje podiel správnych predikcií voči všetkým predikciám a definujeme ju ako

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

Accuracy nemusí byť dobrá, ak je jedna trieda omnoho početnejšia než druhá, pretože model môže predikovať vždy tú väčšiu triedu a aj tak mať vysokú accuracy.

2.1.2 Konfúzna matica

Tieto štyri hodnoty (TP, FP, TN, FN) je bežné prezentovať v 2×2 tabuľke, známom ako konfúzna matica (confusion matrix). Konfúzna matica má nasledujúcu podobu:

	Predikovaná trieda 1	Predikovaná trieda 0
Skutočná trieda 1	TP	FN
Skutočná trieda 0	FP	TN

2.1.3 ROC, AUC

Ďalším spôsobom hodnotenia výkonnosti binárneho klasifikačného modelu je Receiver Operating Characteristic curve (ROC krivka). ROC krivka graficky znázorňuje kompromis medzi dvoma základnými metrikami:

- Miera správne pozitívnych (angl. True Positive Rate, TPR) alebo citlivosť (sensitivity, recall): podiel správne identifikovaných pozitívnych prípadov zo všetkých skutočných pozitívnych prípadov,
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$
- Miera nesprávne pozitívnych (angl. False Positive Rate, FPR): podiel negatívnych prípadov, ktoré boli nesprávne klasifikované ako pozitívne, zo všetkých skutočných negatívnych prípadov,
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

Definícia 2.1 (ROC krivka)**.** *ROC krivka je podľa [18] parametrická krivka definovaná ako* $\mathrm{ROC} := \left\{ \big(\mathrm{FPR}(\tau),\, \mathrm{TPR}(\tau)\big) \;\middle|\; \tau \in \mathbb{R} \right\}$, *kde* τ *je prahová hodnota (threshold). Nech* Y_{pred} *je spojitá náhodná premenná reprezentujúca predikčnú hodnotu klasifikátora. Pre daný threshold* τ *sa pozorovanie klasifikuje ako pozitívne, ak jeho skóre* $Y_{pred} > \tau$*. Hodnoty* Y_{pred} *majú hustotu* $f_1(x)$ *pre skutočne pozitívne triedy (*$Y = 1$*) a hustotu* $f_0(x)$ *pre skutočne negatívne triedy (*$Y = 0$*). Potom platí:*

$$\mathrm{TPR}(\tau) := P(Y^{\mathrm{pred}} > \tau \mid Y = 1) = \int_{\tau}^{1} f_1(x)\,\mathrm{d}x,$$

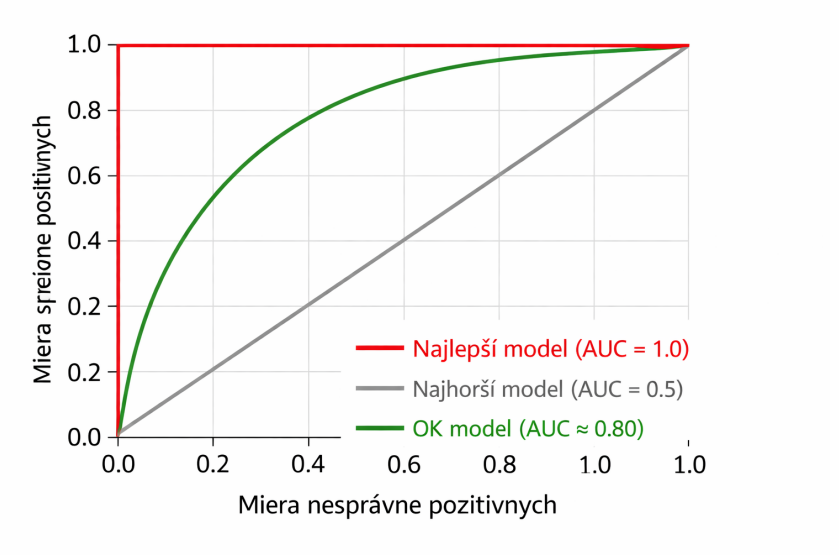
$$\mathrm{FPR}(\tau) := P(Y^{\mathrm{pred}} > \tau \mid Y = 0) = \int_{\tau}^{1} f_0(x)\,\mathrm{d}x.$$

ROC krivka sa vytvára tak, že pre rôzne hodnoty tresholdy τ počítame TPR a FPR a zaznačíme ich ako bod v dvojrozmernom grafe: os x predstavuje FPR, os y predstavuje TPR. Spájaním všetkých bodov pre rôzne tresholdy vzniká krivka, ktorá ukazuje, ako sa mení sensitivita a miera falošne pozitívnych pri rôznych nastaveniach tresholdu. Čím bližšie je ROC krivka k ľavému hornému rohu grafu, tým lepší je model.

Plocha pod krivkou ROC (Area under curve, AUC) udáva, ako dobre model rozlišuje medzi pozitívnymi a negatívnymi prípadmi bez ohľadu na zvolený prah. Formálne je to pravdepodobnosť, že model priradí vyššie skóre náhodne vybranému pozitívnemu prípadu než náhodne vybranému negatívnemu prípadu.

Definícia 2.2 (AUC plocha)**.** *Nech* Y_1^{pred} *označuje hodnotu náhodne zvoleného pozitívneho pozorovania* ($Y = 1$) *a* Y_0^{pred} *náhodne zvoleného negatívneho pozorovania* ($Y = 0$)*. Potom AUC je definované ako* $\text{AUC} := \int_0^1 \text{TPR}\left(\text{FPR}^{-1}(x)\right) \mathrm{d}x$*, alebo v pravdepodobnostnej forme* $\text{AUC} = P\left(Y_1^{\text{pred}} \geq Y_0^{\text{pred}}\right)$.

Na obr. 3 vidíme ROC krivky troch klasifikačných modelov. Na osi y je miera správne pozitívnych (TPR), na osi x miera nesprávne pozitívnych (FPR). Červená krivka predstavuje ideálny model s $AUC = 1$, sivá diagonála znázorňuje náhodný model s $AUC = 0.5$ a zelená krivka zodpovedá modelu s dobrou, no nie ideálnou klasifikačnou schopnosťou ($AUC \approx 0.8$).



Obr. 3: Číselné hodnoty pre najlepší, najhorší a ok model. Vytvorené pomocou [17].

2.1.4 Precision - recall krivka

Ďalším spôsobom hodnotenia výkonu binárneho klasifikačného modelu je Precision-Recall (P-R) krivka, ktorá graficky znázorňuje vzťah medzi dvoma metrikami:

- Precíznosť (Precision): podiel správne predikovaných pozitívnych prípadov zo všetkých predikovaných pozitívnych prípadov,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

- Sensitivita (Recall), popísané vyššie.

Definícia 2.3 (PR krivka)**.** *Precision-Recall krivka je parametrická krivka definovaná ako* $\text{PR} := \left\{ \left(\text{Recall}(\tau), \text{Precision}(\tau) \right) \;\middle|\; \tau \in [0,1] \right\}$, *kde* τ *je prahová hodnota (threshold). Nech* Y^{pred} *je spojitá náhodná premenná (predikovaný pravdepodobnostný odhad klasifikátora). Jedno konkrétne pozorovanie je klasifikované ako pozitívne, ak* $Y^{\text{pred}} > \tau$*. Potom*

$$\text{Recall}(\tau) := P(Y^{\text{pred}} > \tau \mid Y = 1),$$

$$\text{Precision}(\tau) := P(Y = 1 \mid Y^{\text{pred}} > \tau).$$

Ako pri ROC krivke, aj pri PR krivke hodnota tresholdu τ určuje konkrétny bod na krivke, ale nemení samotnú krivku.

AUCPRC (plocha pod PR krivkou) vzniká tak, že pre všetky možné hodnoty tresholdu τ vypočítame precision a recall a tieto body zakreslíme do grafu. AUCPRC je potom plocha pod touto krivkou, ktorá sumarizuje schopnosť modelu rozlišovať medzi pozitívnymi a negatívnymi prípadmi nezávisle od konkrétneho tresholdu.

2.1.5 Lift

Okrem štandardných klasifikačných metrík, ako sú accuracy, ROC krivka alebo Precision–Recall krivka, je v praxi často využívaná aj metrika lift, ktorá hodnotí schopnosť modelu prioritizovať pozorovania s vysokou pravdepodobnosťou pozitívneho výsledku. Lift je obzvlášť vhodný v aplikáciách, kde je cieľom vybrať len obmedzenú časť populácie, napríklad v marketingových kampaniach alebo pri hodnotení rizika.

Postup výpočtu liftu je založený na zoradení pozorovaní podľa predikovanej pravdepodobnosti a na analýze podielu skutočne pozitívnych prípadov v postupne sa rozširujúcich podmnožinách dát.

Najprv zoradíme všetky pozorovania zostupne podľa hodnoty y_i^{pred}, teda od najvyššej po najnižšiu predikovanú pravdepodobnosť pozitívneho výsledku. Pre zvolený threshold definujeme množinu vybraných pozorovaní ako $S(\tau) := \{i \mid y_i^{pred} \geq \tau\}$.

Táto množina obsahuje všetky pozorovania, ktoré model považuje za dostatočne pravdepodobne pozitívne pri danom nastavení thresholdu.

Podiel skutočne pozitívnych prípadov v množine $S(\tau)$ definujeme ako

$$r(\tau) := \frac{\sum_{i \in S(\tau)} y_i^{true}}{|S(\tau)|}.$$

Celkový podiel pozitívnych prípadov v celom dátovom súbore je daný vzťahom

$$r_{\text{total}} := \frac{\sum_{i=1}^{N} y_i^{true}}{N},$$

kde N označuje celkový počet pozorovaní (veľkosť dátového súboru).

Definícia 2.4 (Lift)**.** *Lift pre prahovú hodnotu τ je definovaný ako pomer podielu pozitívnych prípadov v množine $S(\tau)$ k celkovému podielu pozitívnych prípadov v dátovom súbore:*

$$\mathit{Lift}(\tau) := \frac{r(\tau)}{r_{total}}.$$

Hodnota liftu tak vyjadruje, koľkokrát je pravdepodobnosť výskytu pozitívneho prípadu vo vybranej množine vyššia v porovnaní s náhodným výberom z celého dátového súboru. Hodnota Lift = 1 zodpovedá náhodnému modelu, zatiaľ čo hodnoty väčšie ako 1 indikujú lepšiu schopnosť modelu identifikovať pozitívne prípady.

Lift pre top 10 % pozorovaní

V praktických aplikáciách sa často používa zjednodušený prístup založený na rozdelení zoradených dát do rovnakých skupín, najčastejšie decilov. Preto sa v našej práci v tomto prípade hodnotí lift pre prvý decil, teda pre top 10 % pozorovaní s najvyššou predikovanou pravdepodobnosťou.

Po zoradení dát podľa y_i^{pred} zostupne vyberieme prvých 10 % pozorovaní, ktoré označíme ako množinu $S_{10\%}$. Podiel pozitívnych prípadov v tejto množine je

$$r_{10\%} = \frac{\sum_{i \in S_{10\%}} y_i^{true}}{|S_{10\%}|}.$$

Lift pre top 10 % je potom definovaný ako

$$\text{Lift}_{10\%} = \frac{r_{10\%}}{r_{\text{total}}}.$$

Príklad:

Uvažujme dátový súbor s 1000 pozorovaniami, z ktorých 100 je pozitívnych, čo zodpovedá celkovému podielu $r_{\text{total}} = 0.10$. Ak sa v top 10 % pozorovaní (100 pozorovaní s najvyššou predikovanou pravdepodobnosťou) nachádza 40 pozitívnych prípadov, potom

$$r_{10\%} = \frac{40}{100} = 0.40, \quad \text{Lift}_{10\%} = \frac{0.40}{0.10} = 4.0.$$

To znamená, že výber top 10 % pozorovaní na základe modelu obsahuje štyrikrát vyšší podiel pozitívnych prípadov než náhodný výber z celej populácie.

2.2 Meranie príspevkov premenných

Metóda SHapley Additive exPlanations (SHAP) predstavuje nástroj na vysvetľovanie individuálnych predikcií modelov strojového učenia, ktorú v našej práci používame na základe [23].

V tejto práci využívame SHAP hodnoty ako nástroj na identifikáciu a kvantifikáciu dôležitosti vstupných premenných modelu. Naším cieľom je vysvetliť individuálne predikcie modelu rozdelením ich výslednej hodnoty na príspevky jednotlivých premenných tak, aby toto rozdelenie verne odrážalo vplyv jednotlivých premenných na výslednú predikciu.

Po natrénovaní klasifikačného modelu získame funkciu $f : \mathbb{R}^p \to \mathbb{R}$, ktorá každému vstupu $x = (x_1, \ldots, x_p)^\top$ priraďuje hodnotu $f(x)$. Pre pevne dané pozorovanie $x^* \in \mathbb{R}^p$ chceme určiť, akým spôsobom jednotlivé premenné $x_1, \ldots, x_p$ prispeli k hodnote predikcie $f(x^*)$.

Formálne hľadáme rozklad tvaru podľa [22]

$$f(x^*) := \phi_0 + \sum_{j=1}^{p} \phi_j(x^*), \tag{23}$$

kde ϕ_0 predstavuje referenčnú (základnú) hodnotu modelu, $\phi_j(x^*)$ vyjadruje príspevok j-tej premennej k individuálnej predikcii.

2.2.1 Základná myšlienka SHAP

SHAP definuje základnú hodnotu modelu ako

$$\phi_0 := \mathbb{E}[f(X)],$$

t. j. očakávanú hodnotu predikcie modelu vzhľadom na rozdelenie vstupných premenných. V praxi je táto hodnota aproximovaná pomocou dostupných dát, pričom v prípade, že model odhaduje podmienenú strednú hodnotu $\mathbb{E}[Y|X]$, závisí výsledná hodnota od štruktúry vstupov. Použitie upravených dátových distribúcií, napríklad v dôsledku undersamplingu[9], môže viesť k posunu tejto základnej hodnoty a tým ovplyvniť interpretáciu SHAP príspevkov.

Vzťah (23) teda vyjadruje, že každú individuálnu predikciu modelu možno interpretovať ako súčet referenčnej hodnoty modelu a príspevkov jednotlivých premenných, t. j. $f(x^*) := \mathbb{E}[f(x)] + \sum_{j=1}^{p} \phi_j(x^*)$.

Aby bolo možné formálne definovať príspevok jednotlivých premenných, pre túto metódu špecifikujeme, akú hodnotu model nadobúda, keď poznáme iba podmnožinu vstupných premenných.

Nech $S \subseteq \{1, \ldots, p\}$ je indexová množina premenných, definované v [21]. Označme x_S ako podvektor (projekciu) vektora x na komponenty s indexmi v S, t. j. vektor obsahujúci len premenné x_i pre $i \in S$. Potom definujeme charakteristickú funkciu ako

$$v(S) := \mathbb{E}\Big[f(x) \mid x_S = x_S^*\Big].$$

Hodnota $v(S)$ predstavuje očakávanú predikciu modelu, ak premenné zo skupiny S fixujeme na hodnoty z pozorovania x^*, zatiaľ čo ostatné premenné ostávajú náhodné podľa ich rozdelenia v dátach.

SHAP hodnota pre premennú $i \in \{1, \ldots, p\}$ je potom definovaná ako

$$\phi_i(x^*) := \sum_{S \subseteq \setminus \{i\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \Big(v(S \cup \{i\}) - v(S)\Big), \quad i = 1, \ldots, p. \tag{24}$$

kde:

- p je celkový počet vstupných premenných v modeli,

[9] Undersampling je technika znižovania počtu vzoriek dominantnej triedy, aby sa vyrovnalo nerovnovážne rozdelenie v trénovacích dátach.

- i je index premennej, ktorej príspevok k predikcii chceme zistiť,
- S je ľubovoľná podmnožina premenných *bez* premennej i, teda $S \subseteq \{1, \ldots, p\} \setminus \{i\}$,
- $|S|$ je počet premenných v podmnožine S,
- $v(S \cup \{i\}) - v(S)$ je marginálny príspevok premennej i k podmnožine S,
- $\frac{|S|!\,(p-|S|-1)!}{p!}$ je kombinatorický faktor, ktorý zabezpečuje, že každý možný poradový vstup premenných je rovnomerne zastúpený. Tento faktor predstavuje pravdepodobnosť, že podmnožina S predchádza premennej i vo všetkých možných poradiach premenných.

Teda SHAP hodnota $\phi_i(x^*)$ predstavuje očakávaný príspevok premennej i k predikcii modelu pre konkrétne pozorovanie x^*, priemerovaný cez všetky možné poradia, v ktorých môžu byť premenné pridané do modelu.

2.3 Vážené metriky predikcie

Pri vyhodnocovaní modelov sme okrem štandardného prístupu, kde sa metriky počítajú na úrovni jednotlivých pozorovaní/riadkov dátoveho súboru, uvažovali aj alternatívny prístup založený na vážení pozorovaní podľa klienta. Tento prístup je motivovaný tým, že v prvom (webovom) a treťom (kombinovanom) modeli sa jeden klient môže v dátovom súbore vyskytovať viackrát, a to v prípade, že má priradených viac cookies.

Ak by sme metriky počítali priamo na úrovni riadkov bez úpravy, klienti s väčším počtom cookies by mali väčší vplyv na výsledné hodnoty metrík než klienti s menším počtom cookies. Aby sme tento efekt eliminovali a zabezpečili rovnakú váhu každého klienta, zaviedli sme váhy jednotlivých riadkov.

Nech klient i má v dátovom súbore n_i riadkov. Každému riadku patriacemu tomuto klientovi priradíme váhu

$$w_i := \frac{1}{n_i}.$$

Súčet váh všetkých riadkov patriacich jednému klientovi je tak rovný jednej, čím zabezpečíme, že každý klient má pri výpočte metrík rovnaký celkový vplyv bez ohľadu na počet cookies.

Príklad výpočtu váh pre jednoduchý dátový súbor je uvedený v tabuľke 1.

Pri výpočte vážených metrík vychádzame z definícií uvedených v kapitole 2. Rozdiel

Tabuľka 1: Príklad výpočtu váh pri viacerých cookies pre jedného klienta

customerid	cookie	w_i
1001	abc122	1/2
1001	mno566	1/2
1005	pqr677	1
1006	stu788	1

oproti štandardnému prístupu spočíva v tom, že jednotlivé pozorovania sú vážené pomocou váh w_i, ktoré zohľadňujú počet riadkov patriacich danému klientovi.

Váženú accuracy potom definujeme ako

$$\text{Vážená accuracy} := \frac{\sum_i w_i \cdot correct_i}{\sum_i w_i}$$

kde $correct_i$ nadobúda hodnotu 1, ak je predikcia pre pozorovanie i správna, a 0 v opačnom prípade.

Podobne ako v prípade nevážených metrík, aj tu vychádzame z konfúznej matice, ktorá je tvorená hodnotami TP, FP, TN a FN. V tomto prípade sa však jednotlivé kategórie počítajú ako vážené sumy pozorovaní:

- $TP := \sum_i w_i \cdot \mathbf{1}(y_i = 1 \wedge \hat{y}_i = 1)$,
- $TN := \sum_i w_i \cdot \mathbf{1}(y_i = 0 \wedge \hat{y}_i = 0)$,
- $FP := \sum_i w_i \cdot \mathbf{1}(y_i = 0 \wedge \hat{y}_i = 1)$,
- $FN := \sum_i w_i \cdot \mathbf{1}(y_i = 1 \wedge \hat{y}_i = 0)$,

kde $y_i \in \{0, 1\}$ označuje skutočnú triedu pozorovania i, $\hat{y}_i \in \{0, 1\}$ jeho predikovanú triedu podľa modelu. Napríklad TP predstavuje vážený súčet všetkých vážených pozorovaní, pri ktorých model správne predikoval pozitívnu triedu.

Na základe týchto vážených hodnôt sa následne počítajú aj ostatné metriky. Pri konštrukcii ROC krivky sa pre rôzne hodnoty prahovej hodnoty τ počítajú metriky

$$TPR := \frac{TP}{TP + FN}, \qquad FPR := \frac{FP}{FP + TN}.$$

Analogicky sa pri Precision–Recall krivke používajú vzťahy

$$Recall := \frac{TP}{TP + FN}, \qquad Precision := \frac{TP}{TP + FP}.$$

Keďže hodnoty TP, FP, TN a FN sú v tomto prípade definované ako vážené sumy pozorovaní, všetky odvodené metriky prirodzene zohľadňujú váhy jednotlivých riadkov dátového súboru.

Pre lift opäť definujeme množinu vybraných pozorovaní $S(\tau)$ podobne ako v **??**. Podiel skutočne pozitívnych prípadov vo vybranej množine počítame váženým spôsobom:

$$r_w(\tau) := \frac{\sum_{i \in S(\tau)} w_i \cdot y_i}{\sum_{i \in S(\tau)} w_i},$$

kde w_i je váha riadku i podľa počtu riadkov klienta a $y_i \in \{0, 1\}$ je skutočná trieda.

Celkový vážený podiel pozitívnych prípadov v dátovom súbore je

$$r_w^{total} := \frac{\sum_{i=1}^{N} w_i \cdot y_i}{\sum_{i=1}^{N} w_i}.$$

Vážený lift pre threshold τ potom definujeme ako

$$Lift_w(\tau) := \frac{r_w(\tau)}{r_w^{total}}.$$

3 Formulácia problému optimalizácie zobrazovania reklamy

V tejto kapitole sa zameriame na bližšie špecifikovanie riešeného problému a jeho formalizáciu. Cieľom je definovať, aký typ predikčnej úlohy riešime a aké dátové zdroje pri tom využívame.

3.1 Cookies

V súčasnosti je internetové prostredie charakteristické obrovským objemom výmeny dát medzi používateľmi a poskytovateľmi online služieb. Jedným zo základných nástrojov, prostredníctvom ktorého sa tieto dáta zbierajú a uchovávajú, sú tzv. cookies.

Cookies sú aj základnym pilierom našej diplomovej práce o optimalizácii webstránky, a preto si treba stručne vysvetliť, čo cookie je.

Cookies sú malé dátové súbory uložené v prehliadači, ktoré umožňujú webovým stránkam rozpoznať používateľa pri opakovaných návštevách. Podľa [5] slúžia na pamätanie si údajov, nastavení a prihlásenia používateľa, sledovanie jeho správania na stránkach a personalizáciu obsahu podľa jeho preferencií. Obsahujú páry "názov = hodnota" a často slúžia na uchovávanie jedinečného identifikátora, tzv. ID cookie, ktorý umožňuje sledovať konkrétneho používateľa.

Ako sa spomína v článku [4], ktorého autori uskutočnili analýzu jedného milióna najpopulárnejších stránok v roku 2016, pri prehliadaní webu sú používatelia sledovaní dvoma typmi subjektov. "First parties" (weby, ktoré používateľ navštevuje priamo) a "third parties" (skryté sledovače, napríklad reklamné siete vložené do väčšiny stránok). "Third parties" môžu získať prehľad o histórii prehliadania používateľov pomocou kombinácie cookies a cez referer header (údaj, ktorý prehliadač posiela serveru a hovorí mu, z akej stránky používateľ prišiel alebo vie obsahovať aj citlivejšie informácie ako email).

Tieto súbory zohrávajú kľúčovú úlohu pri získavaní personalizovaných informácií. Personalizované informácie predstavujú údaje, ktoré sa dajú priradiť ku konkrétnemu používateľovi. Priraďujú sa podľa tzv. správania klienta, čiže predchádzajúcich činností klienta na internete a webových stránkach. Jedná sa o informácie ako predchádzajúce návštevy stránok, preferencie, jazyk alebo identifikátory používané pri reklamných kampaniach. Na

základe týchto údajov je možné prispôsobiť obsah webu, zobrazovať cielenú reklamu alebo optimalizovať služby pre daného jedného klienta.

Podobne tieto informácie umožňujú vytvárať profil používateľa, ktorý sa môže zdieľať medzi viacerými webmi a reklamnými sieťami, často prostredníctvom tzv. "cookie syncing".

Záver analýzy [4] o "third parties" hovoril o tom, že väčšina "third parties" je špecifická a používa sa len na pár weboch, ale existuje malá skupina (123 subjektov), ktorá sa objavuje na veľkom množstve webov a je skutočne rozšírená. Analýza podľa kategórií ukázala, že spravodajské stránky majú najviac sledovačov, zatiaľ čo stránky vládne, univerzitné či neziskové majú sledovačov málo. Podobne článok posudzoval "third parties" ako riziko pre súkromie používateľov a dá sa proti týmto zásahom do súkromia účinne brániť pomocou vstavaných funkcií prehliadača na ochranu súkromia alebo rozšírení ako Ghostery (špecializované rozšírenie blokujúce sledovače), ktoré tento počet zníži ešte výraznejšie.

3.1.1 Tracking

Tracking na internete prebieha podľa [6] tak, že webová stránka alebo "third party" uloží do prehliadača používateľa cookie, ktorá obsahuje jedinečný identifikátor. Túto cookie zákazník najprv musí odsúhlasiť. Ďalej tento identifikátor umožňuje rozpoznať používateľa pri ďalšej návšteve. Ak cookie vytvorí samotná stránka, sledovanie je obmedzené iba na daný web (angl. within-site tracking). Ak však cookie nastaví tretia strana, napríklad reklamný systém zobrazený na viacerých stránkach, ten istý identifikátor sa prenáša naprieč rôznymi webmi (angl. cross-site tracking). Takto sa dá vytvoriť komplexný profil používateľa, zahŕňajúci jeho zvyky, záujmy a históriu prehliadania.

Uložené cookie spracováva „third-party", resp. právna entita alebo tzv. data brokers, teda externá spoločnosť, ktorá nie je prevádzkovateľom danej webovej stránky. Prostredníctvom vloženého skriptu alebo trackovacieho pixelu získava od používateľa rôzne údaje týkajúce sa jeho správania na internete. Tieto právne entity alebo dátoví sprostredkovatelia typicky poskytujú analytické, marketingové alebo reklamné služby, pričom zhromaždené informácie využívajú na profilovanie používateľa, meranie konverzií či cielenú reklamu. Medzi najčastejšie „third-party" subjekty patria napríklad Google (Google Analytics, Google Ads), Meta (Facebook Pixel, Instagram tracking), TikTok (TikTok Pixel),

ako aj klasickí data brokerage poskytovatelia [5] ako Acxiom, Datalogix či Epsilon, ktorí sa špecializujú na rozsiahle zhromažďovanie a obchodovanie s údajmi o používateľoch. Tie ďalej posielajú personalizované informácie inej stránke, ktorú zákazník navštívil a tá dokáže prispôsobiť obsah (reklamy) danému zákazníkovi.

Trackeri pritom používajú rôzne mechanizmy od klasických cookies, cez úložisko v prehliadači (angl. LocalStorage), až po vložené prvky ako reklamy, popup okná alebo sociálne widgety (napr. tlačidlo „Like“ od Facebooku). Vďaka tomu môžu obchádzať jednoduché obmedzenia a rozširovať rozsah svojho sledovania.

Tento proces je znázornený na obr. 4. Obrázok ilustruje, ako sa pri návšteve webovej stránky ukladajú cookies o používateľovi a ako sa tieto údaje následne využívajú na zobrazovanie cielenej reklamy na iných stránkach, napríklad reklamy súvisiacej s pôvodne navštívenou stránkou.



Obr. 4: Priebeh trackingu cookies

V kontexte tejto práce, ktorá vzniká v spolupráci s ČSOB, vystupuje ČSOB v úlohe „first-party“ subjektu, ktorý sám spracúva vybrané údaje na analytické a marketingové účely.

3.1.2 Cookies a tracking z pohľadu banky

Na webovej stránke ČSOB je implementovaný nástroj Bloomreach, ktorý slúži na analytické a personalizačné účely. Bloomreach umožňuje sledovať správanie používateľov v rámci jednej domény a na základe získaných údajov optimalizovať obsah webovej stránky a zlepšovať používateľskú skúsenosť.

Bloomreach podľa [15] ukladá jedinečný identifikátor používateľa, ktorý umožňuje jeho rozpoznanie pri opakovaných návštevách stránky a analyzovanie interakcií s obsahom. Získané údaje sa využívajú najmä na personalizáciu obsahu, vyhodnocovanie správania návštevníkov a optimalizáciu webu. Sleduje každého používateľa, ktorý navštívi webovú stránku, pričom používateľom môže byť reálna osoba (potenciálny zákazník), ale aj bot (automatický počítačový program).

Celý proces sledovania začína vstupom používateľa na webovú stránku a udelením súhlasu so spracovaním cookies prostredníctvom cookie lišty ako je znázornené na obr. 4. Následne dochádza k zaznamenávaniu udalostí (angl. eventov), ktoré sú dostupné automaticky. Medzi základné udalosti patrí najmä začiatok návštevy webovej stránky (angl. session_start), ktorý sa zaznamenáva pri vstupe používateľa na stránku, a koniec návštevy (angl. session_end), ktorý reprezentuje ukončenie relácie (návšteva stránky) po opustení stránky alebo po uplynutí časového limitu neaktivity.

Tieto udalosti umožňujú analyzovať trvanie relácie a správanie používateľa počas návštevy stránky. Spolu s nimi sa zaznamenávajú aj sprievodné atribúty, ako napríklad krajina a mesto návštevníka, zdroj návštevy (referrer), údaje o kampani ako napríklad UTM parametre (textové parametre pridávané na koniec URL webovej adresy), typ zariadenia, operačný systém či prehliadač. Na základe týchto údajov je možné vyhodnocovať pôvod návštevnosti a technické prostredie používateľov.

Kľúčovou udalosťou z hľadiska analýzy správania je udalosť page_visit, ktorá sa zaznamenáva pri každom zobrazení podstránky. Táto udalosť umožňuje sledovať pohyb používateľa naprieč webovou stránkou a analyzovať, ktoré časti obsahu sú najčastejšie navštevované.

Počas prvej návštevy webovej stránky Bloomreach automaticky vytvorí profil používateľa, ktorý je identifikovaný prostredníctvom unikátneho identifikátora uloženého v súboroch cookie pre dané zariadenie alebo prehliadač. Používateľ tak môže mať viacero

identifikátorov v prípade, že webovú stránku navštevuje z viacerých zariadení.

Ak používateľ vykoná identifikačnú akciu, napríklad prihlásenie/registráciu alebo nákup, jeho profil je následne prepojený s jednoznačným identifikátorom používateľa, ako je e-mailová adresa alebo interné zákaznícke číslo. Tento krok umožňuje spojiť všetky predchádzajúce interakcie a udalosti z anonymného profilu s identifikovaným zákazníkom, čím vzniká jednotný profil. Bloomreach pritom zabezpečuje, že sa nevytvárajú duplicitné profily a historické údaje používateľa sú zachované a správne zlúčené. V kontexte našej práce to znamená to, že ku konkrétnej cookie na stránke vieme priradiť customerid, čo vysvetlíme ďalej v práci.

Okrem automatického zlúčenia profilov je možné zákazníka identifikovať aj pri prekliku z e-mailových alebo iných marketingových kampaní na základe UTM parametrov v URL adrese.

Podobne na stránke ČSOB majú cookie aj tzv. expiráciu. To znamená, že ak zákazník stránku nenavštívi viac ako 2 roky, tak pri ďalšej návšteve sa vytvorí nová cookie, ktorá bude považovaná za nového používateľa. Aj týmto spôsobom môže mať jeden používateľ viacero cookies, aj bez použitia viacerých zariadení.

3.2 Dáta používané v našej práci

V tejto podkapitole predstavíme dátové zdroje použité v našej práci. Dáta pozostávajú z viacerých tabuliek, ktoré zachytávajú rôzne aspekty, od správania klienta a jeho aktivity na webovej stránke až po informácie o transakciách, účtoch a používaní elektronického bankovníctva.

Prehľad jednotlivých tabuliek spolu s počtom premenných je uvedený v Tabuľke 2.

Jednotlivé dátové zdroje sa líšia spôsobom identifikácie používateľa. Tabuľka *attr_web_tracking* je jediná, ktorá je naviazaná na identifikátor *cookie*, a teda zachytáva správanie používateľa počas návštevy webovej stránky. Ostatné tabuľky (*attr_txn*, *attr_elb*, *attr_bank_account*, *attr_sb*) sú viazané na jednoznačný identifikátor klienta *customerid*, ktorý reprezentuje konkrétneho klienta banky. Toto prepojenie umožňuje kombinovať webové dáta s historickými klientskými dátami.

Tabuľka *attr_web_tracking* obsahuje premenné popisujúce správanie používateľa na webovej stránke. Keďže ide o tabuľku, ktorú sme v tejto práci vytvárali, tak uvádzame aj podrobný postup tvorby uvedený v podkapitole 3.2.3

Tabuľka 2: Prehľad tabuliek a počtu premenných

Tabuľka	**Počet premenných**	**Popis**
ATTR_WEB_TRACKING	85	Charakteristika webu
ATTR_TXN	287	Transakcie
ATTR_ELB	254	Elektronické bankovníctvo
ATTR_BANK_ACCOUNT	192	Účet
ATTR_SB	204	Smart Banking (mobilná aplikácia elektronického bankovníctva ČSOB)
Spolu	**1022**	

Tabuľka *attr_txn* obsahuje agregované charakteristiky finančných transakcií klienta. Premenné sú konštruované pre rôzne časové horizonty (mesačný, kvartálny, ročný) a zahŕňajú napríklad počty transakcií, objemy transakcií, mediánové hodnoty či transakcie podľa typu (kreditné, debetné, kartové, ATM, pobočka) alebo meny. Okrem toho obsahuje aj rôzne odvodené premenné, ako sú pomery alebo agregované sumy.

Tabuľka *attr_elb* zachytáva aktivitu klienta v elektronickom bankovníctve. Obsahuje premenné popisujúce počet a typ operácií, frekvenciu prihlasovania, časové rozdelenie aktivity (napr. ráno, večer), ako aj rôzne agregácie a transformácie týchto údajov.

Tabuľka *attr_bank_account* obsahuje charakteristiky bankového účtu klienta. Ide najmä o statické alebo pomalšie sa meniace premenné, ako napríklad typ účtu, jeho vlastnosti alebo základné charakteristiky klienta viazané na účet.

3.2.1 Charakterizácia cookie dát z bankovej webstránky

V rámci diplomovej práce sme pracovali s internými dátami banky, konkrétne s tabuľkou, ktorá obsahovala informácie o interakciách klientov banky s webom. Jednalo sa o tabuľku *page_visit*. Každý riadok tejto tabuľky predstavoval jednu udalosť (operáciu) vykonanú uživateľom s konkrétnou cookie.

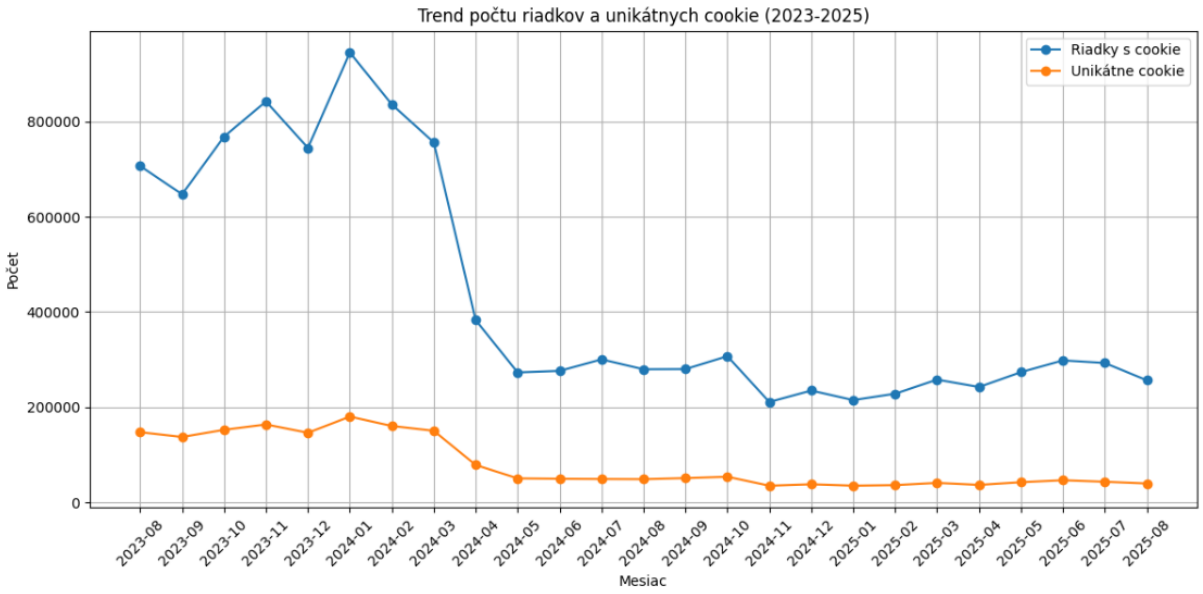
Dáta boli detailne logované a trackované pomocou cookies pre klientov, ktorí ich akceptovali, a obsahovali viacero premenných, ktoré reprezentovali rôzne aspekty správania používateľov. Okrem údajov o klientoch obsahovali napríklad údaje o tom, aké zariadenie s akým operačným systémom používal klient alebo aj URL adresu, ktorú navštívil. Po-

dobne tam boli premenné, ktoré hovorili o tom, akú podstránku mal klient otvorenú alebo aký webový prehliadač používal. Podrobný zoznam premenných sa nachádza v Prílohe B.

3.2.2 Deskriptívna štatistika cookie dát z bankovej webstránky

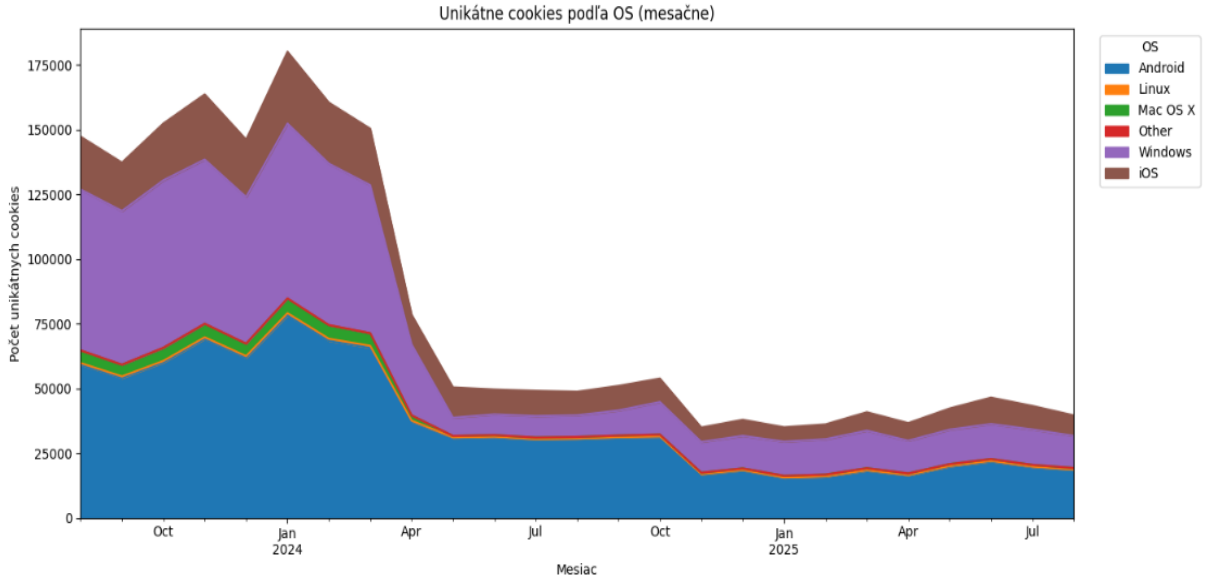
V tejto kapitole uvedieme aj deskriptívne štatistiky dát *page_visit*, aby sme si priblížili pohľad na tento dátový súbor.

Na obr. 5 sledujeme údaje za dvojročné obdobie od augusta 2023 do augusta 2025. Mesačné počty riadkov, ktoré obsahujú v dátovom súbore *page_visit* vyplnenú cookie sú spočiatku okolo 700 000 a následne v období marca a apríla 2024 prichádza pokles, kde sa počty pohybujú okolo 300 000, kvôli poklesu eventov. Obrázok obsahuje aj počet unikátnych cookie, ktorých bolo pred poklesom okolo 180 000 a následne okolo 80 000.



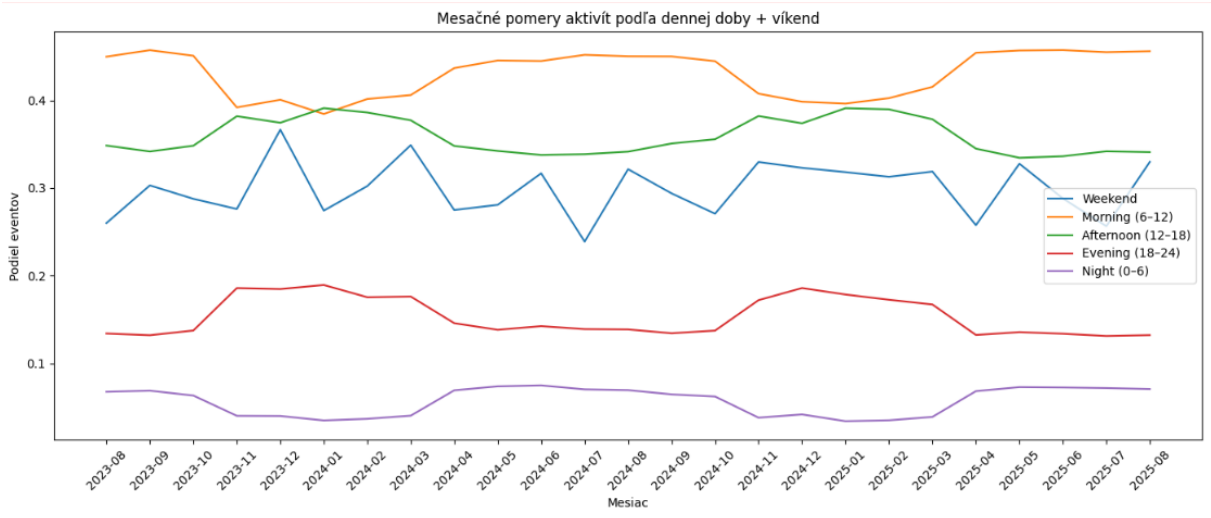
Obr. 5: Mesačné počty riadkov a unikátnych cookies v dátovom súbore *page_visit*

Ďalej zobrazujeme obrázok 6, ktorý je založený na unikátnych cookies, pričom každej cookie je v danom mesiaci priradený prevládajúci operačný systém. Výsledky sme zobrazili pomocou *stacked area chart*, ktorý umožňuje sledovať celkový objem cookie v čase a zároveň ich rozdelenie podľa operačných systémov. Z obrázka je zrejmý výrazný pokles celkového objemu cookies naprieč všetkými operačnými systémami. Zároveň nízke zastúpenie niektorých operačných systémov pravdepodobne súvisí s tým, na ktorých webových portáloch má banka nastavený tracking cookies.



Obr. 6: Mesačné počty hodnoty a počty premennej *os* v *page_visit*

Podobne sledujeme obr. 7, kde vidíme, že okolo 30% všetkých udalostí nastáva cez víkend. Najviac udalostí nastáva v ranných hodinách od 6 do 12 a najmenej podľa očakávaní v nočných.



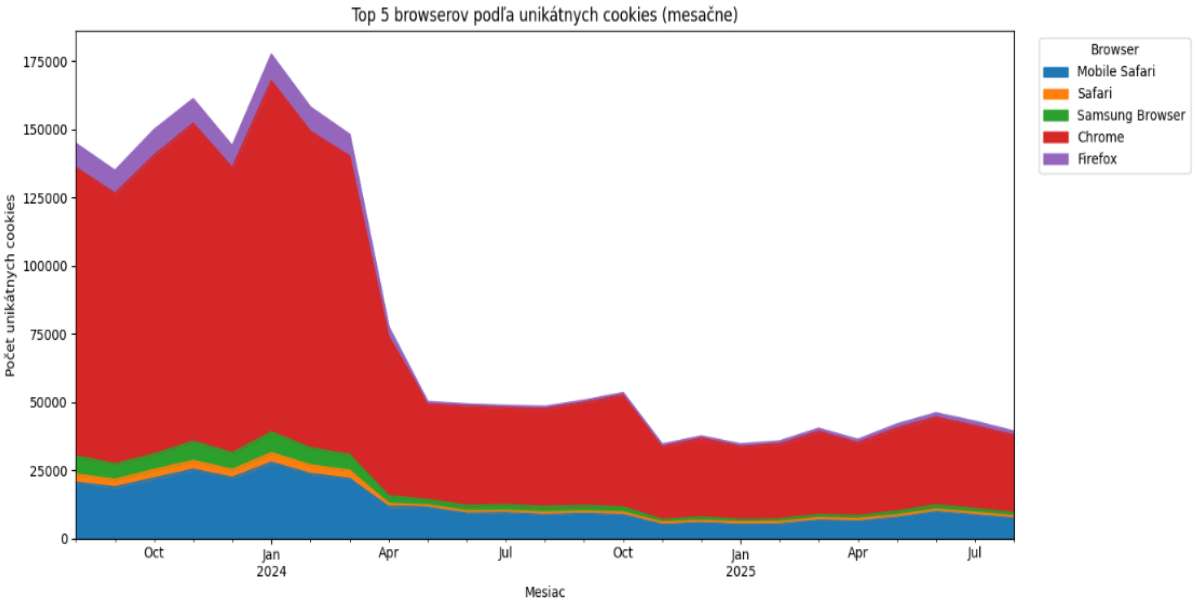
Obr. 7: Mesačné pomery víkendových udalostí a udalostí podľa časových intervalov *page_visit*

Tabuľka 3: TOP päť externých referrerov za celé sledované obdobie. Referrery obsahujúce analyzovanú banku ČSOB neboli zahrnuté, aby sme sa sústredili na externé zdroje návštevnosti.

Rank	Referrer
1	https://www.google.com/
2	https://www.google.sk/
3	https://www.bing.com/
4	android-app://com.google.android.gm/
5	https://3dsecure.gpwebpay.com/

Tabuľka 3 ukazuje top 5 externých referrerov za celé sledované obdobie. Referrer označuje URL, z ktorej používateľ prišiel na sledovanú stránku. Napríklad `https://www.google.com/` znamená, že návštevník prišiel z Google vyhľadávača. Referrer označený číslom päť vzniká, keď je klient počas platby kartou presmerovaný na overenie a následne sa vráti späť na stránku.

Tento prehľad umožňuje analyzovať, z ktorých externých zdrojov web najviac získava návštevnosť a ktoré kanály sú najefektívnejšie. Referrery obsahujúce analyzovanú banku ČSOB boli zámerne vynechané.



Obr. 8: Mesačné počty hodnoty a počty premennej *browser* v *page_visit*

Obr. 8 ukazuje premennú browser, čo predstavuje typ webového prehliadača, ktorý používateľ používa pri návšteve stránky, napríklad Chrome, Firefox, Safari či Edge.

Proces agregácie bol podobný ako pri analýze operačných systémov. Pre každý mesiac sme určili dominantný browser pre každú cookie a následne spočítali počet unikátnych cookies podľa browseru.

Z celkového pohľadu je jednoznačne najčastejším browserom Chrome, čo znamená, že väčšina používateľov pristupuje na stránku práve týmto prehliadačom. Ďalším významným je Mobile Safari a ostatné browsere sú už zastúpené len zriedka.

3.2.3 Transformácia cookie dát do atribútovej tabuľky

Pôvodné dáta *page_visit* (cookie dáta) sme následne transformovali do tzv. `ATTR` tabuľky (tabuľky agregovaných atribútov) na mesačnej úrovni s názvom *attr_web_tracking*. Súčasťou práce bola aj tvorba tejto tabuľky z webovej aktivity. `ATTR` tabuľka je mesačne vytváraná databáza, ktorá ku každému zákazníkovi priraďuje súhrnné ukazovatele správania na webovej stránke banky, odvodené zo sledovaných interakcií pomocou cookie. Táto tabuľka slúži ako vstup pre modelovanie a ďalšiu analytickú prácu. Každý riadok predstavuje jednu známu cookie a obsahuje vypočítané numerické alebo kategorické charakteristiky správania tejto cookie, ktoré sme agregovali mesačne, štvrťročne a ročne. Sú to premenné ako počty navštívených stránok, počty preklikov na web z rôznych zdrojov, počty navštívených stránok súvisiacich s poisteniami a bankovým účtom alebo priemer trvania návštevy. Podrobne sú tieto premenné popísane v Prílohe B.

V tabuľke *attr_web_tracking* sa nachádza 31 premenných počítaných každý mesiac na základe `month_id` a ich štvrťročné (súčet troch mesačných) a ročné (súčet štyroch štvrťročných.) agregácie.

Dátový súbor obsahuje tri typy premenných, ktoré boli vytvorené podobne ako v [7], prostredníctvom využitia RFM modelu (Recency - čas od poslednej činnosti klienta, Frequency - počet, Monetary value - peňažná hodnota). Článok sa zaoberá optimalizáciou direct mail[10] kampaní, kde spomínané tri typi premenných pomáhajú klasifikovať zákazníkov na základe ich minulého správania a priraďuje im pravdepodobnosti odozvy na budúce

[10]Direct mail kampaň je marketingová aktivita, pri ktorej firma oslovuje zákazníkov priamo cez fyzickú poštu alebo personalizované správy.

marketingové podnety. Ďalej poukazuje na to, že ziskovosť kampane nezávisí od oslovenia čo najväčšej populácie, ale od správneho výberu segmentu s najvyššou očakávanou návratnosťou, pričom sa zohľadňujú hraničné náklady a výnosy.

Na dátové transformácie sme využívali dve python knižnice PySpark [8] a Polars [9]. Porovnali sme časovú náročnosť a spotrebu pamäte na serveroch klastra v Prílohe A.

Dostali sme sa teda z dát *page_visit*, ktoré sme transformovali do *attr_web_tracking*, ktorá obsahovala atribúty narátané na cookie. Následne pre kvantifikáciu efektu webových premenných sme ju upravili a zobrali iba tie `cookie`, ku ktorým sme poznali `customerid`.

Tabuľka 4: Ilustratívny príklad dátoveho súboru z *attr_web_tracking*.

month_id	**cookie**	**customerid**	**Premenná_1**	**Premenná_2**
202408	abc122	1001	12	0.45
202408	def233	1002	7	1.23
202408	ghi344	None	15	0.89
202408	jkl455	1003	5	0.72
202408	mno566	1001	20	0.12

Nie všetky riadky tabuľky obsahovali vyplnené `customerid`. Pre ďalšiu analýzu uvažujeme iba tie riadky, pri ktorých sú k dispozícii obidva identifikátory (`cookie` aj `customerid`), keďže len v tomto prípade je možné prepojiť webové dáta s klientskými dátami banky.

- `cookie` predstavuje dáta zo stránky (záznam správania z webu),
- `customerid` predstavuje dáta o zákazníkovi (klientské informácie),

Týmto spôsobom sme zabezpečili, že webové dáta boli správne spojené s klientskými dátami iba v prípadoch, kde je známa identita zákazníka. Na vymyslenom príklade vidíme, podobne ako sme spomínali v 3.1.2, že nejaký konkrétny jeden customer môže mať viac `cookie` (`customerid` = 1001). Výsledný dátový súbor (obsahujúci z ilustratívneho príkladu iba riadky, kde `cookie` = abc122, def233, jkl455 a mno566) bol potom čistý a pripravený na ďalšie spracovanie a analýzu.

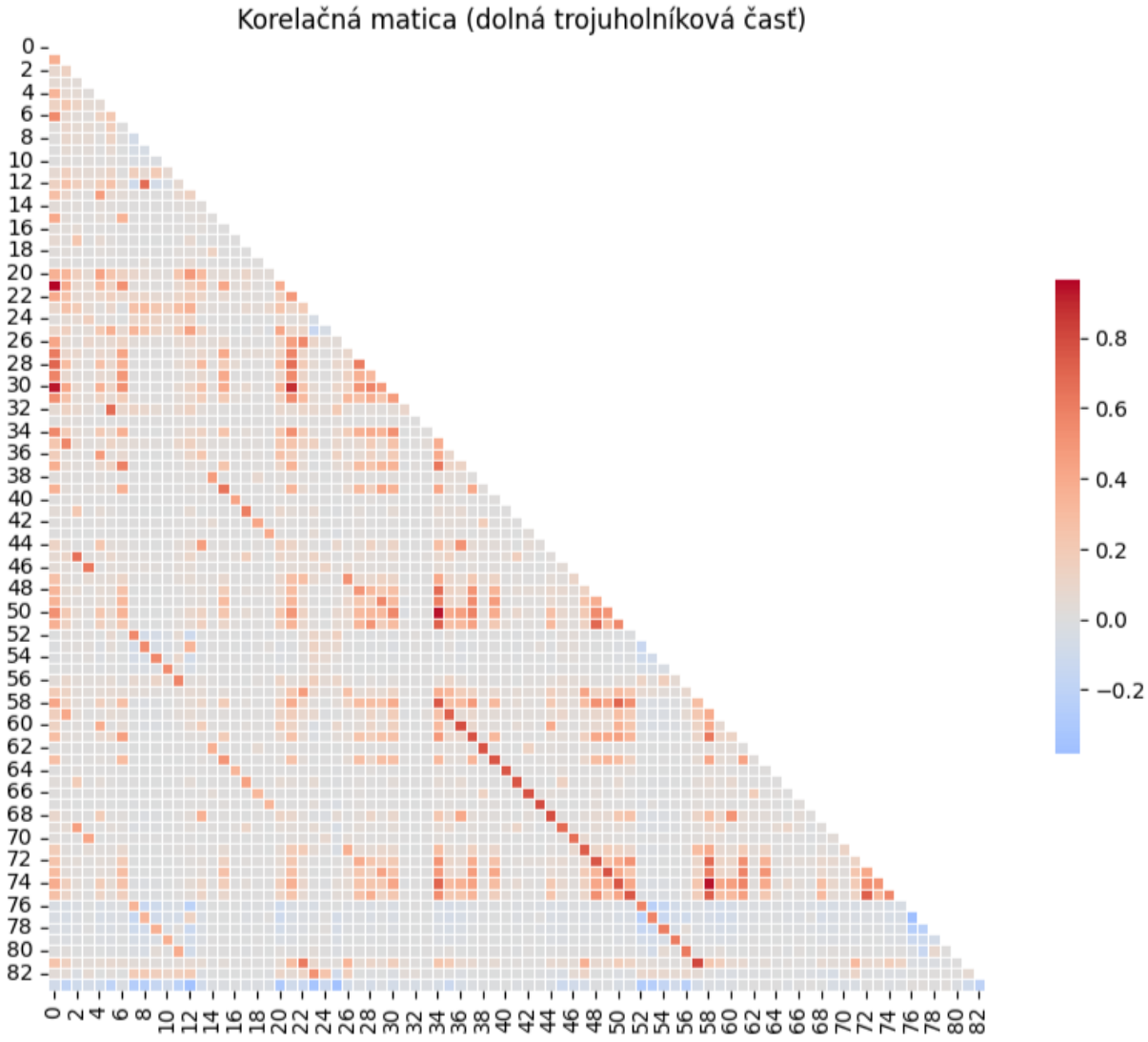
3.2.4 Korelačná matica závislých premenných dát attr_web_tracking

V tejto kapitole ukážeme, že vytvorené premenné sú vo všeobecnosti málo korelované, a preto predstavujú vhodné vstupy do modelu. Na tento účel sme za obdobie od augusta 2023 do júla 2025 extrahovali identifikátory `customerid`, `cookie` a numerické premenné následne skonsolidovali do jednej tabuľky, z ktorej sme vypočítali korelačnú maticu, ktorá umožňuje analyzovať vzájomné vzťahy medzi sledovanými atribútmi webovej aktivity používateľov.

Túto korelačnú tabuľku môžeme vidieť ako obrázok 9, kde je korelácia vizualizovaná na farebnej škále, podľa toho, aká je korelácia medzi dvoma premennými. Čím je farba výraznejšia (tmavočervená alebo tmavomodrá), tým väčšia korelácia.

V korelačnej matici pozorujeme vysoké hodnoty korelačného koeficientu ($\rho > 0{,}9$)) najmä medzi premennými označenými príponami `_q` a `_y` prislúchajúce diagonálnym štruktúram korelačnej matice. Ide o rovnaké typy premenných merané v rôznych časových agregáciách, pričom pri každej premennej s označením `_q` sa vyskytuje silná korelácia s jej zodpovedajúcou premennou `_y`. Napríklad premenná `c_visit_q` vykazuje vysokú koreláciu s premennou `c_visit_y`. Tento vzor sa opakuje aj pri ďalších pároch premenných rovnakého názvu, čo naznačuje silnú lineárnu závislosť medzi štvrťročnými a ročnými charakteristikami rovnakého správania.

Silne korelované premenné predstavujú prevažne rovnakú informáciu, preto bola v ďalších častiach práce (3.5, pri konštrukcii dátového súboru pre modely strojového učenia) aplikovaná redukcia týchto premenných a môžeme očakávať, že z oboch premenných ostane iba jedna.



Obr. 9: Korelačná tabuľka *attr_web_tracking* za obdobie dvoch rokov

3.2.5 Kódovanie kategorických premenných

Ako sme uviedli v kapitole 1, mnohé klasifikačné metódy predpokladajú, že vstupné premenné sú reprezentované ako vektory v priestore $\mathbb{R}^p$. Z tohto dôvodu je potrebné kategorické premenné transformovať do numerickej podoby.

Problematika kódovania kategorických premenných je podrobne spracovaná v práci [10], konkrétne v kapitole 2.5, na ktorú sa v tomto kontexte odvolávame.

V našej práci sa kódovanie kategorických premenných týkalo výlučne premenných z tabuľky *attr_sb*, keďže iba táto tabuľka obsahovala kategorické premenné. Ostatné použité dátové tabuľky obsahovali iba numerické premenné, a preto nevyžadovali túto

transformáciu.

Po zakódovaní sú všetky vstupné premenné reprezentované v numerickej forme, čo umožňuje aplikáciu metód výberu premenných a následné trénovanie modelov.

3.3 Výber dát pre modely strojového učenia

Po základnom očistení finálnej *attr_web_tracking* tabuľky z 3.2.3, potrebujeme dáta pripraviť na modelovanie. Veľká časť prípravy dát je podobná ako v práci [10], konkrétne v štvrtej kapitole. Preto budeme v tejto časti kapitoly odkazovať hlavne na túto prácu a doplníme si prípadne naše zmeny.

3.3.1 Závislé premenné

Definícia závislej premennej (angl. sample value alebo target) vychádza z práce [10], konkrétne z podkapitoly 4.2. Závislá premenná indikuje výskyt modelovanej udalosti, ktorou je v našej aj citovanej práci založenie spotrebného úveru zákazníkom. Podobne modelujeme aj inú udalosť, a to novú jednorázovú investíciu so sumou vyššou ako 500 eur. Tieto dva odlišné finančné produkty sme zahrnuli s cieľom overiť robustnosť a generalizovateľnosť modelov naprieč rozdielnymi typmi zákazníckeho správania a rozhodovacích kontextov. Premenná má binárny charakter, pričom hodnota 1 označuje realizáciu udalosti a hodnota 0 jej nerealizáciu. Na tomto základe sa rozlišuje pozitívna a negatívna vzorka zákazníkov.

V našej práci bude závislá premenná využívaná podobne, a to v klasifikačných modeloch na predikciu pravdepodobnosti realizácie udalosti.

3.3.2 Konštrukcia dátovej vzorky pre modely strojového učenia

Postup výberu tréningových pozorovaní v našej práci vychádza opäť z metodiky uvedenej v [10], konkrétne z podkapitoly 4.3. Cieľom je zostaviť vzorku zákazníkov, ktorá reprezentuje ich správanie v rôznych časových obdobiach, a ku ktorej sú priradené už predpočítané premenné agregované ku konkrétnemu mesiacu.

Vybraná vzorka (z `customerid`, ktorí si zriadili spotrebný úver alebo uskutočnili jednorázovú investíciu so sumou vyššou ako 500 eur) je konštruovaná tak, aby každý zákazník

bol do dátovej vzorky zaradený najviac jeden krát.

Pozitívna vzorka pozostáva zo zákazníkov, u ktorých došlo k výskytu modelovanej udalosti. Každému takémuto zákazníkovi sú priradené premenné popisujúce jeho správanie v období pred jej realizáciou. Ak zákazník vykonal modelovanú udalosť viackrát, do vzorky je zaradený len raz, pričom sa náhodne vyberie jedna z udalostí.

Pri priraďovaní premenných sa zohľadňuje časový odstup medzi dátami a samotnou udalosťou, aby sa zabránilo úniku informácií. V závislosti od dátumu uskutočnenia udalosti sú zákazníkovi priradené premenné z jedného z dvoch celých mesiacov predchádzajúcich jej výskytu, pričom sa uplatňuje posun o 15 dní (pre lepšiu ilustráciu spôsobu priraďovania premenných v závislosti od dátumu realizácie udalosti, odkazujeme na schémy uvedené na Obr. 2 a Obr. 3 v práci [10]). Konkrétny mesiac, z ktorého sú premenné vybrané, je z týchto dvoch možností zvolený náhodne.

Negatívna vzorka zahŕňa zákazníkov, u ktorých modelovaná udalosť nenastala. Pre každého zákazníka je rovnako vybraný jeden náhodný mesiac, z ktorého sú priradené premenné, pričom zákazník sa môže v dátovej vzorke vyskytovať maximálne raz.

Tabuľka 5: Ilustratívny výzor vzorky pre modely strojového učenia

month_id	**customerid**	**sample_value**
202410	1001	1
202502	1005	0

Vo vzorke pre modely strojového učenia z tabuľky 5 boli `customerid` a ich údaje za 23 mesiacov, od augusta 2023 do júna 2025.

3.4 Konštrukcia dátových súborov pre modely strojového učenia

V tejto podkapitole si ukážeme to, ako sme využívali vzorku pre modely strojového učenia z časti 3.3.2 na dosiahnutie cieľov našej práce. Podobne si ukážeme aj to, ako sme využili našu *attr_web_tracking*, teda dáta narátané na `cookie` a ostatné *attr_tabuľky* z 3.2, teda dáta narátané na `customerid` (resp. záznamy z prehliadania webu a klientské dáta).

3.4.1 Cookie model

Prvé modelovanie sme urobili na samotnom dátovom súbore *attr_web_tracking*, ktorý využíva len informácie z trackovania cookies. To znamená, že na vzorku pre modely strojového učenia z tabuľky 5 sme napojili left joinom (spôsob pripojenia premenných ku vzorke pre modely strojového učenia) naše web dáta.

Ukážka 1: Pseudokód: Left join web dát na vzorku pre modely strojového učenia

```
attr_web_tracking = load_web_data()
vzorka_pre_modely = load_sample_data()
vysledne_data = left_join(
    vzorka_pre_modely,
    attr_web_tracking,
    on="customerid")
```

V tabuľke 6, vidíme príklad toho ako vyzeral dátový súbor pre prvý model, to znamená, že ak nejaký `customerid` mal viac `cookies` (ako je spomínané v 3.1.2), tak mal toľko riadkov, koľko mal `cookies`.

Tento dátový súbor obsahoval za dva roky okolo desať tisíc pozitívnych vzoriek pre spotrebný úver, okolo sedem tisíc pre jednorázovú investíciu a dokopy 89 premenných. Negatívne vzorky riešime ako hyperparameter pri trénovaní modelu, kde sa pozeráme na to, aký počet negatívnych premenných je optimálny ku pozitívnym. Tento štýl optimalizácie počtu negatívnych vzoriek budeme používať pre každý z našich modelov a je to popísané v Prílohe A, v časti o výbere hyperparametrov.

Tabuľka 6: Ilustratívny príklad cookie dátoveho súboru na modelovanie po left joine

month_id	customerid	sample_value	cookie	Premenná_1	Premenná_2
202410	1001	1	abc122	12	0.45
202410	1001	1	mno566	20	0.12
202502	1005	0	pqr677	2	0.93

Tým pádom prvý model, slúži na analýzu predikčnej schopnosti výlučne webových dát. Cieľom modelu je overiť, do akej miery samotné informácie o správaní `cookies` (s prísluchajúcim `customerid`) na webovej stránke umožňujú predikovať pravdepodobnosť nákupu produktu.

3.4.2 Klientský model

Druhý model využíva existujúce bankové dáta. Je založený výlučne na bankových klientských dátach získaných z atribútových tabuliek, ktoré obsahujú informácie napríklad o transakciách, bankových účtoch, využívaní elektronického bankovníctva alebo využívaní mobilnej aplikácie ČSOB banky (*attr_txn*, *attr_elb*, *attr_bank_account* a *attr_sb*). Cieľom tohto modelu je analyzovať predikčný potenciál výlučne bankových klientských dát.

Tabuľka 7 znázorňuje príklad dátového súboru pre druhý model, ktorý obsahoval iba `customerid` a príslušné dáta. Keďže nás v tomto prípade zaujímajú iba klientské dáta, berieme do modelu iba unikátne výskyty `customerid` (jeden klient sa vyskytuje iba jedenkrát), ale zároveň tých istých klientov ako v prvom webovom modeli.

Tento dátový súbor obsahoval za dva roky okolo sedem tisíc pozitívnych vzoriek pre spotrebný úver, okolo päť tisíc pre jednorázovú investíciu a dokopy 934 premenných.

Tabuľka 7: Ilustratívny príklad klientského dátového súboru na modelovanie po left joine

month_id	customerid	sample_value	Premenná_3	Premenná_4
202410	1001	1	241	14.2
202502	1005	0	156	1.7

Týmto spôsobom druhý model slúžil na analýzu predikčnej schopnosti výlučne bankových dát, respektíve interných klientských záznamov. Cieľom modelu bolo overiť, do akej miery samotné informácie o správaní klientov, vedených pod príslušným `customerid`, umožňujú predikovať pravdepodobnosť nákupu produktu.

3.4.3 Kombinovaný model

Tretí model je zložený z kombínacie všetkých tabuliek, ktoré sme doteraz spomínali.

V tabuľke 8, vidíme príklad toho ako vyzerá dátový súbor pre tretí model, to znamená, že obsahuje dáta z webu (narátané na `cookie`) a aj dáta z iných *attr_tabuliek* (narátané na `customerid`). Tým, že máme vo webových dátach premennú `customerid` a aj `cookie`, tak sme zaručili to, že klient z viac `cookies`, mal rozlišné riadky presne iba kvôli webovým premenným a preto sme opäť máme taký istý počet pozorovaní/riadkov ako prvý model.

To znamená, že ak je v treťom modeli jeden `customerid` a má viac `cookies`, budú dáta z bankových tabuliek pre všetky tieto riadky rovnaké

Tento dátový súbor obsahoval teda rovnako ako prvý za dva roky okolo desať tisíc pozitívnych vzoriek pre spotrebný úver, okolo sedem tisíc pre jednorázovú investíciu a dokopy 1015 premenných.

Tabuľka 8: Ilustratívny príklad po kombinácii webových a klientských dát po left joine

month_id	customerid	sample_value	cookie	Premenná_1	Premenná_2	Premenná_3	Premenná_4
202410	1001	1	abc122	12	0.45	241	14.2
202410	1001	1	mno566	20	0.12	241	14.2
202502	1005	0	pqr677	2	0.93	156	1.7

Týmto spôsobom tretí model slúžil na analýzu predikčnej schopnosti kombinovaných dát, ktoré obsahovali informácie o správaní používateľov na webovej stránke, klientské záznamy a aktivitu v Smart Banking aplikácii. Cieľom modelu bolo overiť, do akej miery spojené informácie zo všetkých týchto zdrojov umožňovali predikovať pravdepodobnosť nákupu produktu, a dokonca či kombinácia dát umožňovala presnejšie predikcie.

3.5 Výber finálnych premenných

Keďže sme mali tri rôzne dátové súbory s rôznymi počtami premenných, tak jedna z posledných transformácií je selekcia premenných, pri čom sa opäť odvolávame na prácu [10, str. 55]. Redukcia premenných sa oplatí, pretože odstraňuje zbytočné alebo podobné premenné, čím vedie k jednoduchšiemu modelu a stabilnejším predikciám na nových dátach.

Pri selekcii premenných sme využívali predprogramované funkcie v ČSOB. Postup pozostával z nasledujúcich krokov:

- odstránenie premenných na základe nízkej miery variability (po odčítaní horného a dolného kvantilu),
- transformácia kategorických premenných z tabuľky *attr_sb* do numerickej podoby (viď kapitola 3.2.5),
- selekcia premenných na základe korelácie so závislou premennou alebo vzájomnej informácie.

Výsledný počet premenných, ktoré vstupujú do modelu vidíme v tabuľke 9.

Pri výbere vysvetľujúcich premenných pre jednotlivé modely sme vychádzali z ich zoradenia podľa hodnoty vzájomnej informácie[11], ktorá kvantifikuje mieru závislosti medzi danou premennou a cieľovou premennou.

V prípade kombinovaného modelu sme zvolili fixný maximálny počet 600 premenných. Tento fixný počet premenných bol zvolený tak, aby bol model schopný zahrnúť aj webové (cookie) premenné, ktoré vo všeobecnosti dosahujú nižšie hodnoty vzájomnej informácie v porovnaní s klientskými premennými. Pri výrazne nižšom počte vybraných premenných by sa webové premenné do modelu takmer nedostali, čo by obmedzilo možnosť zachytiť ich prínos a ideu našej práce.

Následne sme určili hodnotu vzájomnej informácie prislúchajúcu premennej na 600. mieste v tomto zoradení a túto hodnotu sme použili ako prahovú úroveň pre výber premenných aj pre klientský a cookie model. Inými slovami, do jednotlivých modelov boli zahrnuté všetky premenné, ktorých hodnota vzájomnej informácie presiahla túto hranicu.

Maximálne počty premenných pre jednotlivé modely tak neboli stanovené priamo, ale boli dané touto spoločnou prahovou hodnotou vzájomnej informácie. Takýto postup zabezpečuje konzistentný výber premenných naprieč modelmi a zároveň umožňuje ich porovnanie z hľadiska informačného prínosu použitých prediktorov.

Tabuľka 9: Maximálny počet premenných v jednotlivých dátových súboroch a produktoch po selekcii

Model \ Závislá premenná	Spotrebný úver	Jednorázová investícia
Cookie	48	36
Klientský	492	480
Kombinovaný	600	600

Viac k tomu ako sme pristúpili ku počtu premenných pri trénovaní modelu je popísane v Prílohe A, v časti o výbere hyperparametrov.

[11] Po selekcii premenných boli výsledné premenné zoradené podľa vzájomnej informácie, kde prvá bola premenná s najväčšou hodnotou vzájomnej informácie, druhá bola druhá najväčšia atď.

4 Trénovanie klasifikačných modelov

V tejto časti popisujeme proces trénovania klasifikačných modelov na jednotlivých dátových súboroch. V súlade s formuláciou problému uvedenou v kapitole 3, boli modely trénované na troch rôznych dátových súboroch:

- cookie model,
- klientský model,
- kombinovaný model,

pričom každý z nich reprezentuje odlišný zdroj informácií o správaní klienta.

Pre každý z uvedených dátových súborov bolo natrénovaných päť klasifikačných modelov:

- logistická regresia (LR),
- metóda oporných bodov (SVM),
- náhodný les (RF),
- eXtreme Gradient Boosting (XGB),
- Light Gradient Boosting Machine (LGBM).

Celkovo tak bolo natrénovaných 30 modelov (15 modelov pre produkt spotrebný úver a 15 modelov pre jednorázovú investíciu), čo umožnilo porovnanie medzi jednotlivými algoritmami, medzi rôznymi typmi vstupných dát, a tak isto aj medzi rôznymi produktami.

4.1 Krosvalidácia a výber finálneho modelu

Pri trénovaní modelov sme využívali krosvalidáciu, ktorá je podrobnejšie popísaná v Prílohe A.

V našej práci používame krosvalidáciu, ktorá rozdelí trénovací dátový súbor na päť častí (angl. foldov). Je však dôležité zohľadniť časovú štruktúru dát, pretože jednotlivé záznamy sú usporiadané podľa mesiacov. Aby sme predišli úniku informácií medzi foldami pri trénovaní krosvalidácie, implementujeme *month-group stratified cross-validation*, kde všetky záznamy z jedného mesiaca (month_id) patria do jedného foldu. Navyše sa snažíme zachovať pomer cieľových tried (*stratifikácia*), aby každý fold reprezentoval celú distribúciu dát.

Pre každý fold krosvalidácie sú všetky klasifikačné modely trénované s hyperparamet-

rami optimalizovanými pomocou Optuna.

Po ukončení krosvalidácie sú vybrané najlepšie hyperparametre, na základe ktorých sa celý trénovací dátový súbor použije na natrénovanie finálneho modelu. Tento model následne slúži na hodnotenie predikčných metrík na testovacej množine.

4.2 Výpočtová náročnosť trénovania

Čas trénovania modelov sa výrazne líšil v závislosti od veľkosti a komplexnosti jednotlivých dátových súborov. Kľúčovým faktorom bol najmä počet vstupných premenných a celkový rozsah dát.

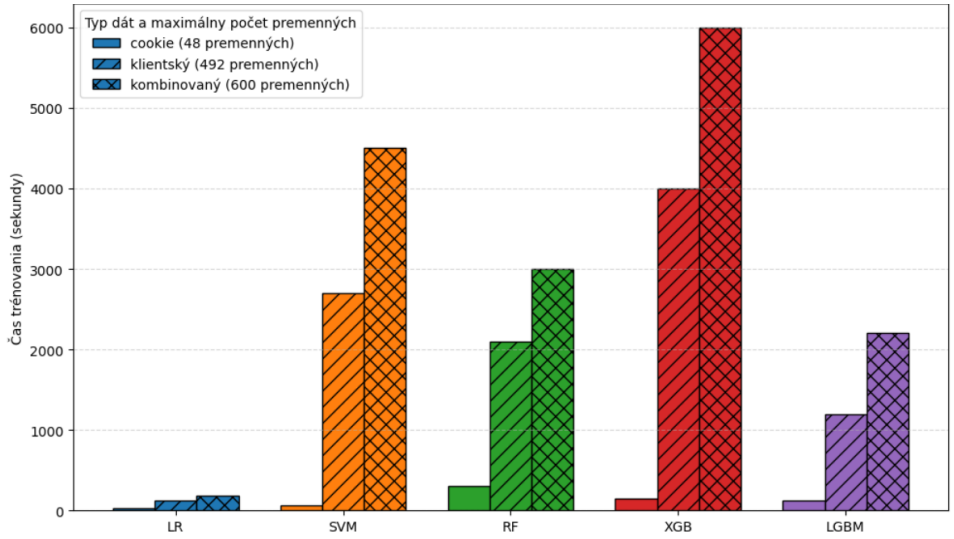
Trénovanie všetkých modelov bolo realizované na jednom serveri s 8 jadrami procesora a 64 GB operačnej pamäte. Použitie jednotnej hardvérovej konfigurácie zabezpečuje konzistentnosť a porovnateľnosť nameraných časov trénovania naprieč jednotlivými modelmi.

Pre príklad niektorých metód uvádzame približné časové nároky na ich trénovanie. SVM má podľa [32] tréningovú zložitosť[12] približne od $O(n^2)$ do $O(n^3)$, XGB[19] má zložitosť $O(K \cdot d \cdot \|x\|_0 \cdot \log n)$ a LGBM dosahuje nižšiu zložitosť približne $O(N \cdot d)$, keďže jeho výpočtová náročnosť je úmerná počtu dát a premenných a je redukovaná pomocou histogramového prístupu a bundlovania premenných [20].
Jednotlivé časy pre trénovanie modelov vidíme na obr. 10.

- Z výsledkov je zrejmé, že s rastúcim počtom premenných rastie aj čas trénovania naprieč všetkými modelmi. Cookie model, ktorý obsahuje najmenej premenných, dosahuje najnižšie časy trénovania, zatiaľ čo klientský model s vyšším počtom premenných vykazuje výrazne dlhšie časy. Kombinovaný model, ktorý obsahuje najviac premenných, je zároveň výpočtovo najnáročnejší. Tento trend je konzistentný pre všetky uvažované klasifikačné metódy.
- Logistická regresia vykazuje približne lineárny nárast času trénovania so zvyšujúcim sa počtom premenných, čo potvrdzuje jej nižšiu výpočtovú náročnosť. Naopak, komplexnejšie modely ako SVM, RF či boosting metódy vykazujú výrazne rýchlejší nárast času.

[12]Symbol $O(\cdot)$ označuje asymptotickú výpočtovú zložitosť, pričom N (resp. n) predstavuje počet vzoriek, d počet premenných, K počet stromov a $\|x\|_0$ počet nenulových prvkov v dátach; tento zápis vyjadruje, ako rastie čas tréningu so zväčšujúcou sa veľkosťou vstupu.

- Pri klientskom a kombinovanom modeli je napríklad viditeľné, že XGB je približne dvojnásobne pomalší než Random Forest a zároveň výrazne pomalší než LGBM, ktoré je približne trikrát rýchlejšie než XGB. Tento rozdiel poukazuje na efektívnosť optimalizačných techník použitých v LGBM, ktoré sa výraznejšie prejavujú najmä pri väčších a komplexnejších dátových súboroch.



Obr. 10: Čas trénovania (v sekundách) klasifikačných metód, modelov a maximálny počet premenných

Z uvedeného porovnania (obr. 10) možno pozorovať, že so zvyšujúcou sa dimenzionalitou a komplexnosťou dát dochádza k nárastu výpočtovej náročnosti trénovania modelov naprieč všetkými klasifikačnými metódami.

Pri trénovaní modelov bola zároveň vykonaná optimalizácia hyperparametrov pomocou knižnice Optuna, bližšie popísané v Prílohe A (aj s tabuľkou hyperparametrov, ich veľkosťami a popisom v tab. 17). Pre každý model bolo realizovaných približne 100 pokusov (angl. trialov), čo predstavuje kompromis medzi kvalitou optimalizácie a výpočtovou náročnosťou. Vzhľadom na relatívne dlhý čas trénovania komplexnejších modelov by vyšší počet trialov viedol k výraznému nárastu výpočtového času bez výrazného zlepšenia výsledkov.

4.3 Konštrukcia testovacieho dátového súboru

Po určení a natrénovaní finálneho modelu sme následne pristúpili k vytvoreniu testovacej vzorky, ktorá slúžila na vyhodnotenie predikčnej schopnosti modelov. Týmto postupom sme zabezpečili, že testovacie dátove súbory mali rovnakú štruktúru naprieč všetkými modelmi a zároveň hodnotenie metrík bolo založené na identickej množine pozorovaní, čo umožňuje priamu porovnateľnosť výsledkov jednotlivých modelov.

Na vyhodnotenie predikčnej schopnosti modelov sme ako testovaciu vzorku zvolili jeden budúci *month_id*, ktorý sa nenachádzal v trénovacej množine dát. Takýmto spôsobom sme zabezpečili, že modely boli hodnotené na dátach z neskoršieho obdobia, ktoré neboli použité počas trénovania.

Zároveň sme sa snažili zabezpečiť porovnateľnosť medzi jednotlivými modelmi a výsledkami. Keďže cookie a kombinovaný model obsahujú viac pozorovaní/riadkov pre jedného klienta v prípade, že má viac cookies, rovnaký princíp sme aplikovali aj pri testovacej množine klientského modelu. To znamená, že ak sa klient v testovacej množine cookie dátoveho súboru vyskytoval napríklad v piatich riadkoch kvôli piatim rôznym cookies, rovnaký počet riadkov bol vytvorený aj v klientskom dátovom súbore. V tomto prípade však klientské premenné zostávali rovnaké a jednotlivé pozorovania boli iba duplikované. Počet pozitívnych a negatívnych pozorovaní v testovacej množine pre finančné produkty a pre zvolený *month_id* je uvedený v tabuľke 10.

Tabuľka 10: Počet pozitívnych a negatívnych vzoriek v testovacej množine

Závislá premenná / Hodnota binárnej premennej	Spotrebný úver	Jednorázová investícia
0 (nenastal nákup)	57 023	56 993
1 (nastal nákup)	397	427

Pre všetky modely sa na testovaciu vzorku následne napojili premenné vybrané v procese selekcie premenných, pričom testovacie dátové súbory obsahovali presne tie isté premenné a v rovnakom poradí ako trénovacie.

5 Kvantifikácia vplyvu cookie dát

Cieľom tejto kapitoly bolo overiť, do akej miery jednotlivé typy dát dokážu predikovať pravdepodobnosť nákupu produktu, a zároveň porovnať predikčnú schopnosť modelov. Očakávalo sa, že model využívajúci kombinované dáta bude vzhľadom na širší rozsah informácií, zahŕňajúcich správanie používateľov na webe aj interné klientské záznamy, dosahovať najlepšie výsledky. V tejto kapitole sa preto venujeme analýze výsledkov jednotlivých modelov a kvantifikácii vplyvu rôznych dátových zdrojov na ich predikčnú výkonnosť, pričom modely porovnávame z pohľadu použitých dátových súborov, klasifikačných metód, ale aj podľa jednotlivých produktov.

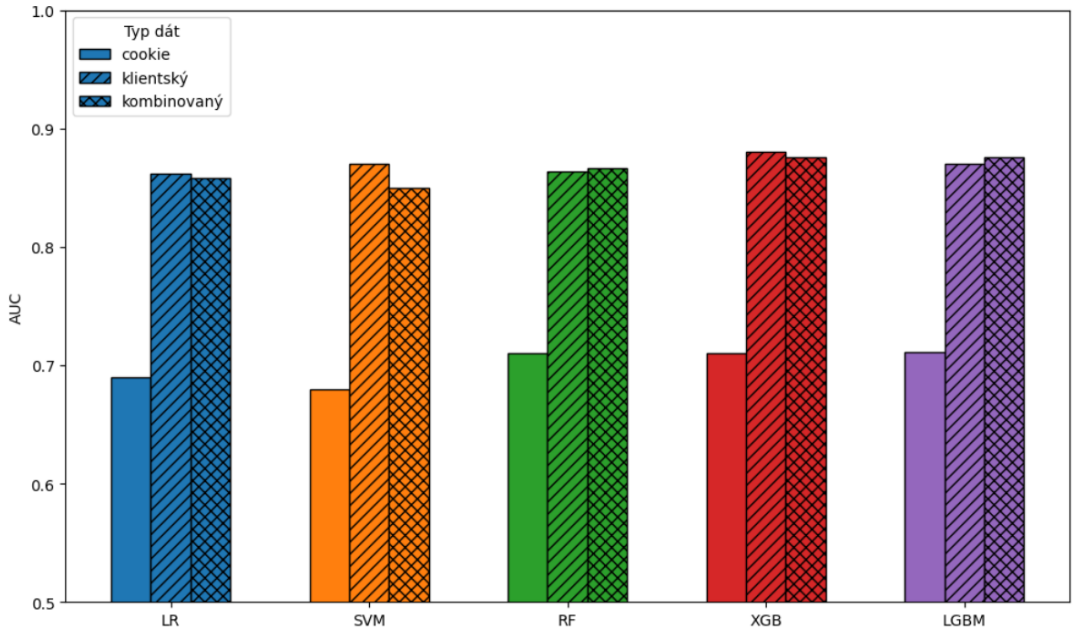
Výber metrík pre vyhodnotenie modelov

V nadväznosti na predchádzajúcu analýzu testovacích dát budeme v ďalšej časti práce klásť hlavný dôraz na metriky AUC a lift. Tieto metriky považujeme za najvhodnejšie, keďže AUC zachytáva celkovú diskriminačnú schopnosť modelu nezávisle od zvoleného prahu, metrika lift má priamu interpretáciu z pohľadu biznisovej aplikácie a sú menej citlivé na nevyváženosť dát.

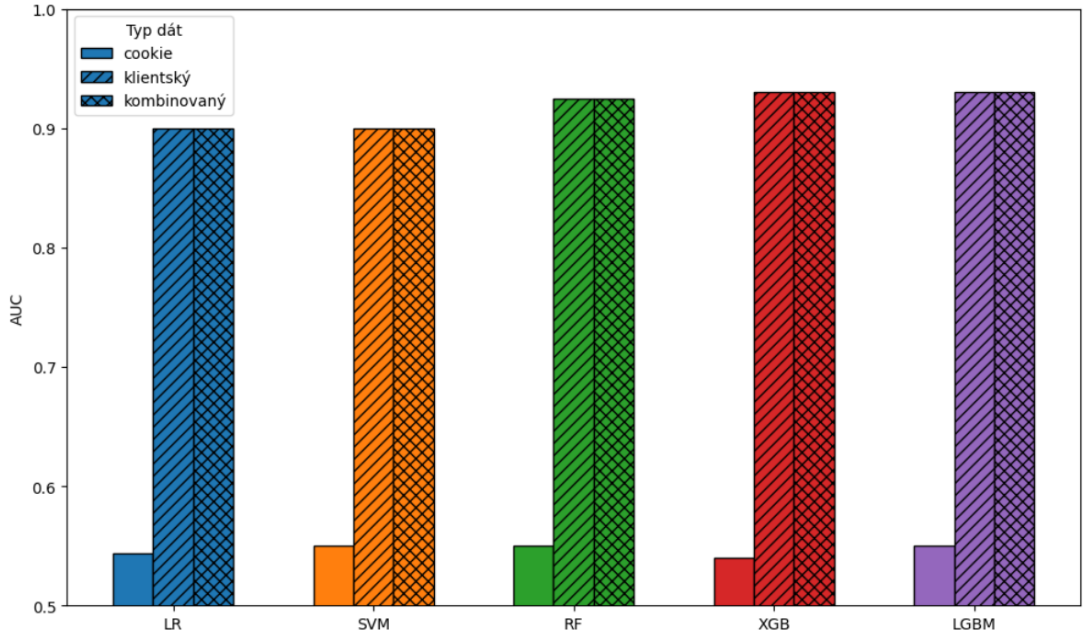
Naopak, metriky accuracy a precisiom-recall nebudú tvoriť hlavnú časť vyhodnotenia. V prípade accuracy je dôvodom nevyváženosť testovacích vzoriek, kde prevláda negatívna trieda, čo môže viesť k nadhodnoteniu kvality modelu aj pri slabšej schopnosti identifikovať pozitívne prípady. Podobne aj metrika precision-recall je v našom prípade ovplyvnená nevyváženosťou dát, čo sa prejavuje nízkymi hodnotami (približne v intervale 0,05 až 0,15) a sťažuje jej interpretáciu a porovnateľnosť medzi modelmi.

5.1 Porovnanie výkonnosti modelov

Na Obr. 11, 12 vidíme porovnanie výkonnosti jednotlivých modelov podľa typu použitých dát (cookie, klientské a kombinované dáta) pre oba produkty podľa AUC. Je potrebné poznamenať, že osi grafov sú priblížené, aby boli lepšie viditeľné aj relatívne malé rozdiely medzi modelmi.



Obr. 11: Spotrebný úver - porovnanie AUC podľa typu modelu a klasifikačnej metódy



Obr. 12: Jednorázová investícia - porovnanie AUC podľa typu modelu a klasifikačnej metódy

Z pohľadu metriky AUC pozorujeme konzistentný trend naprieč všetkými klasifikačnými metódami aj oboma produktmi. Modely založené výlučne na cookie dátach dosahujú výrazne nižšie hodnoty (približne 0,71 pre spotrebný úver a 0,54 pre jednorázovú investíciu), čo naznačuje obmedzenú informačnú hodnotu týchto dát pri samostatnom použití.

Naopak, klientské modely dosahujú výrazne vyššie hodnoty AUC (približne 0,86 až 0,88 pre spotrebný úver a až okolo 0,90 až 0,93 pre jednorázovú investíciu). To potvrdzuje, že klientské dáta obsahujú vysokú informatívnu hodnotu pre predikciu správania klienta.

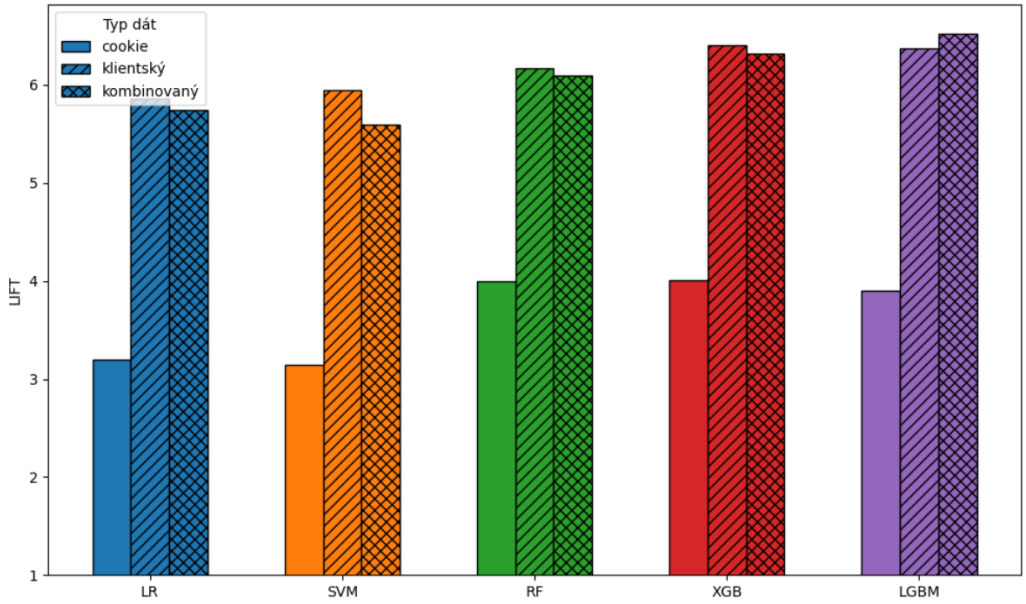
Kombinované modely dosahujú porovnateľné hodnoty pre úver (pre niektoré klas. metódy mierne vyššie hodnoty napr. RF, LGBM, ale v niektorých prípadoch mierne nižšie hodnoty napr. LR, SVM) ako klientské modely, avšak tieto rozdiely sú relatívne malé. Pre investíciu je kombinované AUC rovnaké pre všetky klas. modely ako klientské. Tieto výsledky naznačujú, že pridaná informácia z cookie dát vie, ale aj nemusí byť vždy prínosná z pohľadu diskriminačnej schopnosti modelu.

To, že pre investičný produkt dosahujú kombinované modely rovnaké hodnoty AUC ako klientské modely, možno vysvetliť priamo z definícií ROC krivky a AUC. Podľa Definície 2.1 je ROC krivka určená hodnotami $\mathrm{TPR}(\tau)$ a $\mathrm{FPR}(\tau)$, ktoré závisia od množiny pozorovaní, pre ktoré platí $Y^{pred} > \tau$. Ak pridanie cookie premenných nemení túto množinu pozorovaní pre jednotlivé hodnoty prahu τ, potom zostávajú nezmenené aj hodnoty $\mathrm{TPR}(\tau)$ a $\mathrm{FPR}(\tau)$ pre všetky τ. Z toho vyplýva, že ROC krivky klientského a kombinovaného modelu sú totožné. Z Definície 2.2 zároveň platí, že $\mathrm{AUC} = P\left(Y_1^{pred} \geq Y_0^{pred}\right)$, teda ide o pravdepodobnosť, že náhodne zvolené pozitívne pozorovanie nadobúda vyššiu predikčnú hodnotu klasifikátora než náhodne zvolené negatívne pozorovanie. Rovnosť hodnôt AUC preto implikuje, že pravdepodobnosť nerovnosti $Y_1^{pred} \geq Y_0^{pred}$ zostáva po pridaní cookie premenných nezmenená. Inými slovami, kombinovaný model nemení vzájomné porovnanie predikčných hodnôt medzi pozitívnymi a negatívnymi pozorovaniami.

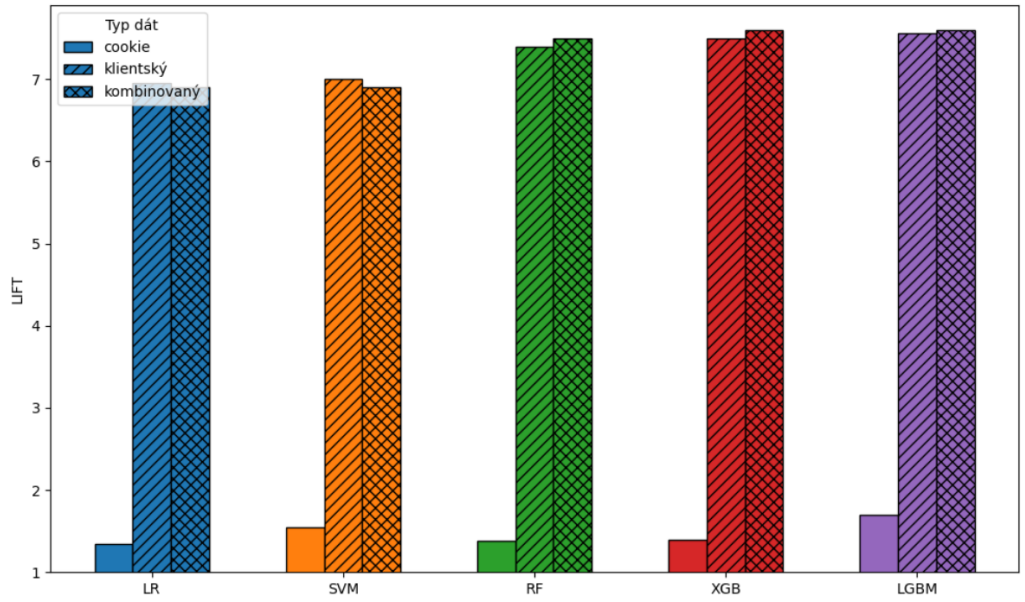
Tento výsledok naznačuje, že cookie premenné pre investičný produkt neprinášajú dodatočnú diskriminačnú informáciu nad rámec klientskych dát, čo je v súlade s ich nízkou samostatnou predikčnou výkonnosťou.

Z pohľadu klasifikačných metód dosahujú najlepšie výsledky modely založené na boostingu (XGB, LGBM), nasledované Random Forest, zatiaľ čo lineárne metódy (LR, SVM) mierne zaostávajú.

Na Obr. 13, 14 sledujeme podobné porovnania ako pre AUC, ale tento krát pre metriku lift (pre top 10% pozorovaní).



Obr. 13: Spotrebný úver - porovnanie LIFT podľa typu modelu a klasifikačnej metódy



Obr. 14: Jednorázová investícia - porovnanie LIFT podľa typu modelu a klasifikačnej metódy

Pozorujeme podobný trend aj pri tejto metrike, ktorá má priamu interpretáciu z pohľadu biznisového využitia. Cookie modely dosahujú výrazne nižšie hodnoty (približne 3 až 4 pre spotrebný úver a okolo 1,3 až 1,7 pre jednorázovú investíciu), čo znamená, že ich schopnosť identifikovať relevantných klientov je obmedzená.

Klientské modely dosahujú výrazne vyššie hodnoty liftu (približne 5,8 až 6,4 pre spotrebný úver a až 6,9 až 7,6 pre jednorázovú investíciu), čo predstavuje výrazné zlepšenie oproti náhodnému výberu.

Kombinované modely opäť dosahujú veľmi podobné (opäť pre niektoré klas. metódy mierne vyššie a pre niektoré klas. metódy mierne nižšie) hodnoty ako klientské modely, pričom rozdiely sú relatívne malé. Pre úver vidíme lift pre kombinovaný model vo všetkých prípadoch jemne nižší ako pre klientský, až na metódu LGBM. Pre investíciu platí pokles liftu v kombinovanom len pre lineárne modely, pričom tie ostatné zaznamenávajú jemný nárast. To naznačuje, že prínos cookie dát nad rámec klientských dát je z pohľadu praktickej aplikácie limitovaný.

Aj pri tejto metrike dosahujú najlepšie výsledky boostingové metódy (XGB, LGBM), čo potvrdzuje ich vhodnosť pre daný typ úlohy.

Z oboch metrík vyplýva, že klientské dáta majú dominantný vplyv na kvalitu modelov, zatiaľ čo cookie dáta samostatne nedosahujú dostatočnú výkonnosť. Kombinácia dát prináša len v niektorých prípadoch mierne zlepšenie, čo naznačuje, že väčšina predikčnej informácie je obsiahnutá už v klientských dátach.

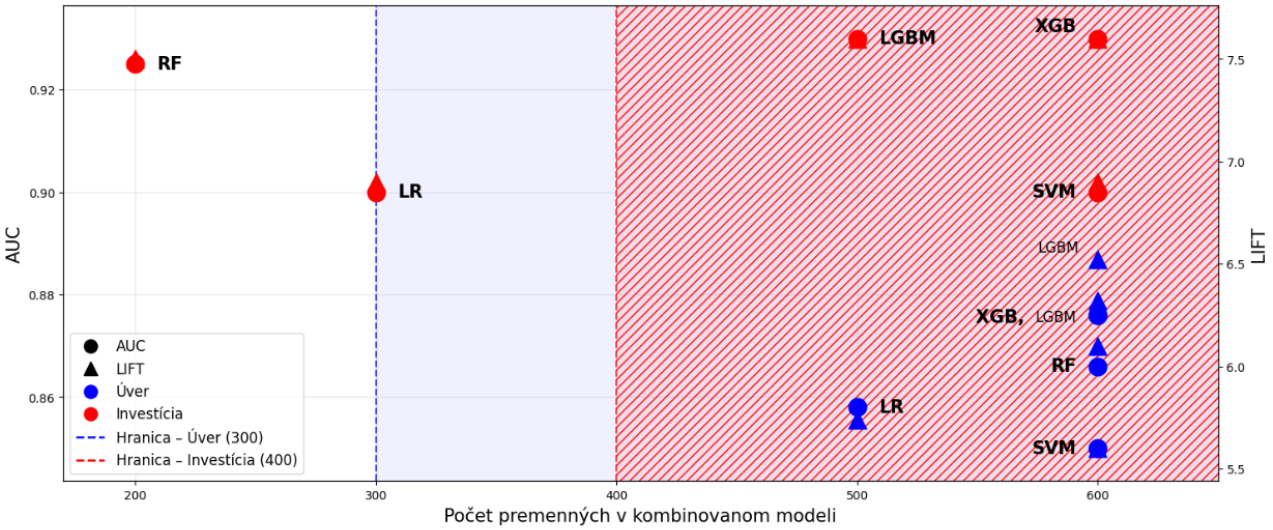
Zároveň je vhodné poukázať na rozdiel medzi produktmi pri použití výhradne cookie dát. Pre spotrebný úver dosahujú cookie modely relatívne použiteľnú výkonnosť, čo naznačuje, že aj samotné správanie na webe obsahuje určitý predikčný signál. Hoci táto výkonnosť nedosahuje úroveň klientských modelov, môže byť v praxi využiteľná najmä v prípadoch, kde klientské dáta nie sú dostupné.

Naopak, pre jednorázovú investíciu sú výsledky cookie modelov výrazne slabšie, čo je len mierne lepšie ako náhodný výber. V tomto prípade teda cookie dáta samostatne neposkytujú dostatočný predikčný signál a ich praktická využiteľnosť je veľmi obmedzená.

Tento rozdiel medzi produktmi je zaujímavý aj z pohľadu správania používateľov na webovej stránke `https://www.csob.sk/`, kde sa klient vie relatívne jednoducho prekliknúť na podstránky týkajúce sa jednotlivých produktov (či už úveru alebo investície). Napriek tejto jednoduchej navigácii však cookie dáta poskytujú výrazne silnejší predikčný signál pre spotrebný úver ako pre jednorázovú investíciu.

5.2 Porovnanie metrík vzhľadom na výsledný počet premenných

V tejto časti budeme sledovať, ako sa mení kvalita modelov pri rôznom výslednom počte premenných, aby sme zistili, či väčšie množstvo informácií (najmä z cookies) skutočne zlepšuje predikciu.



Cookie:	**0**	**1**	**6**	**21**	**40**
Cookie:	**0**	**0**	**1**	**12**	**32**

Obr. 15: Porovnanie metrík, klasifikačných metód a produktu podľa AUC/LIFT vzhľadom na výsledný počet premenných

Na Obr. 15 je znázornený vzťah medzi výsledným počtom premenných v kombinovanom modeli a výkonnosťou jednotlivých klasifikačných metód meranou pomocou metrík AUC (ľavá os) a lift (pravá os). Na horizontálnej osi je uvedený počet vybraných premenných, ktorý bol optimalizovaný pomocou hyperparametrickej optimalizácie (Optuna, popísané v prílohe A) v krokoch po 100, pričom maximálny počet premenných bol obmedzený na 600.

Graf rozlišuje jednotlivé výsledky podľa produktu (úver – modrá farba, investícia – červená farba), použitej klasifikačnej metódy (LR, RF, SVM, LGBM, XGB) a metriky (AUC znázornené kruhmi, lift trojuholníkmi).

Ako sme videli na obrázkoch vyššie, z pohľadu výkonnosti modelov dosahujú najvyššie hodnoty AUC modely založené na boostingu. Lineárne modely dosahujú nižšie hodnoty (najnižšie v porovnaní s klas. metódami pre rovnaké produkty), pričom RF vykazuje

relatívne vysoký výkon už pri nižšom počte premenných (okolo 200). Pre úverový produkt sú hodnoty AUC vo všeobecnosti nižšie v porovnaní s investíciou a rozdiely medzi modelmi sú menej výrazné. Podobný trend je možné pozorovať aj pri metrike lift. Vhodnosť grafu bola zvolená na základe prekrytia metrík pre rovnaké klasifikačné metódy a produkty (preto sledujeme napríklad vznik "kvapiek" na grafe, čo znamená prekrytie metrík pre ten istý produkt a klas. metódu).

Z hľadiska počtu premenných väčšina modelov dosahuje najlepšie výsledky pri vyšších dimenziách (500–600 premenných). Výnimkou je napríklad Random Forest pre investíciu, ktorý dosahuje vysoký výkon už pri menšom počte premenných. Logistická regresia je použitá pre oba produkty pri nižších dimenziách ako ostatné klas. metódy, pričom jej výkon je stabilný, ale nižší v porovnaní s pokročilejšími modelmi.

Vertikálne čiary v grafe indikujú hranice, od ktorých sa v modeloch začínajú objavovať cookie premenné. Modrá prerušovaná čiara (od 300 premenných) reprezentuje hranicu pre úverový produkt, zatiaľ čo červená prerušovaná čiara (od 400 premenných) pre investičný produkt. Vyfarbené resp. vyšrafované oblasti podľa farby zodpovedá rozsahu, v ktorom modely pre produkty obsahujú cookie premenné. Z toho vyplýva, že modely s nižším počtom premenných neobsahujú cookie premenné a tieto sa začínajú vyberať až pri vyšších dimenziách modelu. Modely, ktoré si nevybrali žiadnu cookie premennú sa vyskytujú iba pri investícii (konkrétne RF a LR).

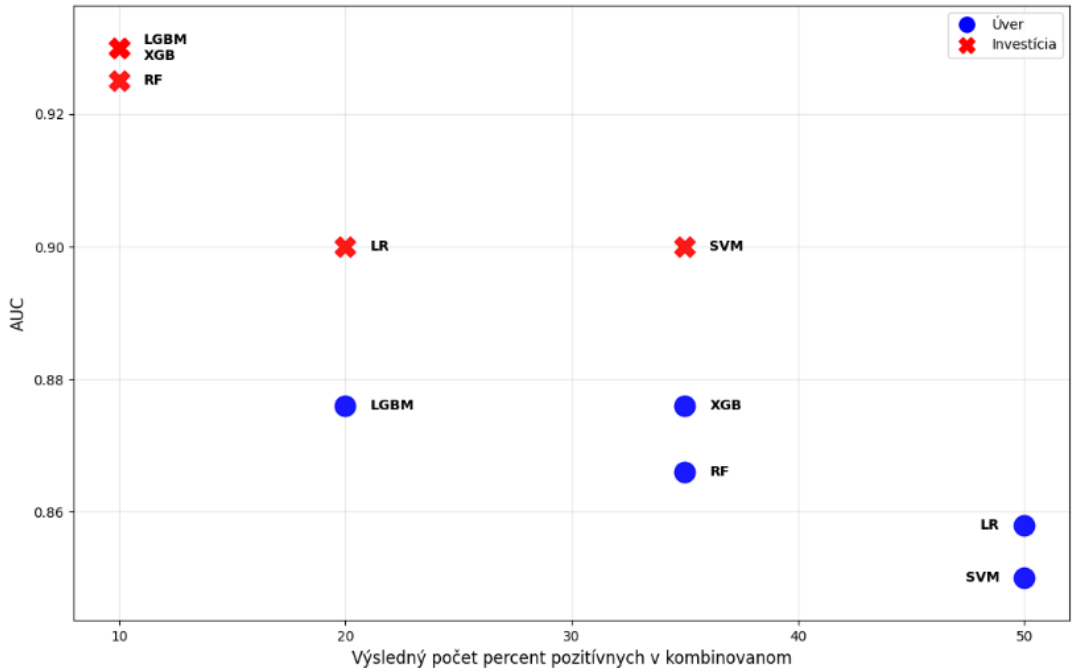
Tabuľka pod grafom zobrazuje počet cookie premenných v závislosti od celkového počtu premenných v modeli. Napríklad pri modeli so 400 premennými sa v prípade úveru vyskytuje 6 cookie premenných, zatiaľ čo pre investíciu iba 1 cookie premenná. Je zrejmé, že podiel cookie premenných rastie s celkovým počtom premenných a významnejšie sa presadzujú až pri väčších modeloch (500–600 premenných).

Na záver možno konštatovať, že kombinovaný model dosahuje najlepšie výsledky pri vyššom počte premenných, pričom najvýkonnejšie sú boostingové metódy. Zároveň cookie premenné predstavujú doplnkový zdroj informácie, ktorý sa do modelu dostáva až pri vyšších dimenziách a v niektorých prípadoch jemne zvyšuje jeho predikčnú schopnosť oproti iba klientským dátam.

5.2.1 Porovnanie metrík vzhľadom na výsledný počet pozorovaní

V tejto časti budeme analyzovať vplyv veľkosti dátovej vzorky na výkonnosť modelov, s cieľom overiť, či viac pozorovaní vedie k stabilnejším a presnejším odhadom.

Na Obr. 16 je znázornený vzťah medzi percentom pozitívnych prípadov v trénovacom dátovom súbore a výkonnosťou klasifikačných modelov meranou pomocou metriky AUC (na testovacej množine). Na horizontálnej osi sú zobrazené hodnoty počtu percent pozitívnej triedy, ktoré boli optimalizované pomocou nástroja Optuna (popísané v prílohe A). Ide o hodnoty 10 %, 20 %, 35 % a 50 %, pričom napríklad hodnota 10 % znamená, že v testovacom dátovom súbore sa nachádza 10 % pozitívnych pozorovaní a 90 % negatívnych, čo zároveň vedie k väčšiemu počtu pozorovaní v testovacej vzorke.



Obr. 16: Porovnanie klasifikačných metód, produktu a počtu percent pozitívnych pozorovaní v trénovacej množine podľa AUC (na testovacej množine)

Graf rozlišuje jednotlivé výsledky podľa produktu (úver – modré body, investícia – červené krížiky) a použitej klasifikačnej metódy. Zobrazená je výlučne metrika AUC (v prípade metriky lift by sme pozorovali veľmi podobné vzťahy, preto sme sa rozhodli iba pre graf s AUC). Kombinované zobrazenie AUC a lift v jednom grafe by však v tomto prípade znižovalo prehľadnosť vizualizácie.

Z pohľadu výkonnosti modelov je možné pozorovať, že pre investičný produkt dosahujú

najvyššie hodnoty AUC modely LGBM, XGB a RF pri najnižšom podiele pozitívnych prípadov v trénovacej množine (10 %). S rastúcim podielom pozitívnej triedy (20 % a 35 %) dochádza k miernemu poklesu výkonnosti, pričom opäť vidno, že lineárne modely zaostávajú.

Pre úverový produkt sú hodnoty AUC nižšie v porovnaní s investičným produktom. Najlepšie výsledky dosahujú modely LGBM a XGB pri prostredných hodnotách podielu pozitívnych prípadov (20 % až 35 %). Pri najvyššom podiele pozitívnej triedy (50 %) dochádza k poklesu výkonnosti, pričom polovicu pozitívnych pozorovaní v testovacej množine vyberali iba lineárne modely.

Celkovo možno konštatovať, že nižší podiel pozitívnych prípadov (viac nevyvážený trénovací dátový súbor) vedie k vyšším hodnotám AUC, najmä pre investičný produkt. S rastúcim podielom pozitívnej triedy sa výkonnosť modelov mierne znižuje, čo naznačuje citlivosť modelov na zastúpenie tried.

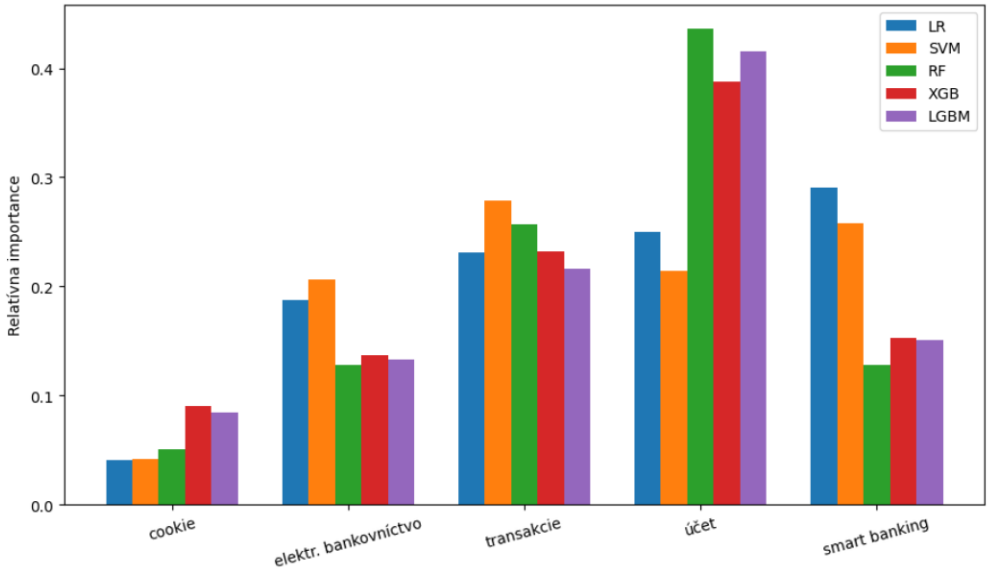
5.3 Kvantifikácia cookie premenných v kombinovanom modeli

V tejto časti sa zameriame na meranie príspevku cookie premenných v kombinovanom modeli, aby sme kvantifikovali ich pridanú hodnotu oproti samotným klientským dátam.

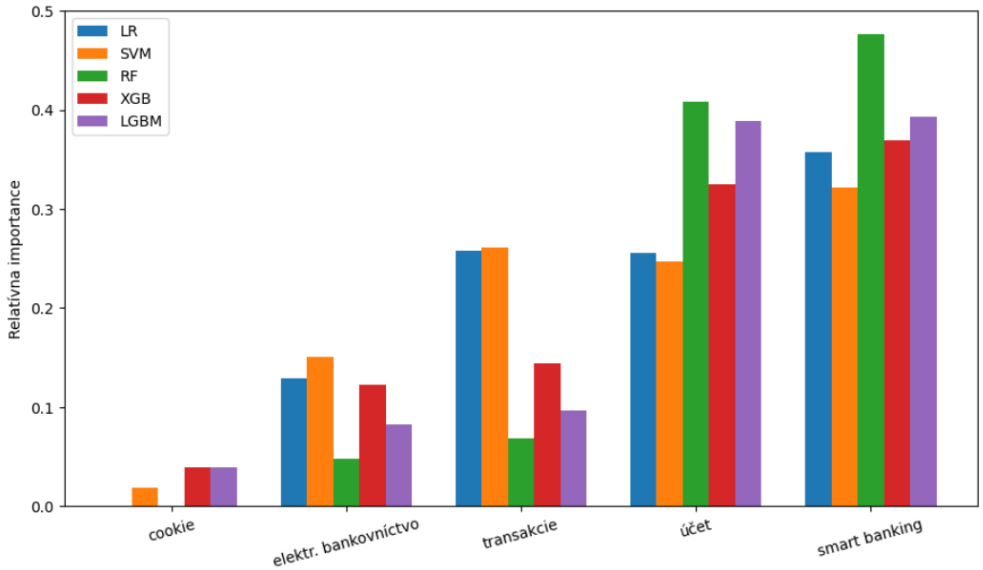
Na obrázkoch 17 a 18 je znázornená relatívna dôležitosť jednotlivých dátových zdrojov v kombinovanom modeli pre produkty spotrebný úver a jednorazová investícia. Dôležitosť je kvantifikovaná pomocou SHAP hodnôt, ktoré vyjadrujú priemerný príspevok jednotlivých premenných k výslednej predikcii modelu.

Keďže jednotlivé dátové súbory (napr. webové dáta, elektronické bankovníctvo, transakčné dáta, údaje o účte a smart banking) obsahujú rôzny počet premenných vstupujúcich do testovacej vzorky a majú odlišnú štruktúru, absolútne SHAP hodnoty nie sú priamo porovnateľné medzi dátovými súbormi. Tento problém je navyše ovplyvnený aj tým, že rôzne klasifikačné modely využívajú odlišné prístupy k výpočtu SHAP hodnôt (lineárne modely vychádzajú z tzv. LinearExplainer, zatiaľ čo modely založené na stromoch využívajú TreeExplainer, pričom metóda RF má dokonca dve dimenzie SHAP hodnôt na rozdiel od ostatných modelov), čo vedie k odlišnej škále a distribúcii výsledných hodnôt. Z tohto dôvodu sme pristúpili k ich relativizácii, ktorá umožňuje porovnanie významu jednotlivých dátových zdrojov na jednotnej škále.

Relativizácia SHAP hodnôt je definovaná ako podiel súčtu absolútnych SHAP hodnôt premenných patriacich do daného dátového súboru na celkovom súčte SHAP hodnôt všetkých premenných v modeli. Táto relatívna dôležitosť umožňuje interpretovať výsledky ako percentuálny príspevok jednotlivých dátových zdrojov k celkovej predikcii modelu. Súčet relatívnych dôležitostí naprieč všetkými dátovými súbormi je rovný 1.



Obr. 17: Spotrebný úver - relatívna SHAP importance podľa dátového súboru a klasifikačných metód



Obr. 18: Jednorázová investícia - relatívna SHAP importance podľa dátového súboru a klasifikačných metód

Z výsledkov je možné pozorovať, že najvýznamnejší príspevok k predikcii pre úver majú premenné z tabuľky o účte klienta a transakciach. Lineárne modely pripisujú väčší príspevok pre smart banking dáta (až okolo 30%), no RF a boostingové metódy jednoznačne preferujú dátový súbor o účte klienta (až na úrovni 40%). Naopak, webové dáta (resp. cookie premenné) majú relatívne nižší príspevok, kde pre boostingové metódy cookie premenné prispievajú ku predikcii okolo 8-9% a pre lineárne s RF okolo 4% zo všetkých dátových súborov. Nižšia relatívna dôležitosť cookie premenných preto neznamená ich zanedbateľnosť, ale skôr doplnkový charakter voči iným stabilnejším zdrojom informácií.

Pri investícii vidíme boostingové metódy s RF preferovať opäť údaje o účte, pričom smart banking dáta významnejšie prispeli ku predikcii pre tento produkt. Lineárne metódy výraznejšie podporili údaje o transakciach oproti iným klas. metódam, no taktiež sú výraznejšie pre účet a smart banking. Vidíme, že cookie premenné pre investíciu majú ešte nižšie hodnoty ako pre úver. Najväčšie okolo 5% príspevku ku predikcii dosahujú boostingové metódy, pričom vidíme, že metódy LR a RF si cookie premenné ani nevybrali.

Z pohľadu interpretácie SHAP hodnôt platí, že vyššia hodnota relatívnej dôležitosti znamená, že daný dátový zdroj má väčší vplyv na výslednú predikciu pravdepodobnosti nákupu produktu. Inými slovami, model pri rozhodovaní viac využíva informácie z tohto zdroja.

Rozdiely medzi produktmi naznačujú, že význam jednotlivých dátových zdrojov je produktovo špecifický. Napríklad pri jednorazovej investícii zohrávajú výraznejšiu úlohu smart banking dáta, zatiaľ čo pri spotrebnom úvere sú dosť dôležité aj transakčné dáta, pričom pre oba produkty je tabuľka o bankovom účte konzistentne dôležitá.

Záverom možno konštatovať, že kombinovaný model efektívne využíva informácie z rôznych dátových zdrojov, pričom cookie premenné predstavujú doplnkovú informáciu, ktorý môže zlepšiť predikciu.

5.3.1 Identifikácia konkrétnych cookie premenných v kombinovanom modeli

Na identifikáciu najvýznamnejších cookie premenných v kombinovaných modeloch sme analyzovali hodnoty SHAP a pre každý model a produkt vybrali päť najdôležitejších premenných. Konkrétne definície jednotlivých premenných sú uvedené v Prílohe B, preto sa v tejto časti obmedzujeme len na ich názvy. Namiesto samostatného hodnotenia jednotlivých modelov sa zameriavame na identifikáciu opakujúcich sa vzorcov naprieč rôznymi

klasifikačnými metódami a produktmi.

Tabuľka 11: Hlavné skupiny významných cookie premenných v kombinovanom modeli

Skupina premenných	**Príklady premenných**
Recency premenná	*recency*
Referrer premenné	*c_referrer_...*
Metriky aktivity na webe	*avg_visit_pages*, *c_visit*, *avg_duration*
Premenné existujúcich produktov	*c_personal_account*, *c_insurance*, *c_other_services*
Premenné správania a kontextu	*i_device_android*, *i_onboarding_page*, časové premenné

Na základe vykonanej analýzy možno identifikovať niekoľko hlavných skupín cookie premenných, ktoré sa konzistentne objavujú medzi najvýznamnejšími naprieč modelmi a produktmi (tabuľka 11). Kompletný zoznam všetkých premenných spolu s ich detailným vysvetlením sa nachádza v Prílohe B.

Najvýznamnejšou premennou je *recency*, ktorá sa opakovane umiestňuje medzi najdôležitejšie vo väčšine modelov, čo potvrdzuje význam aktuálnosti interakcie klienta s webovou stránkou pri predikcii jeho správania. Zároveň treba povedať, že pri investičnom produkte patrí premenná *recency* v boostingových metodách medzi najsilnejšie premenné zo všetkých uvažovaných premenných.

Ďalšiu významnú skupinu tvoria premenné reprezentujúce zdroj návštevy (referrer), ktoré zohrávajú dôležitú úlohu najmä pri spotrebnom úvere. Metriky aktivity na webe indikujú, že vyššia úroveň interakcie používateľa s webovou stránkou je spojená s vyššou pravdepodobnosťou realizácie sledovanej udalosti.

Významné sú aj premenné zachytávajúce existujúci vzťah klienta s bankou, ktoré ovplyvňujú jeho ďalšie správanie. Premenné správania a kontextu sa vyskytujú najmä pri investičnom produkte a odrážajú technické a časové charakteristiky interakcie používateľa.

Medzi produktmi možno zároveň pozorovať určité rozdiely. Pri spotrebnom úvere dominujú premenné spojené so zdrojom návštevy (referrer) a celkovou aktivitou na webovej stránke. Naopak pri jednorazovej investícii majú väčší význam premenné súvisiace s existujúcimi produktmi klienta (najmä *c_insurance*) a časové charakteristiky správania.

Tabuľka 12: Top päť cookie premenných z kombinovaného modelu vybraných zo všetkých premenných zoradených podľa SHAP hodnôt (od najvyšších po najnižšie), podľa produktu a klasifikačnej metódy.

Model	Spotrebný úver	Jednorázová investícia
LGBM	17, 22, 25, 27, 29	5, 288, 329, 342, 386
LR	29, 36, 40, 99, 110	nie sú
XGB	15, 19, 23, 25, 42	nie sú
RF	37, 59, 73, 79, 82	7, 27, 92, 115, 223
SVM	57, 69, 76, 109, 113	43, 98, 101, 111, 147

Vo všeobecnosti možno konštatovať, že cookie premenné sa pri hodnotení podľa SHAP hodnôt neumiestňujú na najvyšších priečkach medzi všetkými premennými modelov. Sledujeme to v tabuľke 12, kde uvedené hodnoty predstavujú pozície top päť cookie premenných v rámci SHAP poradia všetkých vstupných premenných daného modelu. Nižšia hodnota teda znamená vyšší význam pre danú predikciu, zatiaľ čo vyššia hodnota indikuje menší relatívny prínos danej premennej v rámci modelu.

Celkovo výsledky potvrdzujú, že cookie premenné nepatria medzi dominantné vysvetľujúce premenné modelov, avšak v niektorých prípadoch, najmä pri boostingových metódach, poskytujú dodatočnú informačnú hodnotu pre predikciu správania klienta.

Zároveň možno poznamenať, že lineárne modely zaraďujú cookie premenné podľa SHAP hodnôt skôr na nižšie pozície v porovnaní s ostatnými premennými.

Výsledky naznačujú, že cookie dáta nie sú zanedbateľné a v niektorých prípadoch poskytujú významnú pridanú hodnotu.

5.4 Porovnanie vážených a neválených metrík predikcie

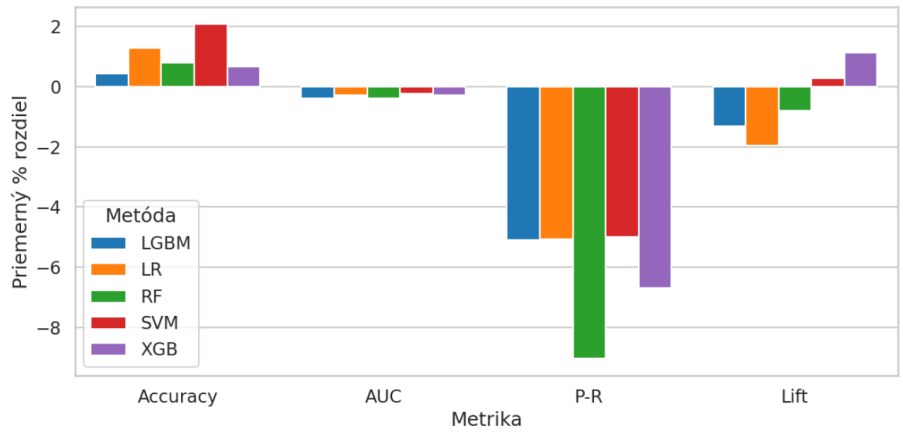
V práci (podkapitola 2.3) sme odvodili vážené metriky predikcie kvality modelu a v tejto podkapitole ilustrujeme to, ako sa odchylujú od bežných metrík. Cieľom tejto analýzy je overiť, či spôsob agregácie dát (na úrovni cookies vs. na úrovni klienta) má významný vplyv na hodnotenie kvality modelov. Tento krok je dôležitý najmä z metodického hľadiska, keďže nám umožní rozhodnúť, či je potrebné rozlišovať medzi týmito prístupmi pri interpretácii výsledkov, alebo či môžeme metriky považovať za ekvivalentné.

Percentuálna zmena metriky je vypočítaná ako podiel rozdielu medzi hodnotou pre vážený variant a hodnotou pre nevážený variant k hodnote duplicitného variantu, vynásobený 100 %, t.j.

$$\frac{x_{\text{vážené}} - x_{\text{nevážené}}}{x_{\text{nevážené}}} \cdot 100\ \%.$$

Čiže napríklad hodnota -8% znamená, že metrika pri váženom variante je o 8% nižšia v porovnaní s neváženým variantom, ktorý slúži ako referenčný základ.

V testovacej vzorke sa nachádza 49 766 unikátnych klientov, pričom 6 170 z nich sa vyskytuje vo viacerých pozorovaniach/riadkoch, čo predstavuje približne 12,4 % všetkých klientov. Priemerný počet riadkov na klienta je 1,15 a maximálny počet výskytov jedného klienta dosahuje hodnotu 15.



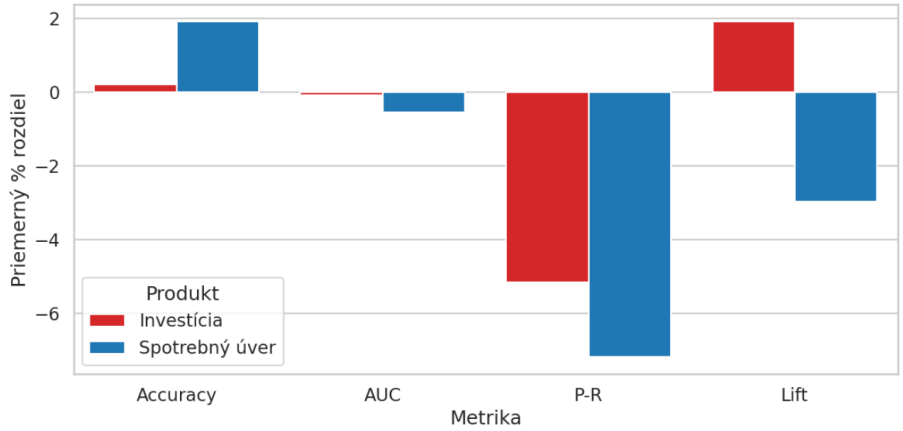
Obr. 19: Priemerné percentuálne rozdiely vážených a nevážených metrík podľa klas. metódy

Z Obr. 19 vidíme, že rozdiely medzi všetkými váženými a neváženými metrikami sú vo všetkých klasifikačných metódach relatívne malé. Metrika *accuracy* sa líši len minimálne (približne do 2 %), pričom vážené hodnoty sú vo väčšine prípadov mierne vyššie. Podobne aj metrika *AUC* vykazuje zanedbateľné rozdiely, čo naznačuje, že schopnosť modelov rozlišovať medzi triedami nie je ovplyvnená spôsobom váženia.

Výraznejšie rozdiely pozorujeme pri metrike *precision-recall*, kde sú vážené hodnoty systematicky nižšie, približne o 5 až 9 %, pričom najväčší pokles nastáva pri modeli Random Forest. Metrika *lift* vykazuje len mierne odchýlky (do približne 2 %), bez jednoznačného trendu naprieč modelmi.

Z Obr. 20, kde sú výsledky agregované podľa produktov, pozorujeme podobný obraz. Rozdiely v metrike *accuracy* a *AUC* sú opäť minimálne. Pri metrike *precision-recall* do-

chádza k poklesu vážených hodnôt (približne o 5 až 7 %) pre oba produkty. Metrika *lift* sa správa odlišne v závislosti od produktu (pri jednorazovej investícii sú vážené hodnoty vyššie, zatiaľ čo pri spotrebnom úvere nižšie).



Obr. 20: Priemerné percentuálne rozdiely vážených a neválených metrík podľa produktu

Na základe týchto výsledkov môžeme konštatovať, že rozdiel medzi váženými a neváženými metrikami je vo väčšine prípadov malý (rádovo do 2 %), s výnimkou metriky *precision-recall*. To znamená, že spôsob agregácie dát nemá zásadný vplyv na hodnotenie kvality modelov, a preto používame metriky predikcie bez ohľadu na ich váženie.

6 Praktická aplikácia optimalizácie marketingových oslovení v banke

V predchádzajúcich kapitolách sme sa venovali konštrukcii klasifikačných modelov a ich vyhodnoteniu pomocou štandardných metrík, ako sú AUC a lift. Z pohľadu bankovej praxe však samotná kvalita modelu nepostačuje. Kľúčovým krokom je transformácia výstupov modelov do konkrétnych rozhodnutí, ktoré určujú, aký obsah bude klientovi zobrazený na webovej stránke.

Cieľom tejto kapitoly je navrhnúť rozhodovací mechanizmus, ktorý na základe predikovaných pravdepodobností určí, či bude klientovi zobrazený banner pre úver, investičný produkt alebo žiadny banner.

6.1 Výber optimálneho klasifikačného modelu

Na základe výsledkov z kapitoly 5 bola ako optimálna klasifikačná metóda zvolená metóda LGBM. Táto metóda dosahovala konzistentne najlepšie výsledky z hľadiska metrík AUC a lift naprieč všetkými skúmanými modelmi, a tak isto ako jediná pre každú zo spomínaných metrík (a aj každý produkt) dosahovala aj jemne lepšie výsledky pre kombináciu dát v porovnaní s klientskými.

Z dôvodu zachovania porovnateľnosti medzi jednotlivými prístupmi (cookie model, klientský model a kombinovaný model) sme sa rozhodli použiť rovnakú klasifikačnú metódu pre všetky modely. Tento prístup umožňuje analyzovať vplyv použitých dát na kvalitu predikcie pri fixnej voľbe klasifikačného algoritmu.

Pre každý produkt (úver a investícia) a každý dátový prístup (cookie, klientsky, kombinovaný) bol opäť natrénovaný samostatný model.

6.1.1 Hyperparametre optimálnych modelov

V tejto časti uvádzame optimálne hyperparametre modelov LGBM pre jednotlivé dátové prístupy a produkty. Okrem toho uvádzame aj počet vybraných premenných a použitý pomer tried, ktorý bol použitý pri trénovaní modelu. Všetky hyperparametre sú popísane a vysvetlené v Prílohe A.

Z tabuľky je možné pozorovať rozdiely v komplexnosti modelov medzi jednotlivými

dátovými prístupmi. Klientské a kombinované modely využívajú výrazne vyšší počet premenných, zatiaľ čo cookie modely pracujú s menším počtom vstupov. Zároveň možno vidieť rozdiely v regularizácii a štruktúre stromov medzi produktmi úver a investícia.

Tabuľka 13: Optimálne hodnoty hyperparametrov modelov LGBM

	Úver			Investícia		
Hyperparameter	Cookie	Klientský	Kombinovaný	Cookie	Klientský	Kombinovaný
počet premenných	48	492	600	24	450	500
pomer tried	50	20	20	20	10	10
n_estimators	252	180	299	271	290	272
max_depth	9	7	6	10	10	10
num_leaves	15	55	19	30	24	56
learning_rate	0.0251	0.0527	0.0360	0.0162	0.0314	0.0189
feature_fraction	0.618	0.779	0.717	0.754	0.717	0.814
bagging_fraction	0.750	0.921	0.604	0.841	0.842	0.677
λ_1	1.303	1.143	1.810	0.927	1.583	0.271
λ_2	0.566	1.948	1.896	1.550	1.120	0.728

6.2 Kalibrácia pravdepodobností modelov

Výstup modelu LGBM vo forme *predict_proba*[13] nie je možné automaticky interpretovať ako pravdepodobnosť vo forme sklonu ku kúpe produktu. Hoci model produkuje hodnoty v intervale $\langle 0, 1\rangle$, tieto hodnoty nemusia zodpovedať skutočnej pravdepodobnosti výskytu cieľovej udalosti, keďže modely sú trénované napr. minimalizáciou stratovej funkcie (rovnica 13), resp. v prípade SVM maximalizáciou marginu formulovanou ako optimalizačný problém (rovnica 6). Tieto kritériá sú zamerané na optimalizáciu predikčnej presnosti klasifikácie, a nie na priamu kalibráciu pravdepodobnostných odhadov, v dôsledku čoho výstupné skóre modelov nemusí zodpovedať skutočným pravdepodobnostiam výskytu triedy.

Z tohto dôvodu je potrebné tieto výstupy dodatočne kalibrovať. Na kalibráciu sme

[13]Metóda predict_proba vracia pre každé pozorovanie odhad pravdepodobnosti príslušnosti k jednotlivým triedam, pričom tieto pravdepodobnosti sú normalizované tak, že ich súčet je rovný 1. Viac k tejto metóde na [28].

využili jednoduchý prístup založený na logistickej regresii, známy aj ako *Platt scaling*, ktorý bol predstavený v práci [27]. Tento prístup spočíva v transformácii výstupu modelu pomocou sigmoidálnej funkcie, ktorej parametre sú odhadnuté tak, aby výsledné hodnoty čo najlepšie aproximovali skutočné posteriorné pravdepodobnosti.

Formálne je kalibračná funkcia definovaná ako:

$$P(y = 1 \mid x) := \frac{1}{1 + \exp(Af(x) + B)}, \tag{25}$$

kde $f(x)$ predstavuje výstup pôvodného modelu a A, B sú parametre odhadnuté pomocou logistickej regresie.

Platt ukazuje, že výstupy klasifikačných modelov (pôvodne SVM) nepredstavujú priamo pravdepodobnosti, ale hodnoty rozhodovacej funkcie modelu, ktoré nie sú kalibrované. Preto navrhuje použiť sigmoidálnu transformáciu, ktorá mapuje tieto výstupy na pravdepodobnosti minimalizáciou log-loss funkcie.

V našom prípade bol tento prístup aplikovaný na výstupy modelov LGBM pre každý model (cookie, klientský a kombinovaný) a pre každý produkt (úver a investícia) samostatne. Ako vstup do kalibrácie bola použitá predikovaná pravdepodobnosť modelu a závislá premenná. Výsledkom je kalibrovaná pravdepodobnosť, ktorú je možné interpretovať ako odhad skutočnej pravdepodobnosti nákupu produktu.

Kalibráciou sme zabezpečili, že výstupy jednotlivých modelov sú porovnateľné medzi sebou, keďže všetky predstavujú pravdepodobnosti v rovnakom zmysle. Bez kalibrácie by porovnávanie modelov na základe ich výstupov mohlo viesť k rôznym výsledkom lebo modely môžu nadhodnocovať alebo podhodnocovať výsledky.

6.2.1 Výber cieľovej skupiny klientov

Na základe kalibrovaných pravdepodobností sme definovali jednoduché rozhodovacie pravidlo pre výber cieľovej skupiny klientov, ktorým bude zobrazený reklamný banner. Pre každý model a produkt sme najskôr určili priemernú hodnotu kalibrovanej pravdepodobnosti v populácii klientov. Táto hodnota predstavuje referenčnú úroveň, ktorá zodpovedá priemernej pravdepodobnosti nákupu daného produktu.

Následne sme definovali rozhodovacie pravidlo, podľa ktorého sa reklamný banner zobrazí iba tým klientom, ktorých kalibrovaná pravdepodobnosť presahuje túto priemernú

hodnotu. Formálne môžeme toto pravidlo zapísať ako:

$$\hat{p}_i > \overline{p} \;\Rightarrow\; \text{zobraziť banner}_i$$

kde $\hat{p}_i$ predstavuje kalibrovanú pravdepodobnosť pre i-teho klienta a $\overline{p}$ je priemerná pravdepodobnosť v populácii pre daný model a produkt.

Týmto spôsobom vzniká tzv. spodný prah (threshold), pričom horný prah nemáme definovaný. Výber klientov je teda založený výlučne na tom, či ich pravdepodobnosť presahuje priemernú hodnotu, čo vedie k identifikácii klientov s nadpriemernou pravdepodobnosťou nákupu produktu. Tento výber je realizovaný samostatne pre každý model a zároveň pre každý produkt.

Títo klienti majú vyššiu než priemernú pravdepodobnosť realizácie cieľovej udalosti, a teda aj vyšší očakávaný prínos zobrazenia reklamy. Výhodou tohto prístupu je jeho jednoduchosť a dobrá interpretovateľnosť, pričom zároveň umožňuje ponechať flexibilitu pri oslovovaní zvyšnej časti populácie podľa aktuálnych marketingových cieľov banky.

6.2.2 Rozhodovacie pravidlo pre výber produktu

Po identifikácii klientov s nadpriemernou pravdepodobnosťou nákupu jednotlivých produktov je potrebné určiť, aký produkt bude klientovi zobrazený. V tejto práci uvažujeme jeden prístup k tomuto rozhodovaniu.

Výber primárneho a sekundárneho produktu: Zavádzame poradie pre produkty v prípade, že klient spĺňa podmienky pre viacero produktov. V takom prípade sa produkty porovnávajú na základe rozdielu medzi kalibrovanou pravdepodobnosťou a príslušnou prahovou hodnotou.

Na tento účel definujeme skóre pre každý produkt ako:

$$s_i^{(j)} = \hat{p}_i^{(j)} \cdot \left(\hat{p}_i^{(j)} - \overline{p}^{(j)}\right),$$

kde $\hat{p}_i^{(j)}$ predstavuje kalibrovanú pravdepodobnosť nákupu produktu j pre i-teho klienta a $\overline{p}^{(j)}$ je priemerná kalibrovaná pravdepodobnosť v populácii pre daný produkt.

Ako primárny produkt je zvolený ten, pre ktorý je hodnota tohto skóre najvyššia. Formálne, pre dva produkty (úver a investícia), platí:

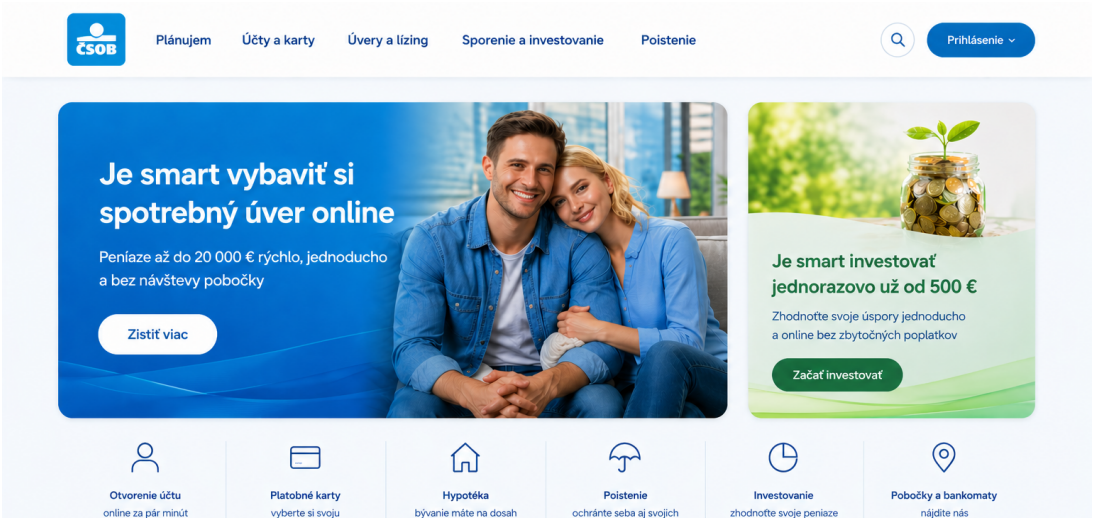
$$s_i^{(uver)} > s_i^{(investicia)} \;\Rightarrow\; \text{primárny produkt je úver},$$

pričom v opačnom prípade je ako primárny zvolený investičný produkt. Druhý produkt je následne označený ako sekundárny, ktorý sa zobrazuje len v prípadoch, keď pozorovanie spĺňa podmienky pre oba produkty súčasne. Z tohto dôvodu sú jeho podiely výrazne nižšie a predstavujú podmnožinu pozorovaní.

Takto definované skóre zohľadňuje nielen absolútnu úroveň pravdepodobnosti nákupu, ale aj relatívne prevýšenie nad priemernou hodnotou v populácii.

6.3 Návrh personalizovaného zobrazovania bannerov

Na obrázkoch 21 a 23 sú znázornené návrhy zobrazovania dvoch reklamných bannerov na webstránke banky, konkrétne pre produkty spotrebného úveru a jednorazovej investície, v ktorých je pre každé pozorovanie určený primárny a sekundárny produkt na základe porovnania definovaného skóre pre oba produkty.



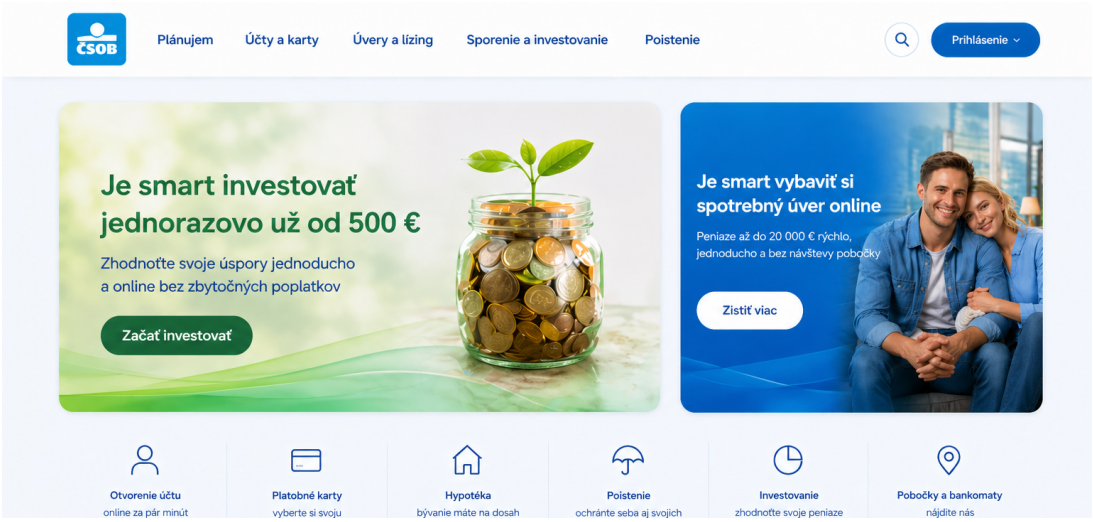
Obr. 21: Ukážka reklamných bannerov (primárny = úver a sekundárny = investícia). Vytvorené pomocou [30].



Obr. 22: Počty percent pozorovaní s nadpriemernou pravdepodobnosťou pre tri modely. Vytvorené pomocou [30].

Pod každým bannerom na obr. 22 a 24 sa nachádza šesť „bubliniek“, ktoré reprezentujú tri modely (cookie, klientský a kombinovaný) pre primárnu a sekundárnu reklamu. Pre každý z týchto modelov je uvedený podiel pozorovaní z testovacej vzorky, pri ktorých model predikoval nadpriemernú kalibrovanú pravdepodobnosť záujmu o daný produkt, a teda by pre tieto pozorovania došlo k zobrazeniu reklamného banneru.

Uvedené percentá sú počítané na úrovni jednotlivých pozorovaní, pričom jedno pozorovanie zodpovedá konkrétnej cookie. Jeden klient môže byť preto v dátach reprezentovaný viacerými pozorovaniami, ktoré môžu mať rozdielne predikované pravdepodobnosti. Z tohto dôvodu percentá nepredstavujú podiel unikátnych klientov, ale podiel pozorovaní, pri ktorých by bol banner zobrazený.



Obr. 23: Ukážka reklamných bannerov (primárny = investícia a sekundárny = úver). Vytvorené pomocou [31].



Obr. 24: Počty percent pozorovaní s nadpriemernou pravdepodobnosťou pre tri modely. Vytvorené pomocou [31].

Pre lepšie pochopenie interpretácie možno uviesť nasledujúci príklad vyplývajúci priamo z obr. 21 a 22. Ak cookie model identifikuje 20,73 % pozorovaní, pre ktoré je primárnym produktom úver, znamená to, že týmto pozorovaniam by bol zobrazený úver ako hlavný

banner. Zároveň 8,03 % pozorovaní predstavuje prípady, kde je popri primárnom úvere zobrazená aj investícia ako sekundárny produkt.

V prípade cookie modelu pozorujeme vyšší podiel pozorovaní nad priemernou pravdepodobnosťou (najmä výrazne pri produkte investícia). Tento jav je spôsobený nižšou predikčnou výkonnosťou modelu, ktorá vedie k tomu, že predikované pravdepodobnosti sú koncentrované v okolí priemeru. Inými slovami, model nedokáže dostatočne rozlišovať medzi pozorovaniami s vysokou a nízkou pravdepodobnosťou realizácie produktu, čo spôsobuje, že veľká časť pozorovaní sa nachádza (jemne) nad rozhodovacou hranicou. Tento jav možno najlepšie vidieť pri produkte investícia, kde je reklama zobrazená takmer 40% pozorovaní. Tento výsledok je v súlade s analýzou v kapitole 5, kde cookie model vykazoval najnižšie hodnoty metrík, a teda aj najslabšiu schopnosť rozlišovať medzi relevantnými a nerelevantnými pozorovaniami.

Naopak, klientský a kombinovaný model dosahujú v kapitole 5 vyššie hodnoty metrík AUC a lift, čo znamená lepšiu schopnosť zoradiť pozorovania podľa pravdepodobnosti výskytu sledovanej udalosti. Model lepšie rozlišuje medzi pozorovaniami a priraďuje im výraznejšie pravdepodobnosti. V dôsledku toho je menší podiel pozorovaní v okolí priemeru, a teda aj menší podiel pozorovaní prekračujúcich priemernú hodnotu pravdepodobnosti.

Súčet podielov pozorovaní podľa modelov, ktorým bol nejaký z produktov zobrazený ako primárny a sekundárny, predstavuje celkový podiel pozorovaní, ktorým bol daný produkt zobrazený. V tom prípade kombinovaný model vedie k mierne vyššiemu podielu zobrazení banneru v porovnaní s klientským modelom pre oba produkty, čo znamená širší zásah reklamy. Tento efekt vyplýva z využitia dodatočných informácií z cookie dát, ktoré v tomto prípade umožňujú identifikovať väčší počet pozorovaní s nadpriemernou pravdepodobnosťou. Nejde teda o jednoznačne lepší alebo horší výsledok, ale o odlišnú stratégiu cielenia, pričom konečné rozhodnutie závisí od preferencií banky medzi širším zásahom a prísnejším výberom.

Rozdiel medzi klientským a kombinovaným modelom možno bližšie interpretovať aj z pohľadu stability rozhodnutí. Napríklad pri produkte úver klientský model identifikuje 16,16 % pozorovaní na zobrazenie banneru, pričom pri prechode na kombinovaný model zostáva 88,46 % z týchto pozorovaní nezmenených a len 11,54 % z nich už banner nezobrazí. Zároveň z pozorovaní, ktorým klientský model úver nezobrazil (83,84 %), zostáva

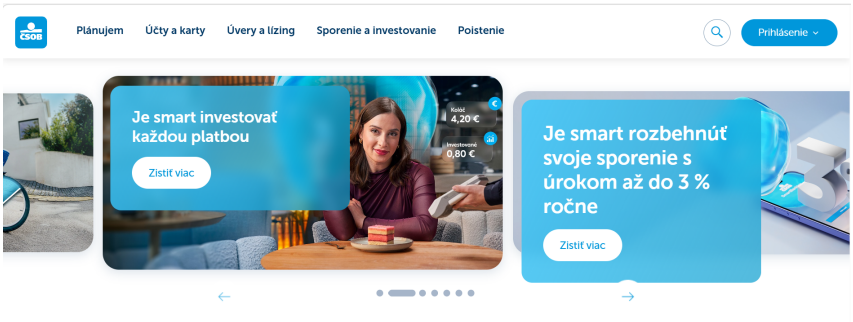
97,72 % rozhodnutí nezmenených a iba 2,28 % pozorovaní je v kombinovanom modeli novozaradených na zobrazenie. Podobný vzťah možno pozorovať aj pri produkte investícia.

Tieto výsledky naznačujú, že kombinovaný model vo veľkej miere zachováva rozhodnutia klientského modelu, pričom dochádza k miernej zmene medzi zobrazenými a nezobrazenými prípadmi. Model tak niektoré pozorovania dodatočne zahŕňa, iné naopak vylučuje, pričom celková štruktúra výberu zostáva veľmi podobná.

Uvedené výsledky zobrazovania bannerov podľa modelov pre produkty poukazujú na dôležitý rozdiel medzi kvalitou modelu a jeho praktickým dopadom pri aplikácii. Model s horšou schopnosťou rozlišovať medzi pozorovaniami môže viesť k vyššiemu počtu zobrazení banneru, čo však neznamená lepšie zacielenie reklamy. Naopak, modely s lepšou predikčnou výkonnosťou zobrazujú reklamu menšiemu, presnejšie vybranému počtu klientov, čo je z pohľadu marketingu vhodnejšie.

Zobrazenie bannerov pre klientov s pravdepodobnosťou pod prahovou hodnotou: V uvedenom prístupe 6.3 platí, že klientom, ktorých kalibrovaná pravdepodobnosť nepresahuje príslušnú prahovú hodnotu, nie je zobrazený personalizovaný reklamný banner. Namiesto toho sa im zobrazuje štandardná (default) stránka, ktorá obsahuje všeobecné reklamné prvky.

Na obrázku 25 je znázornená štandardná (default) verzia webovej stránky, ktorá sa zobrazuje klientom, pre ktorých žiadny z modelov neidentifikoval nadpriemernú pravdepodobnosť pre zobrazenie personalizovaného reklamného banneru.



Obr. 25: Ukážka štandardnej (default) verzie webovej stránky (`https://www.csob.sk/`) zobrazovanej klientom bez nadpriemernej pravdepodobnosti. Zobrazené 18.4.2026.

Záver

V práci sme ukázali, ako možno pomocou webových dát získaných prostredníctvom cookies modelovať preferencie klientov a na základe ich kombinácie s internými klientskými dátami banky predikovať pravdepodobnosť realizácie vybraného bankového produktu. Zostrojili sme tri dátové vzorky, na ktorých sme natrénovali modely strojového učenia a pre ich vzájomné porovnanie sme pripravili testovaciu vzorku, ktorá zodpovedá skutočnej distribúcii pozorovaní.

V práci sme skonštruovali tri dátové vzorky: vzorku založenú na cookie dátach, vzorku založenú na klientských dátach a kombinovanú vzorku. Ich podoba je ilustrovaná na príkladoch v tabuľkách 6, 7 a 8. Na týchto vzorkách sme následne aplikovali päť metód strojového učenia. Pre potreby práce sme sa venovali vytváraniu čŕt vhodných pre charakterizovanie vplyvu cookies. Pri konštrukcii dátových vzoriek sme realizovali výber premenných na základe vzájomnej informácie (tabuľka 9), čím sme dosiahli redukciu dimenzie a zlepšenie interpretovateľnosti. Dátovú vzorku zloženú z kombinácie cookie a klientských dát sme ohraničili na 600 premenných, pričom sme zoradili premenné podľa veľkosti vplyvu na preferencie klienta na nákup vybraných produktov. Na základe tejto hodnoty vzájomnej informácie boli následne vybrané premenné aj pre cookie a klientský model.

Kvantifikácia výkonnosti modelov ukázala, že prínos kombinácie dát nie je jednoznačný naprieč všetkými klasifikačnými metódami. Zlepšenie predikčnej výkonnosti kombinovaného modelu bolo pozorované najmä pri metóde LGBM, zatiaľ čo pri ostatných prístupoch nebol prínos kombinácie dát jednoznačný a v niektorých prípadoch bol kombinovaný model prekonaný modelom založeným výlučne na klientskych dátach.

V práci sme ďalej zovšeobecnili metriky hodnotenia modelov pre prípad viacnásobných pozorovaní jedného klienta a ich aplikáciou sme ukázali na obr. 19, že rozdiel medzi váženými a neváženými metrikami je minimálny. Tento výsledok naznačuje, že váženie pozorovaní nemá v danom kontexte zásadný vplyv na hodnotenie modelov.

Pomocou SHAP hodnôt sme identifikovali relevantné premenné a kvantifikovali ich vplyv na predikciu (obrázky 17 a 18). Cookie dáta prispievali ku predikcii v menšej miere ako ostatné dátové súbory, avšak ich prínos nebol nulový a bol pozorovateľný najmä v boostingových metódach. Analýza ukázala, ktoré typy premenných majú najväčší prínos pre jednotlivé modely a umožnila detailnejšie porozumenie rozdielov medzi nimi. Medzi

najvýznamnejšie cookie premenné patrili najmä premenné typu recency a referrer, ktorých vplyv je podrobnejšie analyzovaný v časti 5.3.1.

Súčasťou práce bol aj návrh praktického využitia modelov pri personalizácii obsahu na webovej stránke banky. Navrhli sme rozdelenie klientov do skupín na základe kalibrovanej predikovanej pravdepodobnosti (25) a ich priradenie k rôznym variantom zobrazovaného obsahu (obrázky 21 až 24), pričom bannery boli zobrazované iba pozorovaniam s nadpriemernou kalibrovanou pravdepodobnosťou. Dôležitou súčasťou bolo aj porovnanie podielu pozorovaní, ktorým bol banner zobrazený, naprieč modelmi. Kvantifikácia podielu pozorovaní so zobrazeným bannerom ukázala, že cookie model viedol k širšiemu, avšak menej presnému zásahu, zatiaľ čo klientský a kombinovaný model umožňoval selektívnejší výber klientov, pričom kombinovaný model mierne rozširuje zásah oproti klientskému.

Naše výsledky poukazujú na potenciál kombinácie webových a klientských dát primárne v boostingových metódach, čo je v súlade so zisteniami v [3]. Zároveň sa ukázalo, že použitie vážených metrík nevedie k významnej zmene hodnotenia modelov, čo naznačuje ich obmedzený praktický prínos v danom kontexte.

Ako možné smerovanie ďalšieho výskumu sa javí modelovanie správania klientov na úrovni jednotlivých návštev. Tento prístup by umožnil detailnejšie zachytiť ich interakcie a rozhodovacie procesy pri práci s webom. Ďalšou možnosťou je rozšírenie modelu o časový rozmer, ktorý by umožnil zohľadniť vývoj správania klientov v čase a identifikovať vzory. V neposlednom rade sa ponúka implementácia navrhnutého riešenia v reálnom prostredí banky, ktorá by umožnila overiť jeho praktickú využiteľnosť prostredníctvom experimentálneho nasadenia.

Zoznam použitej literatúry

[1] Madhuri, A., et al. Exploring the role of personalization in e-commerce: Impacts on consumer trust and purchase intentions. Eur. Econ. Lett, 2024, 14.3: 907-919. Dostupné na internete (25.4.2026): https://www.researchgate.net/publication/383603236_Exploring_the_Role_of_Personalization_in_E-commerce_Impacts_on_Consumer_Trust_and_Purchase_Intentions

[2] Chaudhuri, Neha, et al. On the platform but will they buy? Predicting customers' purchase behavior using deep learning. Decision Support Systems, 2021, 149: 113622. Dostupné na internete (25.4.2026): https://www.sciencedirect.com/science/article/abs/pii/S0167923621001329

[3] Kim, Seongbeom; Shin, Woosik; Kim, Hee-Woong. Predicting online customer purchase: The integration of customer characteristics and browsing patterns. Decision Support Systems, 2024, 177: 114105. Dostupné na internete (30.4.2026): https://www.sciencedirect.com/science/article/abs/pii/S016792362300180X

[4] Englehardt, Steven; Narayanan, Arvind. Online tracking: A 1-million-site measurement and analysis. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016. p. 1388-1401. Dostupné na internete (11.09.2025): https://dl.acm.org/doi/abs/10.1145/2976749.2978313

[5] Cahn, Aaron, et al. An empirical study of web cookies. In: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 891–90. Dostupné na internete (16.11.2025): https://doi.org/10.1145/2872427.2882991

[6] Roesner, Franziska; Kohno, Tadayoshi; Wetherall, David. Detecting and defending against Third-Party tracking on the web. In: 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12). 2012. p. 155-168. Dostupné na internete (11.09.2025): https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/roesner

[7] Bult, Jan Roelf; Wansbeek, Tom. Optimal selection for direct mail. Marketing Science, 1995, 14.4: 378-394. Dostupné na internete (05.09.2025): https://www.researchgate.net/publication/227442110_Optimal_Selection_for_Direct_Mail

[8] Zaharia, Matei; Xin, Reynold S.; Wendell, Patrick; Das, Tathagata; Armbrust, Michael; Dave, Ankur; et al. Apache Spark: a unified engine for big data processing. Communications of the ACM, 2016, 59.11: 56–65. Dostupné na internete (05.09.2025): https://doi.org/10.1145/2934664

[9] Vink, Ritchie; et al. Polars: Blazingly fast DataFrames in Rust and Python. GitHub repository, 2025. Dostupné na internete (05.09.2025): https://github.com/pola-rs/polars

[10] Koshova, Olha: Modelovanie kauzálnych vplyvov marketingových oslovení na preferencie klienta, diplomová práca, FMFI UK, Bratislava 2025. Dostupné na internete (27.12.2025): https://opac.crzp.sk/?fn=detailBiblioForm&sid=C9A1C75A261DD259FF86441E26A3

[11] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning, 2009.

[12] Gareth, James, et al. An introduction to statistical learning: With applications in python, 2023. Dostupné na internete (5.1.2026): https://link.springer.com/book/10.1007/978-3-031-38747-0

[13] Erickson, Bradley J., Felipe Kitamura. Magician's corner: 9. Performance metrics for machine learning models. Radiology: Artificial Intelligence 3.3 (2021): e200126. Dostupné na internete (7.1.2026): https://pubs.rsna.org/doi/full/10.1148/ryai.2021200126

[14] Stanley, Richard P. An introduction to hyperplane arrangements. Geometric combinatorics 13 (2007): 389-496.

[15] [Documentation for Bloomreach], the Agentic Platform for Personalization. Dostupné na internete (27.1.2026): https://documentation.bloomreach.com/

[16] [The Secret Sauce Behind Random Forests]. Dostupné na internete (7.2.2026): https://medium.com/@pacosun/the-secret-sauce-behind-random-forests-c5d77d850d3c

[17] [UI Informačné systémy], (2026). Online. In: ChatGPT. Verzia 5.2. Dostupné na OpenAI, https://chatgpt.com/s/m_698866a4ef3c8191badeb9d3a1b5061e. [zobrazené 8.2.2026]. Zadanie: „Vytvorenie obrázku pre tri AUC krivky, pre tri modely“

[18] Fawcett, Tom. An introduction to ROC analysis. Pattern recognition letters, 2006, 27.8: 861-874. Dostupné na internete (8.2.2026): https://www.sciencedirect.com/science/article/abs/pii/S016786550500303X

[19] Chen, Tianqi. XGBoost: A Scalable Tree Boosting System. Cornell University, 2016. Dostupné na internete (8.2.2026): https://arxiv.org/abs/1603.02754

[20] Ke, Guolin, et al. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 2017, 30. Dostupné na internete (28.2.2026): https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[21] Létoffé, Olivier; Huang, Xuanxiang; Marques-Silva, Joao. Towards trustable SHAP scores. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2025. p. 18198-18208. Dostupné na internete (2.3.2026): https://ojs.aaai.org/index.php/AAAI/article/view/34002

[22] Aas, Kjersti; Jullum, Martin; Løland, Anders. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. Artificial Intelligence, 2021, 298: 103502. Dostupné na internete (2.3.2026): https://www.sciencedirect.com/science/article/pii/S0004370221000539

[23] Lundberg, Scott M.; Lee, Su-In. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems (NeurIPS), 2017. Dostupné na internete (19.3.2026): http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[24] Ng, Andrew Y. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In: Proceedings of the twenty-first international conference on Machine learning. 2004. p. 78. Dostupné na internete (2.3.2026): https://dl.acm.org/doi/10.1145/1015330.1015435

[25] Akiba, Takuya, et al. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019. p. 2623-2631. Dostupné na internete (3.3.2026): https://dl.acm.org/doi/abs/10.1145/3292500.3330701

[26] [Optuna]. Hyperparameter Optimization Framework [online]. Dostupné na internete (3.3.2026): https://optuna.org/

[27] Platt, John, et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers, 1999, 10.3: 61-74.

[28] Pedregosa, Fabian, et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 2011, 12: 2825-2830.

[29] Maron, Oded; Lozano-Pérez, Tomás. A framework for multiple-instance learning. Advances in neural information processing systems, 1997, 10. Dostupné na internete (19.4.2026): https://people.csail.mit.edu/tlp/publications/maron98framework.pdf

[30] [UI Informačné systémy], (2026). Online. In: ChatGPT. Verzia 5.2. Dostupné na OpenAI, https://chatgpt.com/s/m_69f24eb3959081919d53d15f2e032350. [zobrazené 29.4.2026]. Zadanie: „Vytvorenie obrázku reklamných bannerov (primárny a sekundárny) pre produkty a pre modely“

[31] [UI Informačné systémy], (2026). Online. In: ChatGPT. Verzia 5.2. Dostupné na OpenAI, https://chatgpt.com/s/m_69f24ee1babc8191b2444d6d87e8f051. [zobrazené 29.4.2026]. Zadanie: „Vytvorenie obrázku reklamných bannerov (primárny a sekundárny) pre produkty a pre modely“

[32] Abdiansah, Abdiansah; Wardoyo, Retantyo. Time complexity analysis of support vector machines (SVM) in LibSVM. Int. J. Comput. Appl, 2015, 128.3: 28-34.

Príloha A

Definícia hyperroviny

Hyperrovina v priestore $\mathbb{R}^p$ je $(p-1)$-rozmerný podpriestor, ktorý možno definovať podľa [14] ako množinu bodov

$$H = \{x \in \mathbb{R}^p : \beta_0 + \beta^\top x = 0\},$$

kde $\beta \in \mathbb{R}^p$ je vektor normály určujúci smer hyperroviny a $\beta_0 \in \mathbb{R}$ určuje, ako je hyperrovina posunutá. Geometricky hyperrovina delí priestor na dve časti a v kontexte klasifikácie slúži ako rozhodovacia hranica medzi dvoma triedami.

Sign funkcia

Sign funkcia je definovaná pre $x \in \mathbb{R}$ a platí:

$$\operatorname{sign}(x) = \begin{cases} 1, & \text{ak } x \geq 0, \\ -1, & \text{ak } x < 0. \end{cases}$$

Logistická regresia - typy regularizácie

V praktickej časti práce pri trénovaní logistickej regresie uvažujeme mieru regularizácie ako hyperparametre modelu. Konkrétne optimalizujeme uvažujeme typ regularizácie a hodnotu regularizačného parametra λ ako hyperparametre modelu.

Podľa [24], pri odhade parametrov logistickej regresie maximalizujeme penalizovanú log-vierohodnosť

$$\ell(\boldsymbol{\beta}) := \sum_{i=1}^{m} \log p(y^{(i)} \mid x^{(i)}; \boldsymbol{\beta}) - \lambda R(\boldsymbol{\beta}),$$

kde $R(\boldsymbol{\beta})$ je regularizačný člen a $\lambda \geq 0$ je regularizačný parameter.

L1 regularizácia:

Pri L1 regularizácii je penalizačný člen definovaný ako

$$R(\boldsymbol{\beta}) := \|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|.$$

Optimalizačný problém má teda tvar

$$\max_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{m} \log p(y^{(i)} \mid x^{(i)}; \boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

L1 penalizácia podporuje riedke riešenia, t. j. niektoré koeficienty β_j sú presne nulové. To znamená, že model implicitne vykonáva výber premenných. L1 regularizovaná logistická regresia je teda vhodná najmä v problémoch s vysokou dimenziou vstupu.
L2 regularizácia:

Pri L2 regularizácii je penalizačný člen daný ako

$$R(\boldsymbol{\beta}) := \|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p} \beta_j^2.$$

Optimalizačný problém má tvar

$$\max_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{m} \log p(y^{(i)} \mid x^{(i)}; \boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

L2 penalizácia vedie k zmenšovaniu koeficientov smerom k nule, avšak zvyčajne ich nenuluje. L2 regularizácia nevykonáva implicitný výber premenných a v prostredí s veľkým počtom nerelevantných premenných môže vyžadovať väčší počet trénovacích pozorovaní než L1 regularizácia.

Porovnanie knižníc PySpark a Polars pri spracovaní dát

Pri spracovaní a transformácii veľkého objemu dát z mobilnej aplikácie banky sme kód implementovali v dvoch rôznych knižniciach – PySpark a Polars. Obidve knižnice umožňujú efektívnu prácu s veľkými dátovými súbormi, avšak líšia sa svojím prístupom, výkonnosťou a využitím systémových zdrojov.

PySpark

PySpark je rozhranie pre Apache Spark, ktoré umožňuje spracovávať veľmi veľké objemy dát, často rozdelené na viacerých počítačoch alebo serveroch. V našom prípade sme PySpark využili na spracovanie mesačných dát s približne 4,5 milióna riadkov.

Výhodou PySparku je, že spracováva dáta postupne a efektívne využíva diskové úložisko, takže nepotrebuje veľa pamäte lokálneho Pythonu. V našej implementácii však výpočty bežali na serveroch klastra, nie priamo v RAM lokálneho počítača. Nevýhodou je, že spracovanie trvá dlhšie, pretože operácie sa musia najskôr pripraviť a rozplánovať, čo zvyšuje čas odozvy.

Polars

Polars je moderná knižnica, ktorá je navrhnutá tak, aby spracovávala dáta veľmi rýchlo, najmä ak sa zmestia do pamäte počítača. Vďaka efektívnej optimalizácii dokáže Polars vykonávať rôzne transformácie a výpočty výrazne rýchlejšie ako PySpark.

Naopak, Polars spotrebuje viac pamäte a pri veľmi veľkých dátach, ako sú tie naše, môže nastať problém, že sa nezmestia do pamäte, čo spôsobí zníženie výkonu alebo chyby.

Porovnanie v našej práci

Ukážka 2: Pseudokód: PySpark vs Polars

```
spark_data = load_data_pyspark()
unique_users_spark = pyspark_count_unique(
    spark_data, "customer", by="month")
pl_data = load_data_polars()
unique_users_polars = polars_count_unique(
    pl_data, "customer", by="month")
```

Výsledky výpočtu sú zhrnuté v tabuľke nižšie:

Tabuľka 14: Porovnanie PySpark a Polars pre počet unikátnych používateľov za mesiac.

Pseudokód	Čas [s]	RAM [GB]	Unikátni používatelia
PySpark	20	2	200 000
Polars	4	10	200 000

Výsledky ukázali, že spracovanie v Polars bolo výrazne rýchlejšie, čo výrazne urýchlilo spúšťanie a testovanie funkcií.

Na druhej strane, Polars mal problémy s veľkosťou dát, preto sme museli kvôli obmedzeniam pamäte vybrať iba niektoré stĺpce zo vstupného dátoveho súboru. Naproti tomu PySpark zvládal spracovanie celých dát bez nutnosti ich redukcie, avšak spracovanie trvalo podstatne dlhšie.

Z týchto dôvodov sme sa v ďalšej práci rozhodli používať Polars, pretože jeho rýchlosť a efektivita pri práci s vybranými dátami prevažovali nad nevýhodou obmedzenej kapacity pamäte.

Výber hyperparametrov pomocu knižnice Optuna

Hyperparametre sú hodnoty, ktoré sa nastavujú pred začiatkom trénovania modelu a ovplyvňujú jeho správanie (napríklad regularizačný parameter, hĺbka stromu alebo počet stromov). Cieľom optimalizácie je nájsť takú kombináciu hyperparametrov, pri ktorej model dosahuje najlepšie výsledky podľa zvoleného hodnotiaceho kritéria.

Optuna je voľne dostupný program určený na optimalizáciu hyperparametrov modelov strojového učenia. Podľa [25] je v Optune optimalizácia formulovaná ako proces minimalizácie alebo maximalizácie tzv. účelovej funkcie. Táto funkcia pre danú kombináciu hyperparametrov natrénuje model a vráti hodnotu hodnotiacej metriky (napríklad ROC- AUC). Každé jedno vyhodnotenie účelovej funkcie sa nazýva pokus (trial) a celý proces optimalizácie sa nazýva štúdia (study).

Kľúčovou vlastnosťou Optuny je princíp nazývaný define-by-run. To znamená, že sa vyhľadávací priestor hyperparametrov definuje vopred, avšak konkrétne kombinácie hyperparametrov, ktoré sa testujú počas optimalizácie, sa určujú dynamicky počas behu programu. Hyperparametre sa generujú pomocou špeciálnych metód, ktoré vychádzajú z hodnôt získaných vyhľadávacím algoritmom z predchádzajúcich hodnotení účelovej funkcie. Každý testovaný vektor hyperparametrov umožňuje jednoducho vyjadriť aj zložité alebo podmienené vyhľadávacie priestory.

Optuna využíva efektívne vzorkovacie algoritmy (angl. sampling algorithms), ktoré rozhodujú, aké hodnoty hyperparametrov sa budú skúšať. Podporuje napríklad metódu TPE (Tree-structured Parzen Estimator) alebo CMA-ES (Covariance Matrix Adaptation Evolution Strategy). Tieto algoritmy využívajú informácie z predchádzajúcich pokusov na smerovanie vyhľadávania do perspektívnych oblastí priestoru hyperparametrov.

Dôležitou súčasťou systému je aj mechanizmus nazývaný pruning, teda predčasné ukončovanie neperspektívnych pokusov. Počas trénovania sa priebežne sledujú čiastkové výsledky a ak sa ukáže, že daná kombinácia hyperparametrov pravdepodobne nepovedie k dobrému výsledku, výpočet sa zastaví ešte pred jeho dokončením. Optuna implementuje najmä algoritmus Asynchronous Successive Halving (ASHA), ktorý je vhodný aj pre paralelné a distribuované výpočty. Viac informácií o použitých algoritmoch vzorkovania a pruningu je možné nájsť v pôvodnom článku [25] alebo na stránke projektu [26].

Search space (priestor hyperparametrov) predstavuje množinu hodnôt, z ktorých sú vy-

berané jednotlivé hyperparametre modelu počas procesu optimalizácie. Inými slovami ide o spojité (intervaly reálnych čisel) alebo diskrétne (konečné) množiny hodnôt, v ktorých sa hľadá optimálna kombinácia hyperparametrov. Pre každý hyperparameter je definovaný samostatný rozsah hodnôt

V našej práci okrem špecifických hyperparametrov pre jednotlivé klasifikačné modely optimalizujeme aj počet premenných a počet vzoriek zahrnutých do trénovania modelu.

Počet vzoriek je určený percentuálnym zastúpením pozitívnych prípadov v dátových súboroch. Optimalizujú sa hodnoty 10 %, 20 %, 35 % a 50 %. Cieľom je analyzovať vplyv postupného pridávania negatívnych prípadov (pri fixnom počte pozitívnych) na výkonnosť modelu. Pri každom nastavení je zahrnutý maximálny počet pozitívnych prípadov, pričom sa postupne pridávajú negatívne príklady. Napríklad, ak 50 % predstavuje 10 000 pozitívnych a 10 000 negatívnych vzoriek, potom nastavenie 35 % obsahuje rovnakých 10 000 pozitívnych a približne 18 000 negatívnych vzoriek (pribudlo 8 000 negatívnych oproti 50 %).

Počet premenných je prispôsobený typu dátového súboru, konkrétnemu produktu a zároveň aj veľkosti dátového súboru, ktorá je daná podielom pozitívnych prípadov. Výber premenných vychádza z ich zoradenia podľa hodnoty vzájomnej informácie, ako je popísané v časti 3.5. Pre každý dátový súbor je najskôr určený maximálny počet premenných. Tento maximálny počet následne slúži ako horná hranica pre výber premenných v danom modeli. Konkrétne hodnoty maximálnych počtov premenných pre jednotlivé kombinácie modelu, produktu a veľkosti dátového súboru sú uvedené v tabuľkách (15 a 16). Počet premenných je uvažovaný ako hyperparameter vo forme šiestich rovnomerne rozložených hodnôt od nízkeho počtu až po maximálny počet premenných pre daný dátový súbor. Z týchto hodnôt vyberá algoritmus Optuna optimálny počet premenných.

Tabuľka 15: Maximálne počty premenných podľa modelu, podielu pozitívnych vzoriek a produktu (spotrebný úver)

Podiel pozitívnych / Model	50%	35%	20%	10%
Cookie	48	36	42	42
Klientský	486	486	492	486
Kombinovaný	600	600	600	600

Tabuľka 16: Maximálne počty premenných podľa modelu, podielu pozitívnych vzoriek a produktu (jednorázová investícia)

Model \ Podiel pozitívnych	50%	35%	20%	10%
Cookie	36	36	24	24
Klientský	480	474	480	450
Kombinovaný	600	600	600	600

Tabuľka 17: Optimalizované hyperparametre pre jednotlivé klasifikačné modely a ich vyhľadávací priestor.

Model	**Hyperparameter**	**Search space**	**Popis**
LGBM	n_estimators	DI$[100, 300; 1]$	Počet stromov v modeli
	max_depth	DI$[-1, 10; 1]$	Maximálna hĺbka stromu
	num_leaves	DI$[15, 60; 1]$	Maximálny počet listov v každom strome
	learning_rate	CI$_{\log}[0.01, 0.3]$	Rýchlosť učenia
	feature_fraction	CI$[0.6, 1.0]$	Pomer náhodne vybraných atribútov pre každý strom
	bagging_fraction	CI$[0.6, 1.0]$	Pomer náhodne vybraných vzoriek pre každý strom
	lambda_l1	CI$[0.0, 2.0]$	L1 regularizácia
	lambda_l2	CI$[0.0, 2.0]$	L2 regularizácia
RF	n_estimators	DI$[100, 300; 1]$	Počet stromov
	max_depth	DI$[3, 15; 1]$	Maximálna hĺbka stromu
	min_samples_leaf	DI$[2, 20; 1]$	Minimálny počet vzoriek v liste stromu
	min_samples_split	DI$[5, 50; 1]$	Minimálny počet vzoriek pre rozdelenie uzla
	max_features	CAT{`sqrt`, `log2`}	Počet atribútov vybraných pri každom rozdelení

Model	Hyperparameter	Search space	Popis
XGB	n_estimators	DI$[100, 500; 1]$	Počet stromov
	max_depth	DI$[3, 10; 1]$	Maximálna hĺbka stromu
	learning_rate	$\text{CI}_{\log}[0.01, 0.3]$	Rýchlosť učenia
	subsample	CI$[0.6, 1.0]$	Pomer náhodne vybraných vzoriek pre každý strom
	colsample_bytree	CI$[0.6, 1.0]$	Pomer atribútov použitých pri stavbe stromu
	gamma	CI$[0.0, 5.0]$	Minimálna redukcia chyby pre rozdelenie uzla
	reg_alpha	CI$[0.0, 5.0]$	L1 regularizácia
	reg_lambda	CI$[0.0, 5.0]$	L2 regularizácia
LR	penalty	CAT$\{$`None`, `l1`, `l2`$\}$	Typ penalizácie
	solver	CAT$\{$`liblinear`, `lbfgs`$\}$	Typ solvera
	C	$\text{CI}_{\log}[0.1, 10.0]$	Intenzita regularizácie
SVM	C	$\text{CI}_{\log}[0.01, 10.0]$	Intenzita regularizácie
	max_iter	DI$[100, 1000; 100]$	Maximálny počet iterácií
	penalty	CAT$\{$`l1`, `l2`$\}$	Typ regularizácie
	tol	$\text{CI}_{\log}[10^{-4}, 10^{-2}]$	Tolerancia pre konvergenciu optimalizačného algoritmu

Poznámka: DI$[a, b; s]$ označuje diskrétny interval od a do b s krokom s; CI$[a, b]$ označuje spojitý interval, z ktorého sú hodnoty vzorkované rovnomerne; $\text{CI}_{\log}[a, b]$ označuje spojitý interval, z ktorého sú hodnoty vzorkované v logaritmickej mierke (t. j. rovnomerne v logaritme parametra, čo vedie k hustejšiemu vzorkovaniu menších hodnôt); CAT$\{\dots\}$ označuje konečnú množinu diskrétnych (kategorických) hodnôt.

Krosvalidácia

Pre hodnotenie generalizácie modelov klasifikácie používame metódu krosvalidácie (cross-validation, CV). Krosvalidácia je štatistická technika, ktorá umožňuje odhadnúť, ako dobre sa naučený model bude správať na neznámych dátach. Jej cieľom je podľa [11,

kap. 7] aproximovať očakávanú hodnotu stratovej funkcie:

$$\mathrm{Err} := \mathbb{E}\Big[L(Y, f(x))\Big],$$

kde $L(y, f(x))$ je stratová funkcia merajúca, ako dobre predikcia modelu $f(x)$ zodpovedá skutočnej hodnote y. Tento vzorec reprezentuje priemernú generalizačnú chybu modelu na nezávislých pozorovaniach z rovnakého rozdelenia ako trénovacie dáta.

V klasickom K-fold CV sa trénovací dátový súbor rozdelí na K častí (foldov). Model sa postupne trénuje na $K-1$ častiach a validuje na zvyšnej časti, pričom sa tento proces opakuje pre všetky foldy. Výsledkom je priemerná hodnota vybranej metriky (napr. ROC-AUC), ktorá poskytuje stabilnejší odhad schopnosti modelu generalizovať.

Na rozdiel od klasického rozdelenia dát na trénovaciu, validačnú a testovaciu množinu, pri K-fold krosvalidácii validačná množina nevzniká ako fixná podmnožina dát, ale dynamicky sa mení v jednotlivých iteráciách. Tým sa efektívne využije celý dátový súbor na trénovanie aj validáciu.

Testovacia množina sa v procese krosvalidácie nepoužíva. Slúži výlučne na finálne, nezávislé zhodnotenie modelu po výbere jeho hyperparametrov a natrénovaní na celých trénovacích dátach.

Príloha B

Podrobný zoznam premenných:

Cookie dáta z bankovej webstránky

event_internal_customer_id	Interný identifikátor klienta
event_type	Typ udalosti
event_ingest_timestamp	Čas prijatia udalosti
event_timestamp	Čas vzniku udalosti
path	URL navštívenej stránky
referrer	Zdroj návštevy
cookie	Identifikátor cookie
os	Operačný systém
browser	Webový prehliadač
location	Geografická lokalita
device	Typ zariadenia

attr_web_tracking

cookie	Identifikátor cookie
customerid	Identifikátor klienta
month_id	Identifikátor mesiaca
c_visit_m	Počet navštívených stránok
c_referrer_m	Počet unikátnych zdrojov
i_onboarding_page_m	Onboarding stránka (1/0)
i_payment_page_m	Payment stránka (1/0)
c_duration_low_limit_m	Počet návštev kratších ako 30 sekúnd
avg_visit_duration_m	Priemerné trvanie návštevy
c_duration_up_limit_m	Počet návštev dlhších ako 2 hodiny
morning_to_all_m	Podiel aktivity ráno (6–12)
afternoon_to_all_m	Podiel aktivity popoludní (12–18)
evening_to_all_m	Podiel aktivity večer (18–24)
night_to_all_m	Podiel aktivity v noci (0–6)
ratio_weekend_to_all_m	Podiel aktivity cez víkend
c_search_engine_m	Počet prístupov zo search engine
c_media_m	Počet prístupov z médií
c_mail_m	Počet prístupov z emailu
c_sb_m	Počet prístupov zo smart bankingu v appke
c_tablet_m	Počet prístupov z tabletu
c_social_m	Počet prístupov zo sociálnych sietí
c_push_m	Počet prístupov cez push notifikácie
avg_visited_pages_m	Priemerný počet stránok na session
i_device_android_m	Primárne Android zariadenie
i_device_apple_m	Primárne Apple zariadenie
i_device_other_m	Primárne iné zariadenie
change_visit_locations	Zmena počtu navštívených stránok oproti minulému mesiacu
change_avg_visit_pages	Zmena priemerného počtu stránok oproti minulému mesiacu
i_change_device_m	Zmena primárneho zariadenia
c_insurance_m	Počet návštev stránok o poistení
c_bank_account_m	Počet návštev stránok o bankovom účte
c_information_about_m	Počet návštev informačných stránok
c_other_services_m	Počet návštev stránok ostatných služieb
c_pearsonal_account_m	Počet návštev stránok osobného účtu
i_dark_mode_m	Primárne používanie dark mode
recency	Počet dní od poslednej návštevy

Poznámka: Všetky premenné z attr_web_tracking s príponou "_m"majú aj kvartálne a ročné agregácie. Výnimkou sú change_visit_locations, change_avg_visit_pages, i_change_device_m, i_device_android_m, i_device_apple_m a i_device_other_m. Predpona c_ značí počet, i_ značí hodnoty 0 a 1, avg_ značí priemer.