Objavovanie znalostí

Martin Macej, Jozef Varga

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislave

1 Analýza datasetu

1.1 Opis problému (podiel práce: 50% / 50%)

Pri založení spoločnosti si častokrát jej zakladatelia kladú otázku, akú cenu výrobkov majú stanoviť, aby boli konkurencie schopný a zároveň aby dosiahli čo najvyšší zisk. Je teda potrebné zaradiť produkty do správnej cenovej kategórie.

Správne riešenie tohto problému dokáže následne dopomôcť firme so vstupom na trh a stať sa tak konkurencieschopnou už existujúcim a zabehnutým firmám.
My sme sa v tomto projekte zaoberali zaradením mobilných telefónov do rôznych cenových kategórii. Toto zaradzovanie sme sa rozhodli riešiť takým spôsobom, že na základe hardvérovej špecifikácie zariadenia sme zaradzovali mobilný telefón do jednej zo 4 cenových kategórií. Riešili sme teda problém klasifikácie, kde na základe vstupných parametrov sme zaradzovali zaradenie objektu do 4 cenových tried.

Na zadanie sme si vybrali dataset [1] z domény, ktorá nám je srdcu blízka. Keďže sme obaja fanúšikovia IT a nových technológií, rozhodli sme pre doménu IT a to konkrétne dataset obsahujúci údaje o hardvérovej konfigurácii 2000 mobilných zariadení.

Pri tomto datasete sme sa podrobnejšie zamerali na rôzne vzťahy medzi cenou mobilného zariadenia a použitým hardvérom, ktorý obsahuje dané zariadenie. Jednou zo zaujímavostí bolo napríklad to, že najväčší vplyv na cenu mobilného zariadenia má pamäť RAM a je v podstate jediný parameter, ktorý zásadnou mierou vplýva na cenu. Samozrejme, cenu ovplyvňujú aj iné parametre ako to, či má alebo nemá dotykovú obrazovku, dĺžka hovoru a podobne, ale oveľa nižšou mierou ako už spomínaná pamäť RAM. Toto zistenie nás veľmi prekvapilo, čakali sme že práve tento parameter bude mať jeden z najvyšších vplyvov na cenu, ale nečakali sme, že až takto zásadne ovplyvňuje cenu.

[1] Mobile Price Classification, https://www.kaggle.com/iabhishekofficial/mobile-price-classification#train.csv

1.2 Opis dát (podiel práce: 50% / 50%)

Vybraný dataset obsahuje 2000 riadkov a 21 stĺpcov vo formáte csv. Každý riadok predstavuje jedno mobilné zariadenia a každý stĺpec predstavuje 1 parameter konkrétneho zariadenia.
Jednou z výhod datasetu je, že všetky hodnoty sú číselné a všetky záznamy sú kompletné, takže sme nemali žiadne problémy s dopočítavaním prázdnych hodnôt a podobne. Keďže s originálnymi dátami sme nemali žiadnu prácu navyše, rozhodli sme sa vynaložiť úsilie v procese predspracovania, ktorý bude popísaný v práci nižšie.
Hlavnou entitou tohto dasatu sú mobilné zariadenia, takže 1 zariadenie predstavuje 1 riadok súboru.

Keďže rôzne stĺpce majú rôzny význam hodnôt, pozreli sme sa na všetky parametre detailnejšie.
Najväčší rozptyl hodnôt je v stĺpci, ktorý opisuje veľkosť pamäte RAM a je na úrovni 1176643.606, kde minimálna hodnota je 256 a maximálna 3998 GB.

Najmenší rozpyl je 0,1817 a to pri stĺpci hovoriacom o tom, či mobilný telefón má podporu 3G, kde približne 76% má (hodnota 1) a ostatné nie (hodnota 0).

Čo sa týka typu hodnôt, všetky hodnoty máme ordinálne.

Pri analyzovaní hodnôt sme sa pozreli aj na koreláciu hodnôt. Všimli sme si zaujímavú skutočnosť a to takú, že jedinú významnú koreláciu má na cenovú kategóriu iba veľkosť pamäte RAM. Ako ďalšie hodnoty v poradí na cenovú kategóriu vplýva šírka a výška mobilného zariadenia, ale s obrovským odstupom za veľkosťou pamäte RAM.

Ostatné atribúty na cenu nijako významne nevplývajú a tak sme sa rozhodli neskôr dopočítať v implementácii ďalšie stĺpce, ktoré by nám mohli pomôcť lepšie klasifikovať cenovú kategóriu.

Pridať nové stĺpce sme sa rozhodli pre atribúty, ktoré nie sú binárne, to znamená, že neobsahujú iba hodnoty 0 a 1, z dôvodu, že z klasických číselných hodnôt má zmysel pridávať a dopočítavať ďalšie hodnoty, ktoré by mohli mať vplyv na cenu mobilného zariadenia.

	price_range
price_range	1,000
ram	0,917
px_width	0,166
px_height	0,149
int_memory	0,044
sc_w	0,039
pc	0,034
three_g	0,024
sc_h	0,023
fc	0,022
talk_time	0,022
blue	0,021
wifi	0,019
dual_sim	0,017
four_g	0,015
n_cores	0,004
m_dep	0,001
clock_speed	-0,007
mobile_wt	-0,030
touch_screen	-0,030

Tabuľka 1 *Korelácia dát*

Ďalej pri analýze dát sme si vykreslili jednotlivé dáta pomocou algoritmu Interquartile Range. Týmto krokom sme zistili, že väčšina stĺpcov neobsahuje žiadnych outlierov. Medzi stĺpce, ktoré obsahujú outlierov sa nám radili iba tie, ktoré reprezentujú rozlíšenie prednej kamery a výšku zariadení.

Count predstavuje počet záznamov a True count nám hovorí o počte „problémových" riadkov.

	fc	px_height
count	2000	2000
True count	18	2
False count	1982	1998

Tabuľka 2. Interquartile range

Pre samokontrolu sme navyše vyskúšali aj výpočet z_score pre jednotlivé stĺpce. Ako najkritickejší sa nám javil stĺpec s rozlíšením prednej kamery.

Z týchto dôvodov by sme chceli odstrániť hodnoty, ktoré sa priveľmi odlišujú. Použitím algoritmov z_score a iqr by sme chceli získať lepšie výsledky výslednej klasifikácie.

1.2.1 Charakteristiky dát

battery_power – int64, celková energia, ktorú dokáže uložiť batéria v mAh, hodnoty sú od 501 po 1998 mAh, priemerná hodnota je 1238.5185, medián 1226 a smerodajná odchýlka je pomerne vysoká a to 439.418

blue – int64, obsahuje/neobsahuje bluethooth vyjadrené číslami 0 a 1, smerodajná odchýlka má hodnotu 0.5

clock_speed – float64, takt procesora od 0.5 do 3, priemerná hodnota je 1.522 a smerodajná odchýlka je na úrovni 0.816

dual_sim – int64, obsahuje/neobsahuje možnosť duálnej SIM karty, vyjadrené číselne 0 - neobsahuje, 1 – obsahuje

fc – int64, rozlíšenie prednej kamera, hodnoty sú od 0 po 19, priemerná hodnota je 4.3095, medián je 3 a štandardná odchýlka 4.341

four_g – int64, obsahuje/neobsahuje 4G, hodnoty 0 a 1

int_memory – int64, veľkosť internej pamäte v MB, hodnoty sú v rozmedzí od 2 po 64 GB, priemer je 32.0465 a štandardná odchýlka 18.1457

m_depth – float64, hĺbka telefónu v cm, od 0.1 cm po 1 cm, priemerná hĺbka je 0.50175 a štandardná odchýlka 0.2884

mobile_wt – int64, váha telefónu v gramoch, od 80 po 200 g, priemerná váha je 140.249 a štandardná odchýlka 35.4

n_cores – int64, počet jadier procesora, od 1 po 8, priemerne 4.5205, štandardná odchýlka je na úrovni 2.2878

pc – int64, rozlíšenie zadnej kamery, od 0 po 20, priemerné rozlíšenie telefónov je 9.9165 a štandardná odchýlka 6.0643

px_height – int64, výška rozlíšenia v px, od 0 po 1960, priemerne 645.108 a štandardná odchýlka je na úrovni 443.7808

px_width – int64, šírka rozlíšenia v px, od 500 po 1998, priemerne 1251.5155 a štandardná odchýlka je 432.2

ram – int64, veľkosť pamäte RAM v MB, od 256 po 3998, priemerne 2124.213, štandardná odchýlka je teda až 1084.732

sc_h – int64, výška mobilu v cm, od 5 po 19, priemerná výška je 12.3065 a štandardná odchýlka 4.2132

sc_w – int64, šírka mobilu v cm, od 0 po 18, priemerne 5.767, štandardná odchýlka je podobná ako pri výške mobilu a to 4.3564

talk_time – int64, dĺžka hovoru na jedno nabitie, od 2 po 20, priemerne 11.011 a štandardná odchýlka je 5.464

three_g – int64, obsahuje/neobsahuje 3G, hodnoty 0 alebo 1, priemer je 0.7616, takže väčšina mobilov podporuje 3G

touch_screen – int64, má/nemá dotykový displej, hodnoty takisto 0 alebo 1, priemer je na úrovni 0.503, z čoho vidíme že iba približne polovica mobilov z datasetu má dotykový displej

wifi – int64, má/nemá wifi modul, taktiež hodnoty 0 alebo 1, priemer podobný ako pri dotykovej obrazovke a to konkrétne 0.507

price_range – int64, charakterizuje cenovú kategóriu mobilného telefónu, hodnoty sú od 0 po 3, takže celkovo 4 cenové hladiny, priemer je na úrovni 1.5 a štandardná odchýlka 1.1183

2 Definovanie úlohy objavovania znalostí (podiel práce: 50% / 50%)

Dáta v tomto datasete sme použili na zaradenie mobilného zariadenia do jednej zo štyroch cenových kategórií. Riešili sme teda úlohu klasifikácie.

Pre zadané vstupné parametre zariadenia sme predikovali, do akej cenovej kategórie by sme vedeli telefón s danou hardvérovou konfiguráciou zaradiť tak, aby nebol pod cenu a zároveň aby jeho cena nebola príliš vysoká, čo by znamenalo pre potenciálneho predajcu používajúceho náš systém, že si jeho zariadenie nikto nekúpi.

Výhodou datasetu, ktorý sme používali je, že má rovnomerné rozdelenie, čo znamená že predpoveď cenovej kategórie nebude ovplyvnená prevahou niektorej kategórie.

Naša úloha pozostáva z 3 väčších častí.

V prvej časti sme sa zamerali na predspracovanie dát. Keďže všetky stĺpce máme kompletné a ani neobsahujú prázdne/nedefinované hodnoty, nemuseli sme prakticky žiadne predspracovanie vykonávať. Chceli sme si ale skúsiť nejaké techniky z objavovania znalostí a tak sme rozhodli, že niektoré implementujeme a pokúsime sa týmto spôsobom vylepšiť úspešnosť nášho modelu.
Niektoré stĺpce mali hodnoty, ktoré ležali pri mimo bežného rozdelenia, takže prvou vecou, ktoré sme spravili bolo, že sme odstránili tieto hodnoty.

V ďalšom kroku sme dopočítali ďalšie stĺpce, ktorý mali vplyv na cenu a týmto spôsobom sme sa pokúšali zabezpečiť vyššiu presnosť nášho klasifikovania mobilného telefónu do cenovej kategórie. Následne sme použili feature selection na vybranie atribútov, ktoré sú relevantné pre trénovanie modelu.

V druhej časti sme vytvorili už samotný klasifikátor bez použitia hyperparametra aj s jeho využitím.

V poslednej tretej časti sme vyhodnotili jednotlivé modely ktoré sme použili a porovnali ich.

3 Opis prác iných autorov (podiel práce: 50% / 50%)

V práci [1] sa autori zamerali na klasifikačné techniky v data mining-u. Klasifikačné metódy, ktoré zanalyzovali sú napríklad genetické algoritmy, rule-sety (metódy na štýl podmienok ak-tak), C4.5 (meria numerické atribúty, zaoberá sa chýbajúcimi hodnotami a zašumenými dátami), CART klasifikáciu alebo Regression tree, ktorý je založený na presnosť, ak sú hodnoty zašumené alebo chýbajúce.
Na konci sa trochu bližšie pozerajú na vyhodnocovanie, kde predstavujú rady ako správne vyhodnocovať klasifikátory. Spomínajú napríklad delenie datasetu na 2 časti (testovaciu a trénovaciu), krížovú validáciu, ktorú je dobre použiť ak nie je veľa dostupných dát, prípadne rozdelenie datasetu až na 3 časti, kde prvá (najväčšia) je trénovací set, ďalšia je validačný test a posledná slúži na testovanie.

V ďalšej práci [2] riešili autori problém, ako rozdeliť študentov do výkonnostných kategórií.
Pracovali s dátami, ktoré získali zo školského systému Moodle. K dispozícii mali pre každého študenta stĺpce, ktoré vraveli o tom, koľko kurzov študent navštevuje, počet koľko ich prešiel, neprešiel alebo ktoré práve vykonáva, počet správ ktoré sa opýtal na fóre, počet správ ktoré prečítal na fóre, čas ktorý strávil na kvízoch a na fórach a finálne známky z kurzov. Všetky hodnoty boli numerické (tak ako aj v našej práci). Autori sa rozhodli klasifikovať študentov do kategórií 4 : FAIL, PASS, GOOD a EXCELLENT, ktoré by značili úspešnosť študenta, respektíve jeho výsledky. V práci spravili viacero experimentov a využili viacero data mining metód.
Metódy, ktoré sa rozhodli použiť boli Statistial Classifier, Decision tree, Rule Introduction, Fuzzy rule learning a Neural network. Pri týchto metódach rovnako ako my, využili viacero algoritmov ako napríklad ADLinear, Kernel, KNN, CN2, GPP, SIA, AdaBoost. GANN alebo MLPerceptron. Výsledky alebo presnosti klasifikácie, ktoré získali boli v rozsahu od 50% až po 67%. Najhoršie výsledky sa ukázali pri použití neurónovej siete a algoritmu RBFN Decremental a naopak, najlepšie pri Decision Tree a algoritme CART. Veľmi dobré výsledky sa taktiež ukázali aj pri použití Fuzzy rule learning metódy a tak sme sa v našom projekte rozhodli z tejto metódy vyskúšať jeden z algoritmov, konkrétne algoritmus AdaBoost.

Jeden z ďalších článkov [3], ktorými sme sa pri tejto práci inšpirovali sa týkal výberu atribútov. Práve táto časť je pri data mining-u veľmi dôležitá, keďže práve vybrané

atribúty dokážu vo veľkej miere zvýšiť alebo znížiť úspešnosť rôznych algoritmov. Touto problematikou sa zaoberá mnoho autorov v mnohých prácach. V jednej z týchto prác, ktorá nás zaujala a taktiež vo veľkej miere inšpirovala pri tejto práci, sa pokúšali experimentovať s klasifikačnými modelmi KNN, SVM a Naive Bayes a dosiahli veľmi dobré výsledky. Vďaka tomu sme sa aj my rozhodli použiť okrem iných aj tieto modely.

4 Opis metód (podiel práce: 50% / 50%)

4.1 Predspracovanie a výber atribútov

Ako sme už vyššie spomínali, všetky atribúty máme kompletné a žiadne nie sú chýbajúce, nie je teda potrebné dopočítavať prázdne hodnoty.

Preto sme sa rozhodli odstrániť pri niektorých stĺpcoch hodnoty, ktoré sú na okrajoch rozsahu hodnôt. To, ktoré stĺpce obsahujú takéto hodnoty sme zisťovali pomocou metódy Interquartile Range. Na ich samotné odstraňovanie sme použili 2 metódy a to z_score a iqr.

Na samotný výber čŕt, ktorý neskôr budú odstránené sme používali Correlation matrix a Variance inflation factor. Tento výber čŕt sme vykonávali knižničnými funkciami. Pri Correlation matrix sme za vybrané črty považovali tie, ktorých hodnota bola v absolútnej hodnote vyššia ako 0,95.

Ako ďalší krok predspracovania sme sa rozhodli pre dopočítanie ďalších stĺpcov. Viacero stĺpcov máme takých, ktoré predstavujú binárne hodnoty, čo znamená, že majú iba hodnoty 0 a 1. Pri týchto stĺpcoch by nadávalo zmysel dopočítavať ďalšie hodnoty a tak sme ich dopočítavali iba pre tie, ktoré neobsahujú iba 2 hodnoty. Tento krok sa týkal stĺpcov pre veľkosť batérie, clock speed, internú pamäť, hĺbku, výšku a šírku mobilného zariadenia, kvalitu prednej aj zadnej kamery, veľkosť pamäte RAM, dĺžku hovoru, výšku a šírku displeja v pixeloch a váhu telefónu.

Pridávali sme stĺpce s binárnymi hodnotami, kde boolean hodnota reprezentuje, či je aktuálna hodnota viac (1/true) alebo menej (0/false) ako priemer, medián, horný kvartil alebo dolný kvartil. Konečný počet stĺpcov bol 73.

Následne sme robili feature selection, ktorej hlavná myšlienka zakomponovania do tohto projektu spočívala v tom, aby sme odstránili stĺpce, ktoré reprezentujú to isté, respektíve nemajú žiadnu pridanú hodnotu pre učenie sa modelu. Týmto krokom sme chceli zabezpečiť, aby sa trénovaný model neučil nejakú črtu, ktorá nemá žiaden veľký význam pre výslednú klasifikáciu, iba kvôli tomu, že sa jej redundantná informácia nachádza vo viacerých stĺpcoch.

Týmto spôsobom sme zmenili počty stĺpcov, kde sme niektoré odstránili (tie, ktoré nám nedávajú žiadnu informáciu) a ďalšie pridali (stĺpce, ktoré nám môžu pridať dodatočnú informáciu).

Po predspracovaní náš dataset obsahoval pri použití iqr od 1198 po 1216 záznamov a z_score 1600. Počet stĺpcov datasetu po predspracovaní bol pri iqr v intervale od 59 po 65 a pri aplikovaní z_score bol tento počet od 60 po 64.

4.2 DM metódy

Pre samotný klasifikátor sme použili viacero metód, aby sme ich následne mohli porovnať a vedeli povedať, ktorá sa ako správa a ktorou dostávame v našom prípade najlepšie výsledky.

Použili sme nasledujúce modely: AdaBoost, Decision Tree, Extra Trees, Nearest Neighors, Random Forest, SVM Sigmoid, SVM RBF, QDA, Naive Bayes, Linear Discriminant Analysis a Gaussian Process.

Pre tieto sme sa rozhodli na základe prác iných autorov alebo rôznych článkov, ktoré rozoberali problém klasifikácie (primárne pri numerických atribútoch).

Pri jednotlivých modeloch sme vyberali aj hyperparametre na základe metódy Grid search, Random search alebo sme nechávali iba samotný model bez výberu hyperparametrov.

4.3 Vyhodnotenie výsledkov

Na vyhodnocovanie úspešnosti výsledkov sme sa rozhodli použiť overenú metódu, ktorá je pri vyhodnocovaní najpoužívanejšia. Ide o accuracy.

Všetky výsledky, ktoré sú uvádzané v tabuľkách nižšie, sú uvádzané pre accuracy.

5 Experimenty (podiel práce: 50% / 50%)

Keďže sme si chceli vyskúšať viacero modelov, porovnať ich, vyskúšať ako sa zmení ich chovanie pri výbere hyperparametrov, pri odstránení outlierov rôznymi technikami a podobne, spravili sme veľké množstvo obmien a experimentov a sledovali sme, ako sa mení úspešnosť jednotlivých metód.

	Výber stĺpcov		Odstránenie oulierov		Výber hyperparametrov	
	VIF	Correlation matrix	IQR	Z_score	Randomized search	Grid search
AdaBoost	0.7448	0.7372	0.7416	0.7405	0.7433	0.7387
Decision Tree	0.6265	0.6702	0.6618	0.6348	0.652	0.6447
Extra Trees	0.7351	0.7352	0.7272	0.7431	0.7487	0.7216
Nearest Neighbors	0.6905	0.9295	0.8116	0.8083	0.8137	0.8062
Random Forest	0.8633	0.8797	0.8678	0.8752	0.8868	0.8562
SVM RBF	0.6468	0.5185	0.5635	0.6018	0.4776	0.6877
SVM Sigmoid	0.2342	0.2342	0.2415	0.227	0.2342	0.2342

Tabuľka 3. Porovnanie modelov použitím rôznych metód (accuracy) – pri každom z výsledkov bola použitá jedna z metód výber stĺpcov, odstránenie outlierov a výber hyperparametrov, uvádzaný je najlepší výsledok

Ako najlepšia metóda pre klasifikáciu sa nám javila Nearest neighbors, ktorej priemerná úspešnosť klasifikácie accuracy bola na úrovni takmer 93% pri použití correlation matrix. Tuto bolo zaujímavé zistenie, ako veľmi závisí výsledná úspešnosť na metóde výberu parametrov. Pri použití correlation matrich bola táto úspešnosť 93%, pričom pri zmene na VIF klesla iba na hodnotu 69%.

Pri experimentovaní s ďalšími modelmi sa nám podarilo dosiahnuť podobnú, aj keď o 4% nižšiu úspešnosť pri nastavovaní hyperparametrov pomocou randomized v Random forest-e. Random forest preukazoval stabilnú klasifikáciu aj pri výbere atribútov pomocou VIF, correlation matrix, výbere parametrov cez iqr alebo z_score. Vždy sa priemerná úspešnosť pohybovala ood 85% po 89%.

Veľmi slabé priemerné výsledky dával SVM Sigmoid, ktorého úspešnosť bola niekde na úrovni 23-24%, čo je podobná úspešnosť ako náhodné tipovanie. Na zlepšenie nepomohla ani zmena výberu outlierov, zmena metódy pre výber hyperparametrov a podobne. Táto metóda nie je teda vôbec vhodná na takýto typ úlohy, akú riešime.

Model	Feature Selection	Výber hyperparametrov	Odstránenie oulierov	Accuracy
Nearest Neighbors	Correlation matrix	Randomized search	z_score	0.9325
Random Forest	Correlation matrix	Randomized search	z_score	0.896
Extra Trees	Correlation matrix	Randomized search	z_score	0.755
AdaBoost	VIF	Grid search	z_score	0.7535
SVM RBF	VIF	Grid search	iqr	0.7465
Decision Tree	Correlation matrix	Randomized search	z_score	0.7085

Tabuľka 4. Porovnanie modelov použitím rôznych kombinácií metód

Čo sa týka ostatných metód, priemerné výsledky sa nejako veľmi nelíšili, boli relatívne stabilné pri experimentovaní s metódami na výber parametrov, hyperparametrov a na odstránenie oulierov. Výkyv hodnôt v rámci jednej metódy bol najviac 15% (čo bolo ale iba v jednom prípade pri SVM RGB). Všeobecne táto hodnota kolísala okolo nejakých +-5%.

Ďalej sme sa snažili zistiť, ktorý model a s akými metódami na výber hyperparametrov, odstraňovanie oulierov a podobne dosiahneme najlepšiu úspešnosť klasifikácie.

Najlepšiu úspešnosť vypočítanú pomocou accurency a to až na úrovni 93.25% sa nám podarilo dosiahnuť pomocou Nearest neighbors. Feature selection sme vykonávali pomocou correlation matrix, na výber parametrov sme použili randomized search a oulierov sme odstraňovali pomocou z_score.

Práve táto kombinácia (feature selection pomocou correlation matrix, výber parametrov pomocou randomized search a odstránenie outlierov pomocou z_score) nám pokrýva prvé 3 najlepšie výsledky pri rôznych modeloch. Vďaka tomu vidíme, že práve tieto metódy sú najlepšie pre numerické dáta.

Úspešnosť na úrovni necelých 90% sa nám podarilo dosiahnuť ešte pomocou Random forest a úspešnosť 75% pomocou Extra tree.

Ďalšia v poradí bola metóda AdaBoost. Pri tomto modeli sa ako lepšia metóda na feature selection javilo VIF a pre výber hyperparametrov Grid search. Úspešnosť bola len o 2 desatiny percenta nižšia ako pri Extra tree.

Relatívne úspešné výsledky nám dávalo ešte SVM RBF a Decision, ktorých úspešnosti klasifikácie boli nad hodnotou 70%.

Ďalej si uvedieme hyperparameter 3 najlepších modelov z tabuľky 4.
Parametre pri modeli Nearest Neighbors boli nasledovné: n_neighbors bolo z intervalo 3-20, leaf_size [1, 3, 5, 7], použité algoritmy boli auto, kd_tree, ball_tree a brute. V tabuľke 4 je Nearest Neighbors s výsledkom 0.9325, ktorého hyperparameter algoritmus bol nastavený na brute, leaf_size mal hodnotu 7 a n_neighbors obsahovalo hodnotu 17.

Pri modeli Random Forest bol nastavený range v intervale 20-70, n_estimators bolo z intervalu 10-70 a max_features boli sqrt, log2 a None. Nastavenie Random Forestu z tabuľky 4 bolo max_depth na 41, max_features None a n_estimators mal hodnotu 51.

Model Extra Trees zase mal parametre n_estimators z intervalu 10-80 a criterion gini a entropy. Najlepší výsledok, ktorý je uvedený v tabuľke mal nastavený parameter criterion na entropy a počet n_estimators bol 70.

Ďalej sme experimentovali s najlepším modelom, ktorým bol Nearest neighbors a porovnávali sme ako použité metódy vplývajú na výslednú úspešnosť. Vo všeobecnosti correlation matrix dával oveľa lepšie výsledky ako použitie VIF. Čo sa týka odstraňovania krajných hodnôt pomocou metód z_score a iqr, tam neboli rozdiely nejako výrazne. V najlepšej kombinácii rôznych metód síce figuruje z_score ale rozdiel oproti použitiu iqr je len v jednej desatine percenta. Podobný minimálny rozdiel je aj pri ostatných kombináciach.

Model	Feature Selection	Výber Hyperparametrov	Odstránenie oulierov	Accuracy
Nearest Neighbors	Correlation matrix	randomized search	z_score	0.9325
			iqr	0.931
		Grid search	z_score	0.928
			iqr	0.9265
	VIF	Randomized search	iqr	0.699
			z_score	0.6925
		Grid search	iqr	0.69
			z_score	0.6805

Tabuľka 5. Úspešnosť najlepšieho modelu s rôznymi metódami

6 Zhodnotenie (podiel práce: 50% / 50%)

V tomto projekte sme vytvorili viacero typov modelov pre klasifikáciu a porovnali ich. Rovnako sme použili viacero metód pre odstraňovanie krajných hodnôt, feature selection alebo pre výber hyperparametrov a vyhodnocovali sme najlepšiu kombináciu vzhľadom na náš typ úlohy – klasifikácia numerických hodnôt do viacerých kategórií.

Zistili sme, že najlepšie výsledky má kombinácia Correlation matrix pre feature selection, Randomized search pre výber hyperparametrov a z_score pre odstraňovanie parametrov bez ohľadu na model. Práve táto kombinácia nám dávala najlepšie výsledky pri modeli Nearest neighbors (93,25%), Random forest (89,6%) a Extra trees (75.5%) .

Samozrejme, úspešnosť závisela vo veľkej miere od modelu, ktorý bol použitý. Preto sme sa rozhodli pre najlepší model zanalyzovať aj použitie rôznych kombinácií metód pre feature selection, výber hyperparametrov a odstraňovanie krajných hodnôt.

Zistili sme, že pre feature selection je oveľa vhodnejšie použitie Correlation matrix ako VIF, kde pri použití Correlation matrix sme dostali výsledky od 92.65% do 93.25% narozdiel od VIF, kde výsledky boli podstatne horšie a to od 68.05% po 69.9%.

Čo sa týka odstraňovania krajných hodnôt, tam môžeme považovať obe metódy (z_score aj iqr) za vhodné, keďže obe dosahovali veľmi dobré výsledky a líšili sa len minimálne.

Literatúra

[1] Sujatha, Prabhakar, Lavanya Devi : A Survey of Classification Techniques in Data Mining, International Journal of Innovations in Engineering and Technology (2013)

[2] Ryan Shaun Joazeiro de Baker : Educational Data Mining 2008. The 1st Inteational Conference on Educational Data Mining Montréal, Québec, Canada, June 20-21 (2008), pp. 8-15

[3] Nascimento, L.D.O. et al.: An investigation of genetic algorithm-based feature selection techniques applied to keystroke dynamics biometrics An investigation of genetic algorithm-based feature selection techniques applied to keystroke dynamics biometrics. SBSeg, 2–5 (2019).