UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Analýza zhlukov transakčných blockchainových dát pre sieť Etherea

Skúškový projekt z predmetu Analýza zhlukov a klasifikácia dát

2023

Adam Martinka

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Analýza zhlukov transakčných blockchainových dát pre sieť Etherea

Skúškový projekt z predmetu Analýza zhlukov a klasifikácia dát

Študijný program:	Ekonomicko-finančná matematika a modelovanie
Študijný odbor:	1113 Matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky

Bratislava, 2023

Adam Martinka

1 Úvod

V úvode je potrebné poznamenať, že vo veľmi veľkej časti budeme v kapitolách 1 a 2 čerpať z nášho projektu pre predmet Redukcia dimenzie dát [2].

1.1 Intro do blochainu

Pred samotným uvedením cieľu projektu alebo predstavením samotných dát, považujeme za potrebné si vysvetliť aspoň tie najnutnejšie koncepty v oblasti blockchainu pre porozumeniu dát a výsledkov v projekte.

Sieť Etherea sa dá pochopiť ako p2p - peer to peer (tj. decentralizovaná) sieť, ktorá validuje rôzne transakcie medzi ľubovolnými účastníkmi siete. Pre rýchle porozumenie môžeme uviesť jednoduchý príklad: užívateľ A chce poslať užívateľovi B svoje **prostriedky** (napr. 1 ETH) cez sieť Etherea. Ak užívateľ A má vytvorenú tzv. peňaženku a pozná **addresu** (tj. ID) peňaženky užívateľa B, tak po zaplatení gas fees (poplatkov) sa vykoná daná požiadavka na presun prostriedkov. Addresa peňaženky je voľne vyhľadateľná ale je aj kompletne anonymná do bodu, pokiaľ vlastník peňaženky neprezradí jeho vlastníctvo danej peňaženky verejnosti (Tu zrejme A vie, že peňaženka patrí B). Potvrdenie a rôzne validácie zabezpečuje samotná sieť. Ak požiadavka prejde cez sieťové validácie, tak prenos prostriedkov bude do malej chvíle schválený a aktualizovaný.

Pod pojmom 'sieť' je potrebné poznamenať, že je tvorená 'nodmi', ktorí sú viacej mediálne známi ako **miners**. Mineri potvrdzujú, takéto transakcie, za čo sú patrične odmeňovaní v digitalnej mene ETH. Takýchto transakcií sa spracuje niekoľko stoviek v krátkom okamžiku a vložia sa do takzvaného bloku. Okrem toho, že sú mineri platení za poskytovanie služieb (napr. svoj počítač) a za validácie transakcií, tak medzi sebou aj súperia o to, kto zvaliduje celý takýto blok, pred tým ako sa prejde na nový. V takomto prípade sa **vyberie jeden náhodný miner** aby potvrdil blok, ktorý dostane ďalšiu patričnú odmenu v mene ETH.

Na sieti Etherea môže ľubovolný používateľ A poslať svoje prostriedky na peňaženku vlastnenú burzou, ktorá potom prekonvertuje digitálnu menu za skutočné fiat a tie pošle

užívateľovi na jeho bankový účet.

Najpodstatnejším rozdielom medzi sieťou Etherea a Bitcoinu sú jednoznačne, **smart kontrakty**. Jednoduchý scénar použitia smart kontraktu by bol ako v predošlom príklade, s tým rozdielom, že užívateľ A by chcel aby jeho prostriedky odišli hneď z jeho peňaženky ale užívateľovi B prišli presne o 10 dní. Tým pádom prostriedky najprv putujú na adresu kontraktu a o 10 dní automaticky na adresu užívateľa B. Pod smart kontraktom si môžeme predstaviť akýsi napísaný (open source) kód, ktorý vie vykonávať rôzne funkcionality od prevodov a tradingu, až po obchodovanie s derivátmi medzi rôznymi účastníkmi siete alebo pripísanie vlastenectva rôznych obrázkov (NFT). Ak sa kontrakt týka finančného charakteru, tak dostáva označenie **DeFi**.

Na záver podotknime, že adresa v Ethereum sieti môže byť peňaženkou alebo smart kontraktom. Majoritu však tvoria práve peňaženky (99,9%). V projekte však každú takúto entitu bez znalosti klasifikácie budeme označovať ako adresu.

1.2 Spracovanie dát

Úlohou v tomto projekte bude analýza zhlukov adries v Ethereum sieti podľa ich atribútov. Keďže získanie nejakých relevantných dát bolo namáhavé, vhodnou cestou sme si ich získali sami. Na kratšie časové obdobie sme sa pomocou vlastného skriptu v jazyku TypeScript napojili na sieť Etherea a získali transakcie medzi rôznymi adresami z celkových 98 blokov, ponechali sme iba tie, v ktorých sa presúvala nejaká čiastka ETH, čo vo výsledku predstavovalo celkovo 10063 transakcií. V každej transakcií sme mali získané údaje ako **kto** (adresa) posielal **komu** (adresa) s **akým obnosom** (ETH). Takéto dáta sú sieťového (tj. grafového) charakteru, na ktoré sa používajú predovšetkým algoritmy určené pre siete. Pre účely projektu sme si z transakcií vytvorili novú dátovú sadu unikátnych peňaženiek získaných za takéto časové obdobie. Dáta sme uložili, tak že sme ku každej peňaženke vypísali dodatočné číselne atribúty, ktoré boli takisto získane zo samotných transakčných dát. V transakčných dátach sa nachádzala aj informácia, aké adresy zvalidovali rôzne bloky, ak sa takéto adresy nachádzali v našom vytvorenom datasete, tak dané

adresy dostali klasifikáciu s označením 'Minning' a typom adresy ako 'Wallet'. Následne sme sa viacerými spôsobmi snažili olabelovať aj iné adresy, napr. pomocou datasetu v [1] alebo pomocou voľného vyhľadávania cez internet. Takto sme niektoré adresy vedeli označiť, (ďalej ako 'olabelovať') či sa jedná o typ peňaženky a či patrí burze (bude niesť ozn. 'Exchange') alebo je anonymná, alebo či sa jedná o typ smart kontraktu a či jeho konkrétny typ je finančného charakteru (bude niesť ozn. 'DeFi'). Treba poznamenať, že pri labelovaní sme úmyselne vyberali adresy, ktoré boli niečim z daných premenných oproti ostatným signifikantné, napr. za sledovaný čas vykonali čo najviac posielacich transakcií. Príklad rozdielu signifikantnosti môžeme hneď vidieť na Obr. 1, kde riadok s označením 'Exchange' ma oproti ostatným 4 riadkom veľký rozdiel v premennej 'totalSent', síce zatiaľ bez poznania významu premennej.

	minBalance	maxBalance	entity	countAsSender	totalSent	totalReceived	coununtAsReceiver	avgSent	avgReceived	currentBlock	id	avgBalance	addressType
1	8.196888e+03	8.336878e+03	Exchange	136	137.2153	0.000000	0	1.0089361	0.0000000	17196119	0x46340b20830761efd32832a74d7169b29feb9758	8.255174e+03	Wallet
2	1.044000e-02	1.044000e-02		0	0.0000	0.005210	1	0.0000000	0.0052100	17196022	0x2e2ccece147daf7f5c29baafa5357944008cd55b	1.044000e-02	
3	2.449601e-01	2.449601e-01		0	0.0000	0.190400	1	0.0000000	0.1904000	17196022	0x22aa532745236110600 4be7cd4afe348d471dfe9	2.449601e-01	
4	3.627691e-02	3.627691e-02		0	0.0000	0.020990	1	0.0000000	0.0209900	17196022	0x3556516c41edd322414d52cb1d29b94746d4956b	3.627691e-02	
5	7.271570e+02	7.294682e+02		21	12.7011	7.086499	64	0.6048145	0.1107266	17196119	0x6dfc34609a05bc22319fa4cce1d1e2929548c0d7	7.282677e+02	

Obr. 1: Vzorka dát, s ktorými sa bude pracovať v projekte . (Kvôli veľkosti šírky sa uprednostnil obrázok.)

Ak by nebolo z názvov stĺpcov v Obr. 1 zrejmé, ako ich interpretovať, tak nižšie popisujeme význam jednotlivých premenných. Pripomíname, že hodnoty premenných sú získané len z času sledovacieho horizontu a nie za celý čas existencie adries.

- minBalance/maxBalance - minimálny/maximálny balance adresy,
- entity - klasifikácia konkrétneho typu adresy, v prípade peňaženky, to môže byť burza, miner alebo unknown a v prípade smart kontraktu sa môže jednať o DeFi alebo unknown.
- countAsSender/countAsReceiver - koľkokrát daná adresa vykonala/prijala transakciu s ETH,
- totalSent/totalReceived - aký celkový objem ETH daná adresa odoslala alebo prijala,

- avgSent/avgReceived - aká bola priemerná výška (v ETH) jednej poslanej/prijatej transakcie
- id - samotná unikátna adresa,
- avgBalance - priemerný balance adresy,
- addressType - typ adresy, tj. či sa jedná o peňaženku alebo smart kontrakt.

2 Prvotná analýza dát

Pre naše dáta si v aktuálnej kapitole ukážeme nejaké základne štatistiky ako aj vizuálnu reprezentáciu dát v menej rozmernej podobe s použitím metódy PCA pre 2D a 3D zobrazenie. Základne štatistiky pre naše dáta:

- celkový počet dát: 10 049,
- z toho máme iba 49 adries označených typom peňaženky a 10 ako smart kontrakt,
- ďalej zo 49 peňaženiek patrí 13 minerom a 19 burzám a zo spomínaných 10 smart kontraktov sa v prípade 7 jedná o DeFi. Je vidieť, že počet klasifikovaných je v porovnaní s celkovým počtom adries len maličkým zlomkom.
- Ďalej uvádzame výsledky aritmetických priemerov a mediánov pre naše premenné v Tabuľke 1.

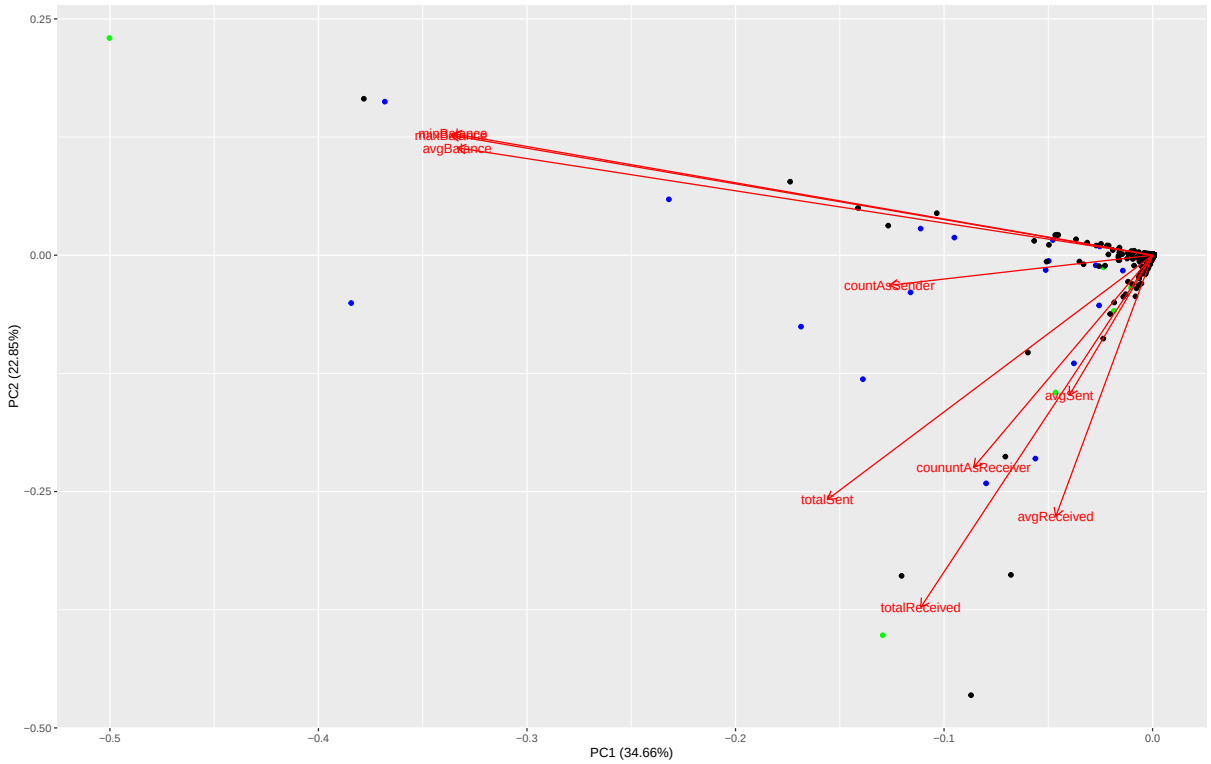
Tabuľka 1: Priemer a medián premenných z našich dát

Premenná	Priemer	Medián
minBalance	528.219775	0.05300330
maxBalance	530.784603	0.08171367
countAsSender	1.001393	1.00000000
totalSent	3.919808	0.01780778
totalReceived	2.577443	0.00000000
countAsReceiver	1.001294	0.00000000
avgSent	1.435696	0.00000000
avgReceived	1.379314	0.00000000
avgBalance	496.785438	0.03116136

Z Tabuľka 1 je jasné vidieť rozdiel medzi výsledkami priemeru a mediánu. To môže zapríčiniť presne to, že väčšina adries má malé hodnoty skúmaných premenných ale existuje pár adries, ktoré majú pre tie isté premenné astronomické hodnoty. Bližšia analýza takýchto adries bola opodstatnená v projekte pre predmet redukcia dimenzie dát v [2]. Pre nás je však dôležité, že medzi dátami sa nachádza jedna špecifická adresa, ktorá dokázala posunúť priemery atribútov spojených s balancom o niekoľko stoviek ETH vyššie. Pre získanie relevantných výsledkov zhlukovacich metód spolu s využitím PCA vyradíme dané dáto z pôvodného datasetu (metódy by ho so signifikantnou pravdepodobnosťou zaklasifikovali ako samostatný zhluk a PCA1 krivka by všetky premenné okrem 'balancových' považovala za nesignifikantné). Pre zaujímavosť môžeme poznamenať, že dané dáto interpretuje smart kontrakt DeFi charakteru. Konkrétne sa jedná o Wrapped Eth kontrakt, ktorého transakcie ako aj stav balancu si vieme pozrieť na `https://etherscan.io/address/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2`. *Jedná sa o kontrakt, ktorý zabezpečí výmenu pôvodnej meny ETH za tzv. ERC-20 token WETH. Niektoré burzy alebo aj DeFi platformy dokážu pracovať iba s tokenovou verziou ETH (tj. ERC-20 tokenom - WETH) a tak ak chce užívateľ využívať takéto služby pre jeho ETH je potrebná najskôr zámena.

Na Obr. 2 môžeme vidieť grafický výstup už zprojektovaných dát na prvé dva hlavné komponenty. Samotné dáta PCA sú vykresľované 4 rozličnými farbami.

- **zelenou** sú vykresľované dáta smart kontraktov s labelom **DeFi**,
- **modrou** sú vykresľované dáta peňaženiek s labelom **Exchange**,
- **červenou** vykresľované dáta peňaženiek s labelom **Mining**,
- **čiernou** sú vykresľované všetky **ostatné adresy**, ktoré sa nám nepodarilo olabelovať aj keď väčšina z nich zrejme budú peňaženky súkromných osôb.



Obr. 2: Výsledok PCA pre dáta bez outliera

Na osi PCA1 vidíme, že minBalance, maxBalance, avgBalance idú takmer identickým smerom, čo sa dá isto aj intuitívne očakávať, keďže sú medzi nimi silné korelácie a aj značná závislosť. Keď sa pozrieme na stranu PCA2 tak vidíme, že posielacie premenné sú naklonené akýmsi vlastným smerom viacej doľava a prijímacie premenné sú zgrupené viacej doprava, čo nám dáva akési dobré informatívne zobrazenie, či daná adresa viacej (alebo silnejšie) participuje ako odosielateľ alebo prijímateľ ETH. Vo výsledku by sme dáta posunuté na osi PCA1 viacej doľava mohli interpretovať ako adresy, ktoré majú väčší stav účtu a dáta, ktoré sú posunuté na osi PCA2 smerom dole, môžeme interpretovať ako adresy, ktoré sú na blockhaine aktívnejšími. Tiež si môžeme všimnúť, že adresy, ktoré vykonávajú viacej posielacich úkonov majú podľa výsledku PCA väčší objem ETH ako tí, ktorí viacej prijímajú, čo je jednoznačne zaujímavý ukazovateľ.

Na záver 2D PCA výsledku si môžeme všimnúť, že síce sme mali zlomok zaklasifikovaných adries, aj tak je jasne vidieť, že Exchange a DeFi, tvoria najaktívnejšiu časť adries, ktoré najčastejšie posielajú/prijímajú transakcie spolu s najväčšími hodnotami ETH a zároveň predstavujú aj signifikantnú časť účtov, ktoré majú najväčší balance. Mineri zrejme podľa

výsledkov 2D PCA nie sú až tak aktívni a nemajú ani veľký balance a sú v nejakom spoločnom 'zhluku' s ostatnými nezaklasifikovanými adresami. Výsledky PCA v projekte použijeme v projekte niekoľkokrát, vždy po aplikacií zhlukovacieho algoritmu a preto sme pokladali za dôležité si najprv predstaviť zobrazenie samotných dat v PCA prostredí. *Čisto pre zaujímavosť prikladáme aj link na interaktívny grafický výstup PCA s využitím prvých troch hlavných komponentov, kde môžeme pozorovať aj umiestnenie peňaženiek s označením Mineri, ako aj lepšiu vizualizáciu toho, že posielajúce premenné v sebe zahŕňajú o niečo väčší balance adresy ako prijímacie premenné:

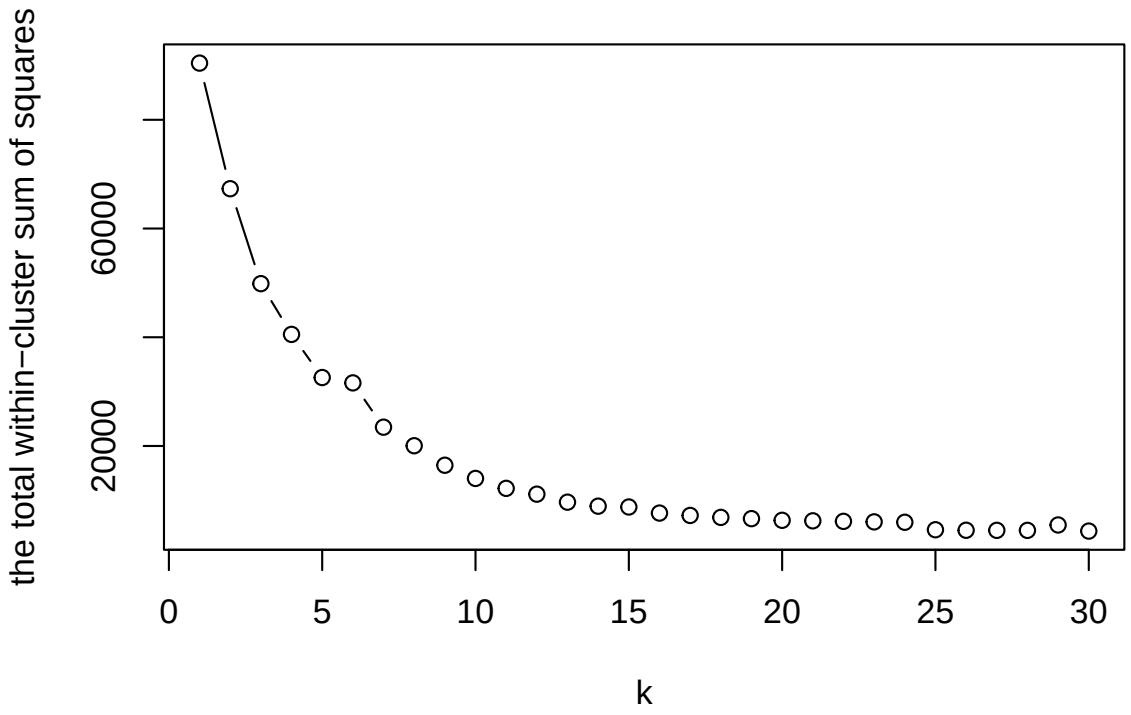
`https://6458e56a0554c82be443ed6d--shimmering-chaja-d1d40b.netlify.app`

3 Zhľukovacie a klasifikačné metódy

V tejto časti sa budeme venovať aplikácií zhlukovacich a klasifikačných metód na naše dáta. Ako predčasné očakávania výsledkov metód bude rozdelenie dát do zhlukov predovšetkým podľa veľkosti balancu a podľa aktivity na sieti, teda podľa Obr. 2, by mohli poskytovať aj informatívne rozdelenie podľa typu entity adresy. Zároveň budeme chcieť vytvoriť model, ktorý by dokázal efektívne klasifikovať dáta na základe ich atribútov.

3.1 K-Means

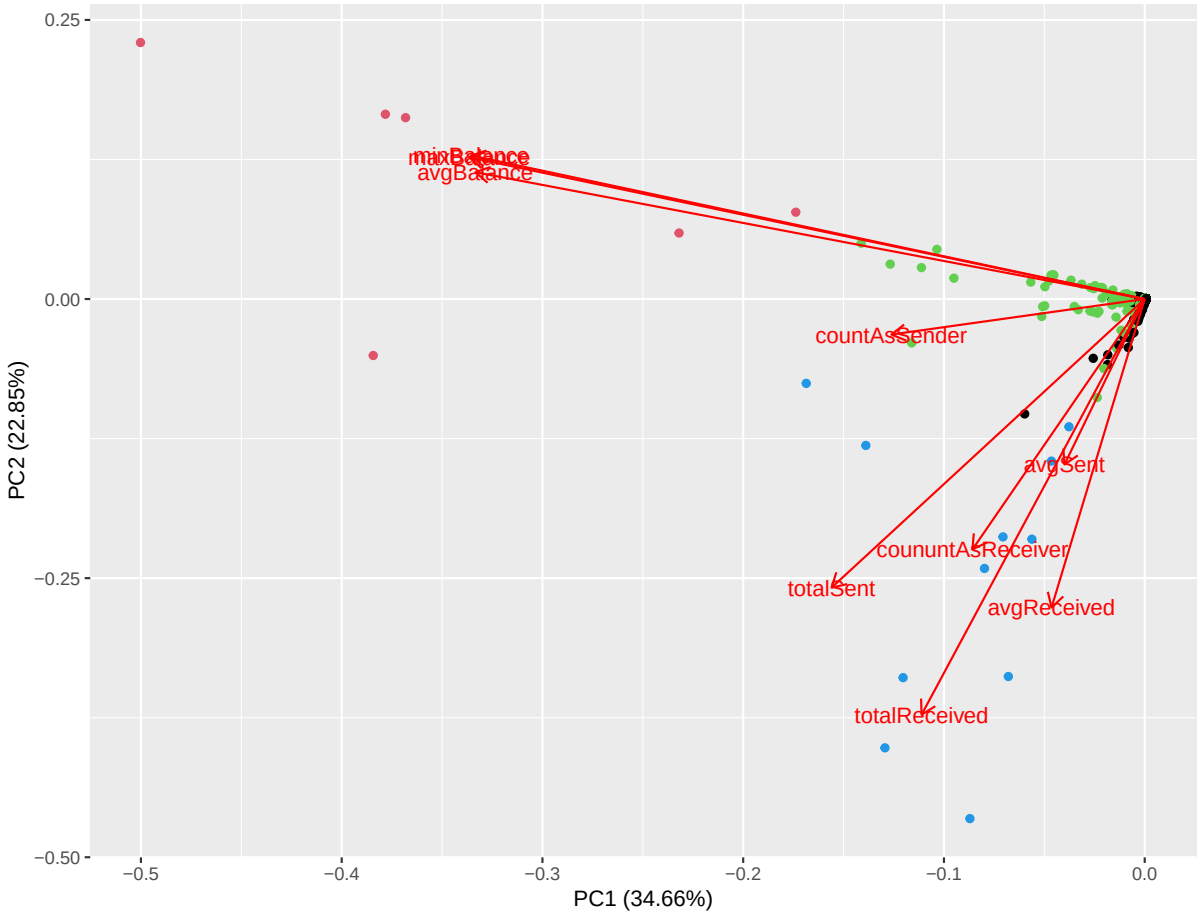
Metóda K-Means funguje na princípe minimalizácie vzdialenosti bodov v jednotlivých zhlukoch od centroidov daných zhlukov, pričom počet zhlukov je predom daný. Keďže dáta nemajú vyvážené číselne hodnoty atribútov, tak ich musíme najprv štandardizovať pre aplikáciu tejto ale aj ostatných metód v projekte. Po štandardizácií si môžeme klásť otázku aký optimálny počet zhlukov zvoliť pre danú metódu. Mohli by sme sa pozrieť na Obr. 2 a z toho určiť aký je podľa nás počet zhlukov. Miesto toho si ale metódu K-Means necháme aplikovať napr. 30krát vždy pre rôzny počet zhlukov tj. $k \in \{1, 2, \ldots, 30\}$ a pre každý jeden výsledok spočítame súčty štvorcov vzdialenosti bodov od centroidu zhluku, v ktorom sa body nachádzajú.



Obr. 3: Výsledky sumy štvorcov vzdialeností od centroidov vzávislosti od zvoleného k

.

Na Obr. 3 môžeme vidieť aké výsledky mala metóda pre zvolené k, my si podľa vlastného uváženia zvolíme $k = 4$. Zvolili sme akýsi zlatý stred, tak aby k nebolo zbytočne veľké a aby nebol veľký súčet vzdialeností. Metódu aplikujeme na preškálované dáta a výsledne zhlukovanie zobrazíme na Obr. 4 s použitím PCA.



Obr. 4: Výsledky zhlukovania metódy K-Means ($k = 4$) s využitím 2D PCA

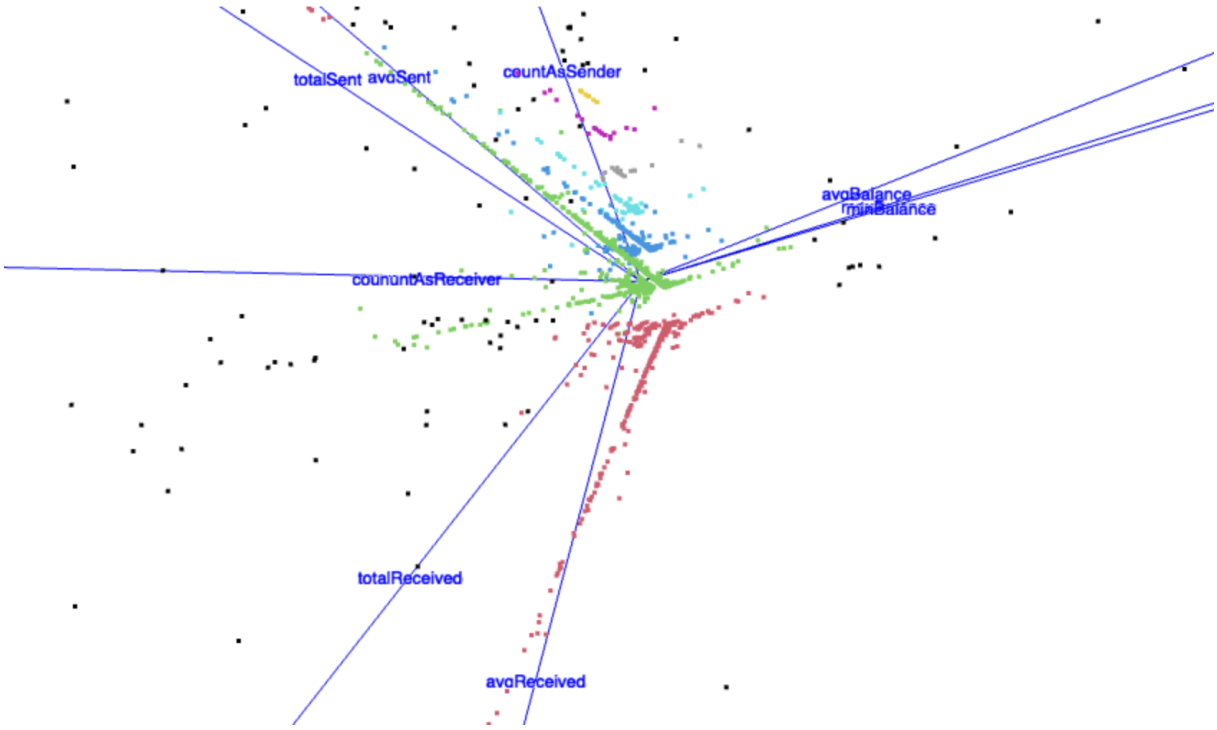
.

Na Obr. 4 môžeme vidieť pozoruhodné výsledky. Aj keď je K-Means jednoduchšou metódou vcelku dobre rozdelil dáta na akési 4 zóny, podľa farieb bodov to sú: zóna vysokého balancu s nižším počtom transakcií, zóna stredného balancu s nižším počtom transakcií, zóna menšieho balancu s vysokým počtom transakcií a nakoniec zóna s malým balancom a malou až stredne veľkou aktivitou na sieti, ďalej ako mŕtva zóna. Aj keď známou nevýhodou metódy je, že výsledne zhluky závisia od náhodne zvolených počiatočných stredov, tak pri opätovnom spúštaní boli výsledky takmer stále identické. V dôsledku 'olabelovania' niektorých adries (Obr. 2) vieme povedať, že metóda nie len efektívne vytvorí zhluky pre dáta podľa ich atribútov ale v celku dobre by sa vedela použiť ako 'unsupervised' klasifikátor pre charakter adresy. Napríklad, výsledok metódy na nových dátach pre $k = 4$ by sa mohol dať interpretovať tak, že adresy, ktoré boli klasifikované do zhlukov a ktoré odpovedajú zónam buď s väčším počtom transakcií alebo s väčším balancom budú zrejme peňaženky

búrz alebo to budú DeFi kontrakty. Na záver metódy pridávame jej zobrazenie klasifikácie s použitím 3D PCA `https://646fd4a6a172410b2e3bfd20--sunny-sable-9a9a80.netlify.app`.

3.2 DbScan

V minulej metóde sme použili zhlukovací algoritmus, ktorý vytvára zhluky guľového typu. Teraz pre zmenu aplikujeme metódu, ktorá dokáže vytvárať zhluky v rôznych útvaroch a skúsime porovnať ich výsledky. Algoritmus funguje na princípe, v ktorom sa každý bod označí ako hlavný bod, ďalej ako core point alebo nehlavný bod, ďalej ako not core point. To či je bod core alebo not core sa určuje podľa toho či sa v jeho okolí $\epsilon > 0$ nachádza aspoň n iných bodov. Oba parametre ϵ, n sú voliteľné a budú ovplyvňovať výsledný počet zhlukov. Ak je nejaký bod core pointom a dosahuje na iné core pointy a tie zas dosahujú na iné core pointy atď., tak takéto dáta dostavajú rovnakú klasifikáciu. Zhluk sa uzavrie tým, že core pointy narazia na dáta, ktoré už vo svojom ϵ okolí majú menej ako n iných bodov (not core points), takéto dáta budú pridané do zhluku ale oni sami už ďalej zhluk nerozširujú. Aplikovať DbScan budeme chcieť s takými parametrami $\epsilon = 0.12, n = 5$,tak aby výsledný počet zhlukov bol taký istý ako v prípade K-Means (pre lepšie porovnávacie účely). Očakávame, že zhlukovanie sa bude odohrávať predovšetkým v mŕtvej zóne (málo transakcií a malý balance) práve kvôli tomu, že zvolené ϵ je malé a počet susedov relatívne veľký na to aby sa vytvorili zhluky podobne ako v prípade výstupu K-Means. Rovno si ukážeme 3D graf, ktorý si môžeme interaktívne zobraziť na: `https://64713c02e519210c6f996a39--melodious-alpaca-f81712.netlify.app`. Po dostatočnom priblížený na zónu menej aktívnych peňaženiek môžeme pozorovať zaujímavé zhlukovanie Obr. 5.



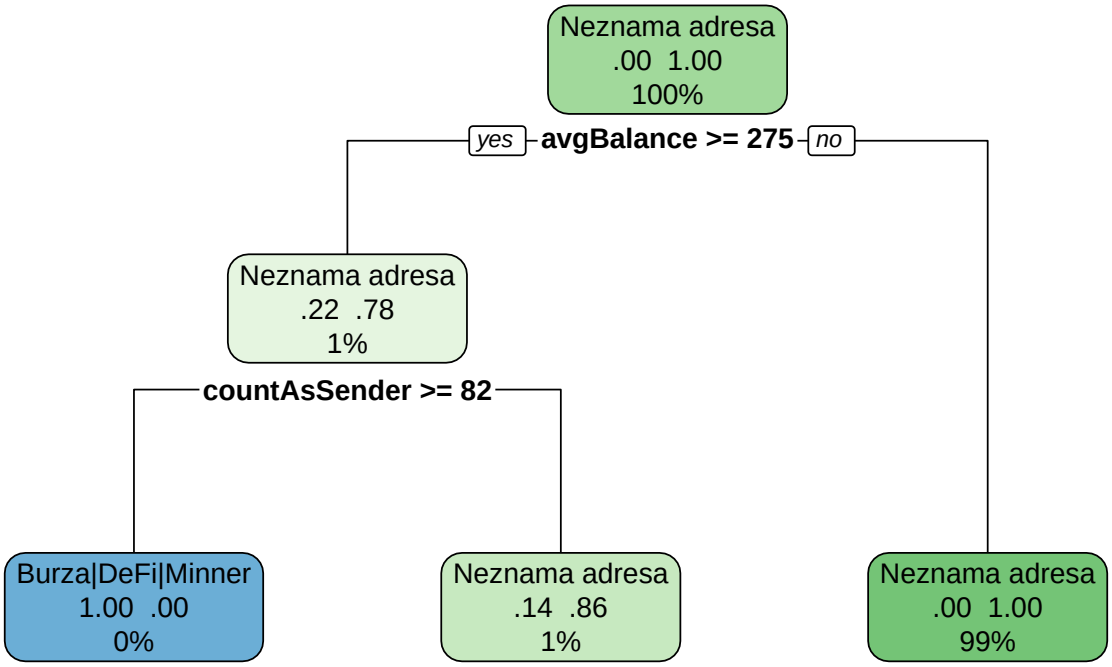
Obr. 5: Grafický výsledok zhlukovania DbScanu($\epsilon = 0.14, n = 5$) v zobrazení 3D PCA po dostatočnom priblížení.

Metóda K-Means pred tým vytvorila nami 4 označené zóny, pričom jedná z nich bola označená ako mŕtva zóna. Tu ale môžeme vidieť výhodu DbScanu v tom, že sa mu podarilo vyrobiť zhluky rôznych tvarov, kde body takýchto zhlukov sú v vzájomnej špecifickej blízkosti a vďaka tomu vieme pozorovať zhluky aj v takejto mŕtvej zóne. Po dostatočnom priblíženi môžeme vidieť, že sa DbScanu podarilo rozdeliť takéto adresy do zaujímavých zhlukov, ktorých dáta v priemere prijímajú a odosielajú objemovo väčšie (ETH) transakcie a na nejaké dva zhluky, ktoré sú blízko seba a tie obsahujú adresy, ktoré častejšie posielajú objemovo väčšie transakcie.

3.3 Klasifikačný strom a les

Keďže sme si dali námahu a niektoré adresy máme 'olabelované', tak sa to posnažíme prakticky zúžitkovať v aktuálnej kapitole, kde si vytvoríme model na predikciu budúcich stavov premennej 'entity'. Pripomeňme si, že premenná entity je klasifikačného chrakteru a v našom datasete nadobúda hodnoty 'Exchange', 'DeFi', 'Minning' alebo je označená

prázdnou hodnotou, čo implikuje neznámu adresu. Ako sme spomínali v úvode, tak 'olabelovaných' dát nemáme veľký počet a ak by sme chceli vytvoriť model, ktorý by predikoval či je premenná 'entity' niektorá z konkrétnych hodnôt, tak v dôsledku ešte menšieho počtu zastúpenia jednotlivých kategórií by sme mohli dostať model, ktorý nezachytí všetky možné klasifikácie. Pre potreby lepšieho modelu modifikujeme dáta tak, aby premenná entity predstavovala iba 2 rôzne hodnoty. Novými hodnotami budú 'zaujímavá adresa', keď sa bude jednať o jednú z olabelovaných a 'nezaujimavá adresa', ak k nej nemáme vytvorený žiaden label. Vo všetkých prípadoch vytvoríme model na trénovacej sade dát (70%) a vyhodnocované úspešnosti budú vykonané na testovacích dátach (30%).



Obr. 6: Výsledny klasifikačný strom pre 2 možné hodnoty premennej entity

.

Takýto strom bol vytvorený z trénovacie dát a jeho podstata je najlepšie vysvetliteľná na nových dátach. Zoberme si nejakú novú adresu za nové sledovacie obdobie siete Etherea. Pre danú adresu máme vytvorené opäť všetky číselne údaje ale nevieme o akú entitu sa jedná. Na danú peňaženku môžeme však spustiť sadu otázok z Obr. 6, postupne z vrcholu až na spodok. Prvá otázka by smerovala na balance peňaženky, ak by bol väčší ako 275 prechádzali by sme na otázku na ľavo (v opačnom prípade na pravo), kde sa strom pýta či daná adresa vykonala za sledované obdobie viacej ako 82 transakcií, ak je opäť naša odpoveď pravdivá tak spadáme do takzvaného terminálneho uzla, ktorý už neobsahuje žiadnu ďalšiu otázku a klasifikuje našu peňaženku s označením zaujímavej peňaženky (prvý textový riadok uzla). Takýmto spôsobom vidíme, že sa na Obr. 6 nachádzajú celkovo 3 terminálne uzly, z čoho práve dva klasifikujú adresy ako nezaujímavé. Každý uzol obsahuje aj dodatočné informácie ako (v poradí z hora na dol): ktorá skupina dominuje v danom uzle, aký je pomer medzi dvoma skupinami a posledný riadok hovorí o tom aký je pomer veľkosti uzla (koľko dát sa tam dostalo) voči veľkosti celkového datasetu alebo voči veľkosti prvého uzlu.

V Tabuľke 2 si môžeme pozrieť výsledky klasifikácie na testovacich dátach.

Tabuľka 2: Kontingenčná tabuľka, zobrazujúca úspechy a neuspéchy klasifikácie modelu na testovacích dátach.

Aktuálne / Predikované	1	0
1	2	11
0	0	3002

Výsledná senzitivita (úspešnosť označenia adresy ako zaujímavú) je približne 15% a výsledná špecificita (úspešnosť označenia adresy ako nezaujímavú) takmer 100%. V oblasti diskriminačných problémov je dôležité poznamenať, že sa musíme pozerať aj na senzitivitu a špecificitu a nie len na celkovú úspešnosť klasifikácie, jednoduchý príklad kedy by sme mali absolútne nepoužiteľný model s veľkou úspešnosťou by bol úbohý model, ktorý by označil každú adresu ako nezaujímavú. Takýto prípad však pre náš model nenastáva, má vysokú 100 % špecificitu a celkom signifikantnú senzitivitu. Zároveň vidíme, že ak model označil nejakú adresu ako zaujímavú, tak v oboch prípadoch sa naozaj jednalo o zaují-

mavú adresu. Model teda môžeme zhodnotiť ako praktický na predikciu typu adries na sieti Etherea. Na záver predpokladáme, že ak by sme sledovali transakcie omnoho dlhšie obdobie a z nich vytvorili dáta ako sú na Obr. 1, tak by sa strom (model) viacej rozvetvil a bol by možno schopný klasifikovať entitu aj podľa toho o akú konkrétnu entitu sa jedná ('Exchange','DeFi','Mining','Unknown'), pri takomto malom počte to však nebolo možné. Aj keď sme s modelom spokojní a vyzerá to, že nedošlo k pretrénovaniu, čo býva pri klasifikačných stromoch bežným prípadom pokúsime sa použiť metódu random forest. Random forest je metóda, ktorá funguje na princípe klasifikačných stromov. Veľmi zjednodušene sa vytvorí niekoľko nových bootstrapovaných datasetov z pôvodnej trénovanej sady dát (prakticky niekoľko stoviek). Nad každou takouto sadou sa vytvorí klasifikačný strom, pričom každý jeden klasifikačný strom participuje pri predikcií rovnako ako v predošlom príklade. Každý klasifikačný strom vyhodnotí pre nové dáto na vstupe jeho najviac pravdepodobnú klasifikáciu a potom prebieha takzvané hlasovanie stromov. Dáto dostane takú klasifikáciu, za ktorú hlasovala väčšina stromov. Vcelku bežnou situáciou je, že random forest poskytuje lepšie výsledky na testovacích dátach ako samotný klasifikačný strom, v čo teda dúfame aj v našom prípade. Jedine ako sa môže model zlepšiť je, že začne označovať viacej adries ako významných.

Tabuľka 3: Kontingenčná tabuľka, zobrazujúca úspechy a neuspéchy klasifikácie modelu - random forest na testovacich dátach.

Aktuálne / Predikované	1	0
1	4	9
0	0	3002

V Tabuľke 3 môžeme naozaj pozorovať zlepšenie senzitivity a to z 15% na 30%. Zároveň o čosi viacej veríme modelu, že ak nejakú adresu zaklasifikuje ako zaujímavú, tak ňou naozaj je.

Záver

Na záver by sme zhrnuli, že cieľom projektu bola analýza zhlukov a klasifikácia transakčných dát na sieti Etherea. V prvej časti projektu sme sa venovali čisto dátam, akého sú charakteru, aké majú atribúty, ako sme ich dostali a ukázali si nejaké základne štatistiky. V ďalších dvoch častiach sme vyskúšali použitie zhlukovacich algoritmov, jeden ktorý tvorí kruhovité zhluky a skôr zaklasifikoval dáta na základe signifikantnosti hodnôt premenných, zatiaľ čo druhy algoritmus tvorí zhluky rôznych tvarov a vo výsledku poslúžil dobre na vytvorenie zhlukoch pre menej signifikantné adresy. Na konci všetkého sme vytvorili dva predikčné modely na predikciu premennej entity na základe sieťovej aktivity adresy s využitím klasifikačných stromov.

Zoznam použitej literatúry

[1] Použité dáta k čiastočnému labelovaniu https://www.kaggle.com/datasets/hamishhall/labelled-ethereum-addresses

[2] Redukcia dimenzie blockchainových dát pre sieť Etherea https://github.com/devAdam117/Data-dimension-reduction

[3] Všetko ohľadom projektu tj. R skript, dáta, TypeScript, projekt https://github.com/devAdam117/clusterAnalysis

Prílohy