

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

**RAZVOJ PROGRAMSKE OPREME ZA ANALIZO JAVNEGA
MNENJA NA PODLAGI OBJAV UPORABNIKOV PLATFORME
TWITTER NA PRIMERU VOJNE V UKRAJINI**

Ljubljana, junij 2023

MARTIN CORETTI

IZJAVA O AVTORSTVU

Podpisani Martin Coretti, študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Razvoj programske opreme za analizo javnega mnenja na podlagi objav uporabnikov platforme Twitter na primeru vojne v Ukrajini, pripravljenega v sodelovanju s svetovalcem doc. dr. Luko Tomatom

I Z J A V L J A M

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;
7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi;
11. da sem preveril verodostojnost informacij, ki izhajajo iz zapisov na podlagi uporabe orodij umetne inteligence.

V Ljubljani, dne _____

Podpis študenta: _____

KAZALO

UVOD.....	1
1 VOJNA V UKRAJINI IN NJENE POSLEDICE	3
1.1 Začetek in potek spopadov	3
1.2 Odziv Zahoda	5
1.3 Sankcije EU proti Rusiji.....	6
2 RAZVOJ PROGRAMSKE OPREME	8
2.1 Življenjski cikel razvoja programske opreme.....	8
2.2 Novejše metodologije razvoja programske opreme	10
2.2.1 Agilna metodologija	10
2.2.2 Vitka metodologija	12
2.2.3 Metodologija hitrega razvoja aplikacij	13
2.2.4 Metodologija funkcionalno vodenega razvoja.....	14
3 PREDSTAVITEV UPORABLJENIH TEHNOLOGIJ IN METOD.....	15
3.1 Programski jezik in moduli.....	15
3.1.1 Programski jezik Python.....	15
3.1.2 Modul Pandas	16
3.1.3 Modul Snsrape	16
3.1.4 Modul Transformers	17
3.1.5 Modul Scipy.....	17
3.1.6 Modul Numpy.....	17
3.1.7 Modul Matplotlib.....	18
3.1.8 Modul NLTK	18
3.1.9 Modul Wordcloud.....	19
3.1.10 Modul Re	19
3.1.11 Modul Geopy	19
3.1.12 Modul Plotly Express.....	20
3.1.13 Modul Datetime	20
3.2 Razvojno okolje	21
3.3 Analiza razpoložanja.....	22
4 PREDSTAVITEV PODATKOV	24
4.1 Predstavitev vira in format podatkov	24

4.2	Analiza kakovosti podatkov.....	28
5	RAZVOJ PROGRAMSKE OPREME.....	30
5.1	Pridobivanje in shranjevanje objav na platformi Twitter.....	30
5.2	Formatiranje podatkov	33
5.3	Analiza razpoloženja objav.....	35
5.4	Razvoj skripte za grafični prikaz rezultatov.....	38
6	ANALIZA REZULTATOV	43
6.1	Predstavitev rezultatov.....	44
6.2	Ključne ugotovitve.....	56
6.3	Omejitve raziskave in možnosti za nadaljnje delo	59
	SKLEP.....	61
	LITERATURA IN VIRI.....	63
	PRILOGE	75

KAZALO SLIK

Slika 1:	Delež negativnih, nevtralnih in pozitivnih tvitov	44
Slika 2:	Besedni oblak najbolj pogosto uporabljenih besed.....	45
Slika 3:	Besedni oblak najbolj pogosto uporabljenih ključnikov.....	45
Slika 4:	Besedni oblak najbolj pogosto omenjenih uporabniških računov	46
Slika 5:	Število objavljenih tvitov glede na dan objave	46
Slika 6:	Število objavljenih tvitov glede na uro objave	47
Slika 7:	Število objav glede na njihovo razpoloženje in dan objave.....	48
Slika 8:	Razporeditev dolžine objav glede na njihovo razpoloženje	48
Slika 9:	Razporeditev števila komentarjev objav glede na njihovo razpoloženje (levo brez osamelcev, desno z osamelci)	49
Slika 10:	Razporeditev števila delitev objav glede na njihovo razpoloženje (levo brez osamelcev, desno z osamelci)	50
Slika 11:	Razporeditev števila všečkov objav glede na njihovo razpoloženje.....	50
Slika 12:	Razporeditev števila ogledov objav glede na njihovo razpoloženje	51
Slika 13:	Razporeditev števila sledilcev avtorjev objav glede na njihovo razpoloženje	51
Slika 14:	Razporeditev števila sledenj avtorjev objav glede na njihovo razpoloženje ...	52
Slika 15:	Razporeditev števila objav avtorjev glede na njihovo razpoloženje.....	53
Slika 16:	Delež verificiranih avtorjev objav glede na njihovo razpoloženje	53
Slika 17:	Seštevek vseh komentarjev, delitev in všečkov glede na razpoloženje objav	54
Slika 18:	Pogostost uporabe besed “war” in “special military operation” znotraj objav glede na njihov datum objave.....	55
Slika 19:	Število uporabniških računov glede na leto njihovega odprtja	55

Slika 20: Zemljevid razpoloženja	56
--	----

KAZALO PRILOG

Priloga 1: Primeri kod za generiranje grafikonov	1
--	---

SEZNAM KRATIC

angl. – angleško

BPE – (angl. byte-pair encoding); kodiranje parov bajtov

EU – (angl. European Union); Evropska unija

FDD – (angl. feature-driven development); funkcionalno voden razvoj

NATO – (angl. North Atlantic Treaty Organization); Severnoatlantska pogodbeni organizacija

NLTK – (angl. natural language toolkit); modul v programskem jeziku Python, ki vključuje orodja in funkcije za obdelavo naravnega jezika

RAD – (angl. rapid application development); hiter razvoj aplikacij

SDLC – (angl. software development life cycle); življenjski cikel razvoja programske opreme

SWIFT – (angl. Society for Worldwide Interbank Financial Telecommunications); od 8 do 11 znakov dolga standardna koda, ki označuje banke in finančne institucije po vsem svetu

t. i. – tako imenovan

ZDA – Združene države Amerike

UVOD

Pred dobrim letom se je pričela ruska invazija na Ukrajino. Enote oboroženih sil Ruske federacije so 24. februarja 2022 nenapovedano prečkale takrat veljavno mejo med državama iz severa (iz Belorusije proti Kijevu), vzhoda (iz Belgoroda v smeri mest Sumi in Harkov), jugovzhoda (iz Rostovske oblasti proti Donecku in Lugansku) ter iz juga (iz Krimskega polotoka proti reki Dneper) (Ministry of Defence, 2022), mednarodno priznana mejo pa so kršile že leta 2014 s priključitvijo polotoka Krim (Bebler, 2015).

Ministrstva za obrambo in zunanje zadeve obeh vpletenih strani, zahodne obveščevalne agencije, NATO, mediji idr. redno objavljajo novice o stanju na ukrajinskih bojiščih, v današnjih časih pa se te novice objavljajo in posledično tudi širijo po spletu. Twitter je eno izmed novih bojišč informacijske vojne, zato je mogoče na tej platformi zaslediti veliko število časopisnih člankov, vojnih poročil in različne propagande. Njegovi uporabniki se vključujejo v operacije objavljanja in širjenja vojne propagande ene in druge strani, v preteklih letih pa so takšne platforme vojske iz vsega sveta uporabljale za kampanje odnosov z javnostmi (Ojala, Pantti & Kangas, 2018). Jasno je, da so torej uporabniki te platforme podvrženi propagandam obeh strani, na podlagi prebranega pa si ustvarijo mnenje, ki ga potem tudi objavijo na platformi v obliki tvita (Garcia & Cunanan-Yabut, 2022).

Mnenja uporabnikov platforme Twitter je mogoče analizirati na več načinov. Takšno analizo bi lahko opravili ročno, a bi se zelo kmalu izkazalo, da je to zelo zamuden proces, še posebej glede na obseg števila objav na to temo, saj gre za milijone tvitov od začetka spopadov (Twitter Archive, 2022), in glede na dejstvo, da so objave napisane v različnih jezikih. Proces analize objav pa je mogoče avtomatizirati z nekaterimi programskimi orodji, jeziki in okolji, ki podpirajo izvedbo ti. analize vsebine (angl. content analysis) in analize razpoloženja (angl. sentiment analysis) (Godnov & Redek, 2014).

Analiza razpoloženja je proces avtomatizirane obdelave naravnega jezika (angl. natural language processing). Gre za analizo vsebine na ravni dokumenta, v zadnjem času z vzponom družbenih omrežij in posledično krajšega obsega besedil in sporočil, pa vedno bolj na ravni posameznih povedi (Agarwal, Xie, Vovsha, Rambow & Passonneau, 2011). Gre za postopek, s katerim izpeljemo razpoloženje določene izjave ali stavka. Analiza razpoloženja je torej tehnika razvrščanja in izpeljave mnenja in oblikovanja stališča (Gupta in drugi, 2017). Obstaja več metod za izdelavo analize razpoloženja, ki se med seboj precej razlikujejo, saj ne upoštevajo enakomerno specifik medijev, na katerih se besedilo nahaja (Gonçalves, Araújo, Benevenuto & Cha, 2013). Najbolj primerna metoda, ki je zmožna upoštevati besedila, krajša od maksimalnih 280 znakov za tvit (Twitter, 2023), trenutno pomembnost oz. popularnost teme, slovnične napake, prisotnost znakov, kot so # ali @ idr., je zato analiza razpoloženja (Giachanou & Crestani, 2016).

Namen magistrskega dela je predstaviti, kako poteka razvoj programske opreme, ki omogoča prebiranje objav uporabnikov in pripadajočih metapodatkov na platformi

Twitter, ter predstaviti analizo javnega mnenja na podlagi zbranih podatkov. Rezultati magistrskega dela lahko služijo raziskovalnim inštitutom in organizacijam, ki se ukvarjajo z izbrano temo, pa tudi splošni javnosti, da bo lahko bolje razumela pogled ljudi na dogajanje v Ukrajini.

Cilji magistrskega dela so:

- na konkretnem primeru prikazati, kako poteka razvoj programske opreme za prebiranje podatkov s spletnih strani
- na realnih podatkih prikazati, kako poteka izdelava analize razporeditve
- izdelati analizo javnega mnenja o prvem letu vojne v Ukrajini na podlagi objav uporabnikov platforme Twitter
- podati priporočila za boljše razumevanje in interpretacijo javnega mnenja o vojni v Ukrajini organizacijam, ki jih obravnavana tematika zadeva.

Magistrsko delo je razdeljeno na teoretičen in praktičen del. Najprej je v teoretičnem delu predstavljena splošna analiza vojne v Ukrajini in njene implikacije na Zahod. Fokus analize stanja v Ukrajini je na bojiščih in teritorialnih spremembah, analiza stanja na Zahodu pa je osredotočena na sankcije in inflacijo ter vplive, ki jih imata na gospodarstvo. Ob tem je poseben poudarek dan javnemu mnenju in podpori, ki jo prebivalci posameznih držav Evropske unije in ZDA dajejo Ukrajini v obliki humanitarne, ekonomske in vojaške pomoči. Pri tem sem večinoma uporabil članke iz priznanih in uglednih medijev, ustanov in organizacij, kot so npr. Ministrstvo za obrambo Združenega kraljestva, BBC, Bloomberg, Financial Times in drugi. Predstavil sem tudi spletno platformo Twitter kot primarni vir podatkov oz. mnenj o vojni v Ukrajini. Teorija razvoja programske opreme temelji predvsem na strokovni in znanstveni domači in tuji literaturi, sledi pa analiza programskega jezika Python in pripadajočih modulov, razvojnih okolij, metodologij razvoja programske opreme, predstavitev analize razporeditve ter metode pridobivanja in razčlenjevanja podatkov iz spleta.

V praktičnem delu sem razvil in po korakih predstavil razvoj programske opreme, ki omogoča pridobivanje podatkov o objavah uporabnikov na platformi Twitter in njihovo analizo. Razvoj programske opreme je potekal na podlagi teoretičnega dela o razvoju programske opreme ter znanja, ki sem ga pridobil v okviru dodiplomskega študija na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Sproti sem opisal težave, s katerimi sem se pri razvoju programske opreme soočal, predstavil pa sem tudi njihove rešitve.

Na koncu sem v analizi rezultatov na podlagi sinteze ugotovitev teoretičnega in praktičnega dela predstavil ključne ugotovitve magistrskega dela. Sledi še poglavje, v katerem omenjam omejitve raziskave, hkrati pa navedem tudi smernice za nadaljnji razvoj programske opreme.

1 VOJNA V UKRAJINI IN NJENE POSLEDICE

1.1 Začetek in potek spopadov

Konec leta 2021 so ameriške obveščevalne agencije izdale poročilo o morebitnih pripravah ruske vojske na invazijo Ukrajine. ZDA so s svojimi evropskimi zaveznicami delile podatke o kopičenju enot ruske vojske na Krimu, ob rusko-ukrajinski meji ter v Belorusiji. V informacijah je bil opisan scenarij, po katerem bi od 100.000 do 200.000 ruskih vojakov pričelo operacijo zasedbe strateških točk Ukrajine: vladna poslopja, televizijski in radio oddajniki, mostovi, logistični centri, pristanišča, letališča, železniške postaje, vojašnice in bolnišnice (Nardelli & Jacobs, 2021). Rusko obrambno ministrstvo je zanikalo trditev, da se Rusija pripravlja na napad na Ukrajino, in prisotnost velikega števila ruskih vojakov v Belorusiji pripisalo skupnim vojaškim vajam z beloruskimi oboroženimi silami (Kelly, 2022).

Skupne vojaške vaje v Belorusiji so se zaradi stopnjevanja napetosti večkrat podaljšale in tako nadaljevale tudi v januarju in februarju, v drugi polovici februarja 2022 pa je ameriška obveščevalna agencija sporočila, da so ruske enote prejele ukaz za sprožitev invazije (Pilkington, 2022). Rusija je še naprej zanikala namen napada, 21. februarja 2022 pa je ruski predsednik Vladimir Putin podpisal odlok, s katerim je priznal Ljudsko republiko Doneck in Ljudsko republiko Lugansk ter za zagotavljanje varnosti novopriznanih republik ukazal premik ruske vojske na njuno območje (Meyer, Champion & Halpin, 2022). V zgodnjih jutranjih urah 24. februarja 2022 je ruski predsednik Vladimir Putin nagovoril domačo in tujo javnost v televizijskem nagovoru, v katerem je sporočil začetek t. i. posebne vojaške operacije za demilitarizacijo in denacifikacijo Ukrajine. V tistih trenutkih so enote ruske vojske začele invazijo na Ukrajino (Osborn & Nikolskaya, 2022).

Rusija je Ukrajino napadla brez vojne napovedi iz treh smeri: iz Belorusije v smer ukrajinskega glavnega mesta Kijev, iz vzhoda v smer Harkovške, Doneške in Luganske oblasti ter iz Krima v smer mest Herson in Odesa, rusko letalstvo pa je medtem bombardiralo tarče po celotnem ozemlju Ukrajine (BBC, 2022).

V prvi fazi invazije je ruska vojska nameravala zavzeti ukrajinsko prestolnico Kijev, za doseg tega cilja pa si je morala najprej zagotoviti zadostno logistično podporo. Ruske padalske enote so zato v prvem dnevu vojne izvedle desant na letališče Antonov v bližini mesta Gostomel in ga tudi za kratek čas zavzele (McGregor, 2022), na vzhodu so prodirale proti mestoma Harkov in Sumi (Khurshudyan & Berger, 2022), na jugu pa so prodrle vse do jeza Nova Kahovka ob reki Dniper (The Kyiv Independent, 2022). Jez je strateško pomemben za obe strani, saj dovaja večino pitne vode Krimskemu polotoku (Reuters, 2022a). V naslednjih tednih so ruske enote nadaljevale z obleganjem Harkova in Kijeva, medtem so zasedle mesto Herson in se približevale Odesi (Schwartz & Pérez-

Peña, 2022). Iz Krima so ruske enote prodrle v smer mesta Mariupol, a ga niso uspele zavzeti. Konec marca so ukrajinske enote uspešno izvedle protinapad v okolici Kijeva in osvobodile nekaj manjših mest, v prvih dneh aprila pa so se ruske enote dokončno umaknile iz severa in severovzhoda Ukrajine (Institute for the Study of War, 2022).

V drugi fazi invazije so ruske enote nadaljevale z obleganjem Harkova, a ga še vedno niso uspele zavzeti (McCann, Gamio, Lu & Robles, 2022). V smeri Harkova so se ruske enote počasi približevale tudi iz Luganske oblasti, večina napora ruske vojske pa je bila usmerjena v zavzetje mesta Mariupol ob Azovskem morju. Enote ukrajinske nacionalne garde so skušale mesto ubraniti pred ruskim napredovanjem, a so bile izolirane od vseh ostalih ukrajinskih enot, saj je bilo mesto obkoljeno. Po obleganju, ki je trajalo več kot 2 meseca in zahtevalo več tisoč ali deset tisoč življenj na obeh straneh, so ruske enote v drugi polovici maja 2022 dokončno zavzele jeklarno Azovstal in s tem končale obleganje mesta (Pannett, Thebault, Dixon, Stern & Harris, 2022).

V tretji fazi invazije je napredovala ukrajinska stran. V okolici Harkova je ukrajinska vojska septembra 2022 uspešno izvedla protinapad in prišla do mednarodne ukrajinsko ruske meje. S tem se je obleganje Harkova zaključilo, ukrajinske enote pa so osvobodile še Izjum in Kupjansk z okolico (Bachega & Murphy, 2022). Zaradi niza uspehov ukrajinske vojske in porazov ruske je ruski predsednik Vladimir Putin konec septembra 2022 izdal odlok o delni mobilizaciji ruske vojske in vpoklicu 300.000 rezervistov (Reuters, 2022b). Teden pozneje so bile štiri ukrajinske regije, to so oblasti Lugansk, Doneck, Zaporožje in Herson, po izvedenem referendumu na teh območjih, tudi formalno priključene Ruski federaciji (Dickson & Morrow, 2022). Tudi iz tega razloga so ukrajinske enote na jugu začele protinapad v smeri mesta Herson. Zaradi vse večjega ukrajinskega pritiska in izoliranja ruskih enot na desnem bregu reke Dnpr so se bile ruske enote v Hersonu primorane umakniti na levi breg reke, s tem pa so Ukrajinci v začetku novembra 2022 dokončno osvobodili mesto Herson in znova zavzeli jez Nova Kahovka (Demianyk, 2022).

V četrti fazi invazije, ki so ga zaznamovali ruski raketni napadi na ukrajinsko energetska in drugo kritično infrastrukturo (Santora, Mpoke Bigg & Nechepurenko, 2022), so se boji znova premaknili na vzhod Ukrajine. Trenutno potekajo hudi boji v predmestju mesta Bahmut, saj so ruske enote od začetka decembra 2022 do danes zavzele mesta Soledar, Pidhorodne, Klišivka in Krasna Hora, trenutno pa se približujejo mestu Ivanivske, skozi katero je speljana avtocesta, ki predstavlja še edino povezavo Bahmuta z zaledjem. Ruske enote so namreč mesto obkolile in pretrgale vse ostale vezi in vire pomoči ukrajinskim vojakom v mestu. Ruske enote so v začetku leta 2023 poskušale napredovati tudi v smeri mesta Zaporožje na jugu in Kupjansk na severovzhodu, a niso uspele prodreti (MilitaryLand, 2023).

Rusija trenutno nadzoruje 104.000 kvadratnih kilometrov oz. 17 % celotnega ukrajinskega ozemlja, a ne brez izgub (Gutiérrez & Kirk, 2023). Obe strani sta precej zadržani glede lastnih izgub, izgube sovražnika pa po navadi precej napihujejo, zato je zanesljive podatke težko najti in preveriti. Po besedah vrhovnega poveljnika norveških oboroženih sil je v prvem letu dni invazije Rusija izgubila približno 180.000 mož, Ukrajina več kot 100.000, civilnih žrtev pa je več kot 30.000 (AFP, 2023). Rusija naj bi izgubila 300 letal, skoraj toliko helikopterjev, več kot 3.000 tankov in 6.500 drugih oklepnih vozil, 2.300 havbic, 2.000 dronov ter 18 ladij. Skupna vrednost izgubljene opreme naj bi na ruski strani presegala 17 milijard ameriških dolarjev (UAWar, 2023).

1.2 Odziv Zahoda

Med letoma 2014 in 2021 so Združeno kraljestvo ter ZDA in ostale NATO članice Ukrajini zagotavljale nesmrtonosno vojaško pomoč, predvsem v obliki urjenja enot oboroženih sil Ukrajine (Mills, 2022). Ko je Rusija konec leta 2021 na ukrajinskih mejah začela kopičiti opremo in vojake, so ZDA skupaj z drugimi članicami NATO Ukrajini začele posredovati protitankovsko orožje, predvsem sisteme NLAW in Javelin (Mills & Curtis, 2022). Po invaziji so se države članice NATO, ki so bile do takrat precej zadržane glede pošiljanja vojaške opreme in streliva, vključno z Nemčijo, strinjale z dobavo protitankovskega orožja, čelad, neprebojnih jopičev, streliva in ročnih bomb.

Dobave orožja so se skozi mesece stopnjevale. Na začetku so jih sestavljale predvsem avtomatsko orožje, strelivo ter protitankovsko in protiletalsko orožje, nato havbice in raketni sistemi, v času ruskih raketnih napadov na ukrajinsko kritično infrastrukturo protizračni sistemi, trenutno pa tanki. Skupna vrednost orožja dobavljenega Ukrajini naj bi presegala 65 milijard evrov, finančna in humanitarna pomoč pa 91 milijard evrov (Antezza in drugi, 2023).

Največ pomoči, predvsem vojaške, je Ukrajina prejela od ZDA, sledi jim EU in Združeno kraljestvo. ZDA so Ukrajini namenile več kot dve tretjini vse prejete vojaške pomoči, predvsem v obliki havbic M777, raketnih sistemov HIMARS, oklepnih vozil Stryker in Bradley, tankov Abrams, raket srednje-dolgega dosega in streliva (Masters & Merrow, 2023). Tudi Slovenija je prispevala pomembno humanitarno, a predvsem vojaško opremo. Kmalu po začetku spopadov je Slovenija Ukrajini dostavila čelade, polavtomatske puške jugoslovanske izdelave in strelivo zanje (STA, 2022), konec junija 2022 pa 35 goseničnih bojnih vozil jugoslovanske izdelave M-80A (Knific, 2022). Septembra istega leta je Slovenija v Ukrajino poslala še 28 bojnih tankov M-55S, ki jih je Slovenska vojska nadgradila leta 2001 iz starejših sovjetskih tankov T-55, od takrat pa so bili hranjeni v skladiščih in nikoli dejansko uporabljeni (P., 2001). V zameno za dobavo tankov je Slovenija od Nemčije prejela 40 težkih transportnih vozil (Cerar, 2022).

1.3 Sankcije EU proti Rusiji

Zaradi nezakonite priključitve Krima, vojaške in ekonomske podpore separatističnim samooklicanim republikam v Donbasu, priznanja in priključitve samooklicanih republik v Rusko federacijo ter invazije na Ukrajino z novo nezakonito priključitvijo Zapororoške in Hersonske oblasti, je EU uvedla več svežnjev sankcij proti Ruski federaciji (European Council, 2022).

Prvotne sankcije so stopile v veljavo že marca 2014, kmalu po ruski zasedbi Krimskega polotoka (Council of the EU, 2014d). Sestavljene so bile iz zasedbe premoženja in prepovedi potovanja znotraj EU posameznikom političnega in vojaškega vrha Ruske federacije, konec julija istega leta pa je bil sprejet nov sveženj strožjih sankcij, ki je vključeval omejitev dostopa ruskih finančnih institucij v državni lasti do kapitalskih trgov EU, uvedbo embarga na trgovino z orožjem, prepoved izvoza blaga za vojaške namene in omejitev ruskega dostopa do občutljivih tehnologij, zlasti na področju naftne industrije (Council of the EU, 2014c).

Septembra 2014 so se sankcije še zaostrole. Tokrat je bilo na vrsti pet večjih ruskih bank v državni lasti, tri večja ruska obrambna podjetja in tri večja energetska podjetja, prepovedane so bile tudi nekatere dejavnosti, potrebne za raziskovanje in pridobivanje nafte v globokih vodah, povečal se je tudi seznam oseb, ki jim je bila izrečena prepoved vstopa in odobrena zamrznitev premoženja v EU (Council of the EU, 2014a). Skupno je bilo do konca leta na seznamu 132 oseb s prepovedjo vstopa v EU, osemindvajsetim pa so bila zamrznjena sredstva (Council of the EU, 2014b).

Skozi leta so se te sankcije še delno stopnjevale, predvsem na račun daljšanja seznama oseb s prepovedjo vstopa v EU in zamrznitvijo njihovega premoženja znotraj EU. Prvi novi večji sveženj sankcij je EU sprejela 23. februarja 2022, na dan, ko je ruski parlament Duma priznal ti. samooklicani Ljudsko republiko Doneck in Ljudsko republiko Lugansk. Sankcije so bile usmerjene proti 351 članom Dume, vključevale pa so tudi omejitev gospodarskih odnosov s tema dvema separatističnima republikama (Council of the EU, 2022c).

Hujše sankcije pa je EU sprejela po 24. februarju 2022, ko so ruske enote začele invazijo na Ukrajino brez vojne napovedi. Prvemu svežnju sankcij je sledil drugi sveženj 25. februarja, vključeval je prepoved vstopa in zamrznitev premoženja v EU Vladimirja Putina, predsednika Ruske federacije, in Sergeja Lavrova, ministra za zunanje zadeve Ruske federacije. Tarča drugega svežnja sankcij so bili tudi vsi ostali predstavniki ruskega Sveta za nacionalno varnost in Dume, ki so podprli rusko priznanje samooklicanih republik (Council of the EU, 2022k).

Tri dni pozneje, 28. februarja, je sledil tretji sveženj sankcij, ki je ukinil vse transakcije z rusko centralno banko, prepovedal vstop ruskim letalskim prevoznikom v zračni prostor EU, obenem pa nudil ukrajinskim oboroženim silam pol milijarde evrov za financiranje opreme in oskrbe enot na bojišču (Council of the EU, 2022b). Par dni pozneje je bila kot del tega svežnja uveljavljena izključitev sedmih ruskih bank iz sistema SWIFT (Council of the EU, 2022j), še isti dan pa je EU prepovedala oddajanja znotraj EU ruskima medijema Sputnik in Russia Today (Council of the EU, 2022e). V sledečem tednu je EU razširila finančne sankcije še na belorusko centralno banko in tri druge beloruske banke (Council of the EU, 2022i), obenem pa uvedla še sankcije proti štirinajstim ruskim oligarhom (Council of the EU, 2022d).

Četrty sveženj sankcij, ki je bil sprejet 15. marca 2022, je vključeval prepoved novih naložb v ruski energetski sektor, obenem pa je omejil trgovanje z železom, jeklom in luksuznim blagom (Council of the EU, 2022l). Zaradi nadaljevanja ruskih napadov in prvih novic o pokolu civilistov, ki so ga izvedle ruske sile v ukrajinskem mestu Buča severozahodno od Kijeva, je EU 8. aprila 2022 sprejela peti sveženj sankcij, ki je vključeval prepoved uvoza premoga in drugih trdnih fosilnih goriv iz Rusije, prepoved dostopa vseh ruskih plovil v pristanišča EU, prepoved vstopa v EU ruskim in beloruskim cestnim prevoznikom, prepoved izvoza letalskega goriva ter prepoved depozitiranja sredstev ruskih državljanov v kriptodenarnice (Council of the EU, 2022f).

Šesti sveženj sankcij, sprejet 3. junija 2022, je vključeval prepoved uvoza surove nafte in rafiniranih naftnih derivatov iz Rusije (z določenimi izjemami), izključitev iz sistema SWIFT za tri dodatne ruske in eno belorusko banko ter prepoved oddajanja trem dodatnim ruskim medijem v državni lasti (Council of the EU, 2022h). S sedmim svežnjem sankcij, ki je bil sprejet 21. julija 2022, je bila uvedena prepoved nakupa, uvoza ali prenosa zlata ruskega porekla (Council of the EU, 2022g), z osmim svežnjem, sprejetim 6. oktobra 2022, pa je bila določena omejitev cen pomorskega prevoza ruske nafte za tretje države, hkrati pa se je še dodatno omejilo blago, ki lahko prispeva k vojaški in tehnološki krepitvi Rusije (Council of the EU, 2022a).

Pred sprejetjem devetega svežnja sankcij, je EU sprejela še zgornjo mejo cene za surovo nafto in naftna olja ter olja, pridobljena iz bituminoznih mineralov, ki izvirajo iz Rusije ali se iz nje izvažajo, v višini 60 USD na sod (Council of the EU, 2022n). Trenutno zadnji sprejeti deveti sveženj sankcij, ki je stopil v veljavo 16. decembra 2022 po ruskih napadih na ukrajinsko civilno energetsko infrastrukturo, pa vključuje prepoved izvoza motorjev za brezpilotna letala, prepoved naložb v ruski rudarski sektor, navsezadnje pa tudi prepoved opravljanja storitev oglaševanja, tržnih raziskav in javnomnenjskih raziskav znotraj EU (Council of the EU, 2022m).

Ruska federacija je v manj kot dveh tednih postala država proti kateri je bilo uvedeno največje število sankcij, saj je presegla Islamsko republiko Iran, Sirsko arabsko republiko

in Demokratično ljudsko republiko Korejo (Wadhams, 2022). V trenutku pisanja tega poglavja je proti Rusiji uvedenih 13.596 sankcij, proti Iranu, Siriji, Severni Koreji, Belorusiji, Mjanmaru in Venezueli pa skupno “le” 11.449 (Statista Research Department, 2023).

2 RAZVOJ PROGRAMSKE OPREME

2.1 Življenjski cikel razvoja programske opreme

Življenjski cikel razvoja programske opreme (angl. System Development Life Cycle, v nadaljevanju SDLC) je izraz, ki opredeljuje tradicionalen sistematičen pristop, ki se uporablja za razvoj in vzdrževanje zanesljive in visoko kakovostne programske opreme. Običajno vključuje različne faze, od predhodne analize razvoja do testiranja in vrednotenja programske opreme po razvoju. Upravljanje celotnega razvojnega procesa je zahtevno in naporno delo, saj je potrebno zagotoviti, da končni izdelek vsebuje visoko stopnjo celovitosti in robustnosti ter da ga stranka sprejme. Zato je sistematičen razvojni proces, ki poudarja razumevanje obsega in zapletenosti celotnega razvojnega procesa, bistven za doseganje omenjenih značilnosti uspešnega sistema.

SDLC je nastal v 60. letih prejšnjega stoletja v ameriških obrambnih in vesoljskih podjetjih, ki so gradila obsežne programske sisteme. Šlo je za večletne projekte programske opreme za velike računalniške sisteme, ki so se razvijali razmeroma počasi. Klasični model SDLC obravnava postopek razvoja programske opreme kot linearno in zaporedno vrsto faz, ki razvijalce vodijo vse od začetnih zahtev do končnega testiranja sistema in odpreme izdelka (Stoica, Ghilic-Micu, Mircea & Uscatu, 2016).

SDLC upošteva več faz razvoja. Najprej oblikovalci napišejo specifikacijo, ki mora biti čim bolj popolna. Nato specifikacijo razdelijo na dele oz. module, za vsak modul specificirajo še dodatne funkcionalnosti in podrobnosti, nato pa jih predajo različnim posameznikom ali skupinam v razvoj. Vsaka ekipa testira in odpravlja napake v svojem modulu. Šele v zadnji fazi projekta, ki lahko pri obsežnejši programski opremi traja od nekaj mesecev do nekaj let, poskušajo oblikovalci, razvijalci in testerji sestaviti vse dele skupaj in testirati programsko opremo kot celoto. Ta postopek integracije in testiranja sistema običajno zahteva preoblikovanje modulov in pisanje nove kode, da se odpravijo težave v delovanju ali interakcijah delov zaradi nepredvidenih težav ter napak, nesporazumov ali morebitnih sprememb (Shaydulin & Sybrandt, 2017).

SDLC se zato dokaj dobro obnese na stabilnih problemskih področjih, kjer je razvoj programske opreme sestavljen predvsem iz dodajanja postopnih sprememb obstoječemu jedru funkcionalnosti. Primeren je za projekte, pri katerih je malo ali nič vmešavanja strank v sam proces razvoja, ni pa primeren za razvoj programske opreme, ki vključuje

veliko nove vsebine, saj so v tem primeru spremembe med razvojem neizogibne (Dima & Maassen, 2018).

SDLC poudarja načrtovanje v zgodnjih fazah, zato pogosto odpravlja napake, še preden se te pojavijo. Poleg tega pa se zaradi intenzivnega dokumentiranja in načrtovanja dobro obnese pri projektih, pri katerih je nadzor kakovosti glavna skrb. Sestavljen je iz več faz, ki se nikoli ne prekrivajo (Alshamrani & Bahattab, 2015):

1. **Zahteve:** opis obnašanja sistema, ki ga je treba razviti. Običajno so to informacije, ki jih posredujejo odjemalci. V tej fazi se vzpostavlja dogovor med strankami in razvijalci o specifikacijah in značilnostih programske opreme. Zahteve se zberejo, analizirajo in nato se pripravi ustrezna dokumentacija, ki pomaga pri nadaljnjem razvoju.
2. **Načrtovanje na visoki ravni:** ovrednotijo se informacije, zbrane v prejšnji fazi, oblikuje se ustrezna izvedba. Gre za postopek načrtovanja in reševanja problemov za programsko rešitev. Ukvarja se z izbiro ustrezne zasnove algoritmov, zasnovo arhitekture programske opreme, konceptualno shemo zbirke podatkov, zasnovo logičnega diagrama in opredelitvijo podatkovne strukture.
3. **Kodiranje:** v tej fazi se v produkcijskem okolju prične razvoj programske opreme po predhodno zbranih zahtevah in določenem načrtu.
4. **Preizkušanje:** dejansko testiranje in preverjanje programskih rešitev, ki so bile razvite za izpolnitev prvotnih zahtev. To je faza, v kateri se najdejo, odpravijo in izpopolnijo napake in sistemske pomanjkljivosti.
5. **Vzdrževanje:** po tem, ko je programska oprema že izdana, lahko pride tudi do sprememb, popravkov novih napak in ustreznih izboljšav. Za te težave se poskrbi v tej fazi.

Osnovno načelo SDLC je razdeljenost projekta na zaporedne faze, pri čemer se faze med seboj ne prekrivajo. Poudarek je zato na načrtovanju, časovnih razporedih, ciljnih datumih, proračunu in implementaciji sistema v celoti. V času trajanja projekta se ohranja strog nadzor z obsežno pisno dokumentacijo, formalnimi pregledi ter odobritvijo s strani stranke in vodstva projekta razvoja, ki se zgodi na koncu vsake faze oz. pred začetkom naslednje (Balaji & Murugaiyan, 2012).

SDLC je najprimernejši takrat, ko so zahteve za programsko opremo dobro opredeljene, dokumentirane, razjasnjene in določene, ko je uporabljena tehnologija razumljiva in ni dinamična in ko so na voljo vsi potrebni viri.

Prednosti SDLC so enostavnost razumevanja in uporabe, vidnost rezultatov vsake faze, zaporednost faz, jasnost opredeljenih zahtev, primernost za manjše projekte, enostavnost razporejanja nalog ter podrobna dokumentacija vsake faze in celotnega procesa (Alshamrani & Bahattab, 2015).

Slabosti SDLC so pozen nastop programske opreme med življenjskim ciklom razvoja, veliko število tveganj, neprimernost za velike projekte, nefleksibilnost glede spreminjanja zahtev, težavna meritev napredka med fazami ter odpornost na prilagajanje obsega projekta (Shylesh, 2017).

2.2 Novejše metodologije razvoja programske opreme

Razvoj programske opreme poteka po strukturiranih postopkih, ki se uporabljajo pri delu na določenem projektu, čemur pravimo model. Modeli razvoja programske opreme so torej skupki načel pri načrtovanju in uresničevanju projekta, prvi modeli pa so nastali že v začetku računalništva. Cilj modelov je običajno zagotoviti sistematičen pristop k razvoju programske opreme, ki se ga mora držati celotna ekipa razvijalcev, saj je to pogoj za učinkovitejše sodelovanje v ekipi, jasnejše komuniciranje in natančnejši prenos informacij (Jiang & Eberlein, 2009).

Skozi leta so nastale različne metodologije razvoja programske opreme, da bi izkoristili razpoložljive tehnologije in vire ter se prilagodili zahtevam uporabnikov programske opreme skozi čas. Ravno zato je vprašanje, katera metodologija je najboljša, še vedno brez pravega odgovora. Vsaka metodologija ima namreč svoje prednosti in slabosti. Največji vpliv na končni učinek uporabe določene metodologije pa je odvisen od strukture ekipe, managementa, zahtev in ciljev projekta razvoja programske opreme (Berenbach, 2006).

V naslednjih podpoglavjih bom zato predstavil štiri metodologije razvoja programske opreme, ki spadajo v družino novejših agilnih metodologij.

2.2.1 Agilna metodologija

Agilna metodologija razvoja programske opreme je v zadnjih časih najbolj najpogostejše uporabljena tehnika razvoja programske opreme. Namesto dolgotrajnega razvijanja in izdajanja funkcionalnosti, agilna metodologija sloni na krajših, a rednih ciklih izdajanja novih funkcionalnosti. To strankam omogoča večjo vključenost med samim procesom razvoja programske opreme. To posledično pripomore k večji kakovosti končnega izdelka, saj odpravlja težavno nalogo, da stranka v celoti razume in opredeli vse zahteve že na samem začetku oz. v fazi načrtovanja projekta razvoja programske opreme (Matharu, Mishra, Singh & Upadhyay, 2015).

Agilna metodologija razvoja programske opreme se osredotoča na kratek iterativni cikel izdaje programske opreme, ki mu pravimo iteracija (angl. sprint), traja pa predvidoma od 1 do 4 tedne. Iteracije so zasnovane na intenzivnem vključevanju stranke in nenehnem prikazovanju trenutnega stanja programske opreme. Tako lahko stranka med aktivnim razvojem programske opreme daje priporočila ali predlaga spremembe, tako da

programska oprema sledi dejanskim željam stranke (Abrahamsson, Salo, Ronkainen & Warsta, 2002).

Metodologija agilnosti predlaga, da ekipe za razvoj programske opreme poskušajo velike zahtevne zahteve razdeliti na veliko majhnih in preprostih zahtev. Zato je pred vsako iteracijo potreben sestanek, da se določi, katere funkcije, izboljšave in popravke je treba izvesti v naslednjem ciklu. Med samo iteracijo pa prav tako potekajo dnevni sestanki, na katerih razvijalci razpravljajo o napredku in ugotavljajo ovire. To omogoča visoko transparentnost razvoja in komunikacije (Al-Saqqa, Sawalha & AbdelNabi, 2020).

Prednosti agilne metodologije so minimalno število napak zaradi ponavljajočega testiranja in prilagajanja programske opreme, malo nesporazumov med člani ekipe med razvojem zaradi pogostega pregleda razvoja, spremembe zahtev pa se običajno obravnavajo brez večjih zapletov in z majhnim vplivom na časovni načrt zaradi stalne vključenosti stranke v proces razvoja.

Slabosti agilne metodologije so morebitna preobremenjenost razvijalcev zaradi prepogostih zahtev po spremembah stranke, dokumentacija je v ozadju, kar lahko povzroča težave pri poznejših fazah razvoja, zaradi velikega števila sestankov pa lahko ekipi zmanjka časa za razvoj funkcionalnosti v predvideni iteraciji (Mirza & Datta, 2019).

Primer agilne metodologije je npr. metodologija gruč (angl. scrum), verjetno ena najbolj fleksibilnih metodologij za razvoj programske opreme, priljubljena pa je predvsem zaradi svojih postopnih in iterativnih pristopov. Metodologija gruč, poleg vodje projekta in razvijalcev, vključuje tudi strokovnjaka za metodologijo gruč. Vodja projekta pridobi potrebne informacije od stranke in zagotavlja, da je ekipa na pravi poti pri izpolnjevanju podanih zahtev, strokovnjak za gruče deluje kot moderator in zagotavlja, da so člani ekipe seznanjeni z vsemi postopki in procedurami, ki jih metodologija predpisuje, razvijalci pa prevzamejo odgovornost za razvoj funkcionalnosti.

Metodologija gruč je idealna metodologija za hiter razvoj programske opreme, saj se naloge izvajajo v iteracijah, ki trajajo največ 4 tedne. Hitra izvedba razvijalcem omogoča, da v kratkem času ugotovijo težave, uvedejo rešitve, testirajo in zberejo povratne informacije.

Strokovnjak za gruče organizira dnevni sestanek in njegovo lokacijo, na sestanku pa poskuša odpraviti vse ovire pri razvoju. Deluje kot koordinator med ekipo razvijalcev in vsemi drugimi vpletenimi oddelki. Strokovnjak za gruče tudi spremlja, kaj se dogaja na vsakodnevnih sestankih zato, da bi ocenil hitrost ekipe. Dnevni sestanek je običajno kratek, traja do največ 15 minut, na njem pa se razvijalci pogovorijo o tem, kaj je bilo storjeno od zadnjega sestanka, kaj bo storjeno do naslednjega sestanka in kakšne so trenutno prisotne ovire (Beedle, Devos, Sharon, Schwaber & Sutherland, 1999).

Poleg neformalnih dnevnih sestankov se običajno izvedeta tudi dva bolj formalna mesečna sestanka, na katerih se načrtuje iteracija. Razvijalci razdelijo eno obširno funkcionalnost v niz majhnih in obvladljivih nalog, ki se vnesejo v zbirko iteracij (angl. backlog). Ta vsebuje prednostni seznam vseh zahtev izdelka, ki jih določi vodja projekta. Na podlagi ocenjenega časa za razvoj posamezne naloge se nabor vseh nalog za določeno iteracijo določi tako, da jih je mogoče dokončati v enem mesecu. Na ta način lahko razvijalci z dokončanjem manjših nalog zlahka vidijo napredek. Glavni cilj drugega mesečnega sestanka pa je preveriti, kaj je bilo narejeno, katere stvari je treba izboljšati in katere stvari je ekipa naredila dobro (Hossain, Babar & Paik, 2009).

Uspeh razvoja programske opreme z metodologijo gruč je v veliki meri odvisen od tega, kako je sestavljena posamezna ekipa razvijalcev. Metodologija prepušča veliko svobode razvijalcem, da vodijo ekipo in rešujejo naloge, kot si sami zamislijo. Razvijalcem je običajno všeč zamisel, da se ekipa odloča o tem, kako bo delala stvari, na podlagi soglasja ekipe, in da ima ekipa večji nadzor nad tem, kako se razvoj vodi in zaključi. Na splošno razvijalcem omogoča boljše timsko delo in boljšo komunikacijo, ki se odraža v izdelkih višje kakovosti.

V metodologiji gruč se prioriteta in dodelitev nalog razvijalcev dinamično spreminja glede na napredek članov ekipe pri izvajanju nalog in poslovnih zahtev. Za metodologijo pa je značilno tudi iterativno in stalno spremljanje napredka projektov. Vsakodnevni sestanki in sestanki za pregled iteracij pomagajo razvijalcem iterativno in neprekinjeno spremljati napredek projektov, pri nalogah z visoko prioriteto pa dnevni sestanki pomagajo razvijalcem, da ostanejo pri nalogah, in jih opomnijo, kaj je treba storiti vsak dan (Cho, 2009).

Prednosti metodologije gruč so enostavna vključitev novih funkcionalnosti ali sprememb, obvladljivost sprememb brez večjih stroškov ponovnega načrtovanja, zadovoljstvo stranke ter stalno sodelovanje razvojne ekipe.

Slabosti metodologije gruč so pogoste težave v razvojni ekipi zaradi nerazumevanja, nesodelovanja ali odlaganja obveznosti na druge člane ekipe, pogostost dvomljivih časovnih okvirov razvoja ter pomanjkanje obširne dokumentacije (Arora & Arora, 2016).

2.2.2 Vitka metodologija

Vitka metodologija je nastala na podlagi Toyotinih načel proizvodnje avtomobilov, ki se osredotočajo na zmanjševanje zapravljenih količin in povečevanje produktivnosti. Metodologija, prenesena iz avtomobilske v računalniško industrijo, narekuje razvijalcem izogibanje neproduktivnim aktivnostim, a redno zagotavljanje kakovosti opravljenih nalog (Poppendieck & Cusumano, 2012).

Metodologija se po Toyoti zgleduje tudi v nenehnem učenju in odlaganju prenapetih odločitev, saj razvijalcem omogoča, da je razvoj odprtega tipa, zato praviloma dokončno odločitev sprejmejo proti koncu ob upoštevanju vseh dejavnikov. Pri tej metodologiji je za razvijalce ključno, da prepoznajo ozka grla, ki bi lahko ovirala proces razvoja (Cockburn, 2002).

Vitka metodologija je osnovana na petih načelih: optimiziranje procesov, odpravljanje nepotrebne, osredotočenost na stranke, vsesplošno vključevanje in sodelovanje ter nenehno izboljševanje (Ebert, Abrahamsson & Oza, 2012).

Prednosti vitke metodologije so zmanjšanje vsega nepotrebne pri projektu (odvečna koda, nepotrebna dokumentacija, ponavljajoči deli), posledično pride tudi do zmanjšanja skupnih stroškov razvoja in zmanjšanja časa, potrebnega za razvoj celotne programske opreme, poveča pa se motivacija in vključenost razvijalcev v delo, saj imajo več pooblastil pri sprejemanju odločitev.

Prav zaradi večje odgovornosti razvijalcev je glavna slabost vitke metodologije potreba po ekipi visoko usposobljenih razvijalcev, saj so lahko manj usposobljeni razvijalci preobremenjeni z odgovornostmi in izgubijo usmerjenost v projektu (Mirza & Datta, 2019).

2.2.3 Metodologija hitrega razvoja aplikacij

Metodologija hitrega razvoja aplikacij (angl. Rapid Application Development, v nadaljevanju RAD) nima splošne definicije, ki bi natančno opredeljevala metodologijo, a jo vseeno lahko opredelimo na dva načina: kot metodologijo, ki predpisuje določene faze razvoja programske opreme, in kot razred orodij, ki omogočajo hiter razvoj objektov, grafičnih uporabniških vmesnikov in ponovno uporabo kode za aplikacije (Coleman & Verbruggen, 1998).

RAD se je pojavil v zgodnjih devetdesetih letih prejšnjega stoletja, saj so si programerska podjetja želela metodologijo, ki bi se razlikovala od tradicionalnih modelov razvoja programske opreme (npr. metodologija vodnega slapa). Obstajajo štirje temeljni vidiki hitrega razvoja: orodja, metodologija, ljudje in upravljanje. Med temeljne značilnosti metodologije RAD spadajo zmožnost načrtovanja, modeliranje podatkov in procesov, pisanje kode ter testiranje in odpravljanje napak. Štiristopenjski cikel, na katerem je metodologija RAD tudi zasnovana, je tako sestavljen iz načrtovanja zahtev, oblikovanja uporabniškega vmesnika, izdelave aplikacije in prevzema s strani stranke (Beynon-Davies, Carne, Mackay & Tudhope, 1999).

Za razliko od linearnih modelov RAD sloni na izdelavi prototipov z danimi zahtevami in njihovem preizkušanju pri stranki. To se izvaja v več iteracijah, vse dokler stranka ni

zadovoljna s končnim rezultatom. Strogo testiranje prototipa daje zato dragocene povratne informacije, ki pomagajo odpraviti velik del tveganja pri razvoju programske opreme (Agarwal, Prasad, Tanniru & Lynch, 2000).

Prednosti RAD metodologije so zmanjšanje tveganj zaradi pogostih povratnih informacij s strani stranke, posledično tudi večje zadovoljstvo strank, primernost za majhne in srednje velike aplikacije zaradi velikega dela ponovno uporabljene kode ter razmeroma kratek čas za lansiranje programske opreme na trg zaradi hitrega razvoja.

Slabosti RAD metodologije so velika odvisnost od odzivnosti stranke, potreba po ekipi zelo usposobljenih in izkušenih razvijalcev, neprimernost za razvoj programske opreme z zelo omejenimi financami ter pomanjkanje dokumentacije napredka razvoja zaradi morebitnih sprememb zahtev s strani kupca (Iqbal & Idrees, 2017).

2.2.4 Metodologija funkcionalno vodenega razvoja

Metodologija funkcionalno vodenega razvoja (angl. Feature Driven Development, v nadaljevanju FDD) je metodologija razvoja programske opreme, ki sodi v družino agilnih metodologij. Cilj metodologije je dokaj preprost – preprečiti zmedo, ki bi lahko vodila v časovno in stroškovno drago popravljanje razvite programske opreme.

FDD predpisuje razčlenitev razvojnih dejavnosti na podlagi seznama funkcij, ki jih mora programska oprema na koncu vsebovati. Za vsako od funkcij gredo razvijalci skozi iteracijo načrtovanja, oblikovanja in gradnje. Časovni okvir za izvedbo določene funkcije naj ne bi presegal dveh tednov, ravno zaradi tega pa so rezultati metodologije hitro vidni in razmeroma učinkovitejši od preostalih metodologij. Prav zaradi hitrega poteka razvoja pa je nujno potrebna podrobna dokumentacija, še posebej zato, ker metodologijo uporabljajo večinoma večje razvijalske ekipe (Goyal, 2008).

Načrtovanje metodologije FDD temelji na diagramu razredov, ki prikazujejo glavne objekte v problemski domeni in odnose med njimi. Ko so dokončani objekti združeni, tvorijo rešitev večjega problema. Ob tem se definirajo tudi potrebne funkcije, vsaka funkcija, ki je preveč zapletena, da bi jo bilo mogoče izvesti v dveh tednih, pa se nadalje razgradi na manjše funkcije, dokler ni vsak podproblem dovolj majhen, da je rešljiv v dveh tednih (Apel & Kästner, 2009).

Prednosti metodologije FDD so razčlenitev zapletenih nalog na manjše aktivnosti, kar povečuje učinkovitost, neodvisno delo na več nalogah hkrati in uporaba vnaprej določenih standardov in dobrih praks, kar vodi do predvidljivega uspeha.

Slabosti metodologije FDD so neprimernost za manjše projekte, saj je časovno in stroškovno potratna, velika odvisnost od vodje projekta, saj je ta zadolžen za usklajevanje

nalog med razvijalci, ter morebitno odstopanje od želja strank, saj metodologijo vodijo dejavnosti ter potreba po zelo natančni in sprotni dokumentaciji (Mirza & Datta, 2019).

3 PREDSTAVITEV UPORABLJENIH TEHNOLOGIJ IN METOD

3.1 Programski jezik in moduli

V tem podpoglavju bom predstavil programski jezik Python in vse module, ki sem jih uporabil pri razvoju programske opreme za analizo javnega mnenja na podlagi objav uporabnikov platforme Twitter na primeru vojne v Ukrajini. Pri predstavitvi modulov se bom posebej osredotočil na funkcije in funkcionalnosti, ki so mi koristile pri razvoju skript za pridobivanje, analiziranje in prikaz podatkov.

3.1.1 Programski jezik Python

Python je interaktiven in objektno usmerjen interpretativni programski jezik, kar pomeni, da za zagon skript ne potrebuje programskega prevajalnika in je torej prenosljiv na vse sisteme, ki imajo nameščen Python paket. Podpira module, dinamične podatkovne tipe in razrede. Poleg objektno usmerjenega programiranja podpira več programskih paradigem, kot sta proceduralno in funkcionalno programiranje. Programski jezik Python zaznamuje predvsem zelo jasna in kratka sintaksa, zato je po navadi prvi programski jezik, ki se ga začetniki naučijo. Nenazadnje pa je Python tudi prenosljiv: deluje na številnih različicah Unixa, vključno z Linuxom in MacOS, ter v operacijskem sistemu Windows (Python Software Foundation, 2023c).

Programerji pogosto vzljubijo programski jezik Python zaradi večje produktivnosti, ki jo zagotavlja z zelo enostavno in kratko sintakso. Ker je Python interpretativni programski jezik, nima vmesne stopnje prevajanja programske kode v strojno kodo, prav zato pa je cikel pisanja kode, testiranja in odpravljanja napak hitrejši kot pri drugih programskih jezikih (Python Software Foundation, 2023e).

Python je nastal konec osemdesetih let prejšnjega stoletja, zasnoval ga je Guido van Rossum, nizozemski programer zaposlen v podjetju Centrum Wiskunde & Informatica (CWI). Osebnostno je vodil projekt razvoja naslednika programskega jezika ABC od decembra 1989 do julija 2018, ko je prenehal opravljati funkcijo vodenja projekta (Venness, 2003). Od leta 2019 projekt vodi petčlanski nadzorni svet, ki ga sestavljajo razvijalci jedra programskega jezika Python (The Python core team and community, 2018).

Zadnja verzija programskega jezika Python je verzija 3.11, izdana oktobra 2022 in razvita na podlagi predhodne verzije 3.10. Verzija 3.11 je za 10 – 60 % hitrejša od predhodne

verzije, vključuje pa tudi poenostavljeno asinhrono izvajanje skript, izboljšanje statičnega tipiziranja, podporo konfiguracijskim datotekam in jasnejša sporočila o napakah. Verzija 3.11 naj bi bila podprta do oktobra 2027, torej za obdobje petih let (Galindo Salgado, 2021).

3.1.2 Modul Pandas

Pandas je Pythonov pomožen modul, ki se najpogosteje uporablja za naloge podatkovne znanosti/analize podatkov in strojnega učenja. Pandas je eden izmed najbolj priljubljenih paketov za obdelavo podatkov, zato je v njem zagotovljena interoperabilnost s številnimi drugimi moduli za podatkovno analitiko znotraj ekosistema Python in je običajno vključen v vse distribucije programskega jezika Python (NumFOCUS Inc., 2023b).

Pandas omogoča preprosto opravljanje številnih zamudnih in ponavljajočih se nalog s podatki, kot so npr. čiščenje podatkov, vnos podatkov, normalizacija podatkov, združevanje in povezovanje podatkov, vizualizacije podatkov, statistične analize, preverjanje podatkov ter nalaganje in shranjevanje podatkov (Pandas, 2023).

Podatki se dandanes večinoma nahajajo v tabelarni obliki, kjer ena vrstica predstavlja en vnos oz. en podatek. Tak primer so denimo Excel preglednice ali relacijske tabele podatkovnih baz. Tudi v primeru tega dela so bili podatki strukturirani na ta način, zato sem potreboval podatkovno strukturo, ki bi znala brati, spreminjati, analizirati in izvažati podatke iz teh struktur. Taka podatkovna struktura obstaja in je del Pandas paketa.

Posebnost Pandas paketa je t.i. podatkovna struktura `DataFrame`. `DataFrame` je dvodimenzionalna tabelarna podatkovna struktura z možnostjo spreminjanja velikosti, podatki pa so v njej tabelarično razvrščeni v vrstice in stolpce. `DataFrame` je torej sestavljen iz treh glavnih komponent, in sicer podatkov, vrstic in stolpcev. Pandas omogoča hiter uvoz in izvoz podatkov iz Excel, CSV, SQL, tekstovnih in JSON datotek, hrani pa jih v strukturi `DataFrame`, na kateri lahko programer izvaja vrsto operacij: indeksiranje, filtriranje, spreminjanje, računanje, analiziranje in brisanje podatkov (NumFOCUS Inc., 2023a).

3.1.3 Modul Snsrape

Modul Snsrape omogoča luščenje (angl. scraping) podatkov iz družabnih omrežij. Podpira luščenje objav, komentarjev, všečkov, ogledov, delitev, foto in video vsebin na družbenih omrežjih, kot so Facebook, Instagram, Reddit, Twitter, VKontakte, Mastodon, Telegram in Weibo. Omogoča tudi luščenje podatkov o uporabnikih, ki so določeno vsebino objavili, kot so npr. ime in priimek, uporabniško ime, profilna slika, datum ustvarjanja uporabniškega profila itd.

Posebnost modula Snsrape je enostavnost uporabe, hitrost luščenja podatkov ter odprtost uporabe, saj od uporabnika ne zahteva nobenega API ključa. To pomeni, da lahko programer, ki uporablja ta modul, dostopa do vseh zgoraj navedenih podatkov na vseh podprtih družbenih omrežjih, ne da bi za vsako družbeno omrežje posebej zaprosil za API dostop (JustAnotherArchivist, 2023).

3.1.4 Modul Transformers

Modul Transformers zagotavlja orodja za enostaven prenos in uporabo najsodobnejših predtreniranih modelov za obdelavo naravnega jezika, računalniški vid ter zvok. Z uporabo predtreniranih modelov se proces analize podatkov precej poenostavi, predvsem energetske in časovne. Če bi hotel uporabnik ustvariti in testirati svoj model za obdelavo naravnega jezika na namišljeni bazi 100.000 zapisov, bi moral najprej ročno pregledati in etiketirati 70 – 80 % vseh zapisov, nato pa bi na preostalih 20 – 30 % preizkusil pravkar razviti model. Razvoj modela pa je časovno in energetske zelo potratna naloga, zato lahko uporabnik uporabi predtrenirane modele in tako zmanjša količino časa in električne energije, ki bi jo moral vnesti v ta proces.

Modul Transformers trenutno podpira 190 predtreniranih modelov. Veliko teh so jih razvila tehnološka podjetja, kot so Google, Facebook, Microsoft, Salesforce, nekaj pa je tudi takih, ki so jih razvili znanstveniki in raziskovalci iz ameriških, evropskih in kitajskih univerz (Hugging Face, 2023b).

3.1.5 Modul Scipy

Modul Scipy (kratica za Scientific Python) je namenjen predvsem matematičnim, znanstvenim in inženirskim nalogam. Vključuje funkcije za statistiko, optimizacijo, integracijo, linearno algebro, Fourierjeve transformacije, obdelavo signalov in slik ter še veliko drugih (SciPy, 2023).

Modul Scipy je razširitev modula NumPy, saj vključuje še dodatna orodja za računanje z matrikami in specializiranimi podatkovnimi strukturami, kot so k-dimenzionalna drevesa. Modul Scipy je zasnovan za delo z Numpy tabelami, ki so zasnovane in implementirane v programskem jeziku C, prav zaradi tega pa je hitrejši od standardnih Pythonovih tabel. To je priljubljena lastnost, še posebej za velike podatkovne zbirke (Python Software Foundation, 2023d).

3.1.6 Modul Numpy

Modul Numpy ponuja obsežne matematične funkcije, generatorje naključnih števil, Fourierjevo transformacijo in še mnogo drugih, posebnost Numpy modula pa so n-dimenzionalne Numpy tabele (Python Software Foundation, 2023a).

Numpy tabele so implementirane v programskem jeziku C, kar jih naredi do 50-krat hitrejše od standardnih Pythonovih tabel. Ker so Numpy tabele hranjene na enem samem celovitem mestu v računalniškem glavnem pomnilniku, je procesiranje njihovih podatkov in upravljanje z njimi veliko hitrejša (NumPy, 2023).

3.1.7 Modul Matplotlib

Modul Matplotlib se uporablja za vizualizacijo podatkov v Pythonu. Temelji na Numpy tabelah, zasnovan je za delo s širšim sklopom funkcij modula Scipy, omogoča pa risanje in izvažanje vizualizacij podatkov na več platformah. Podatke, ki jih želi uporabnik prikazati, bi lahko predstavil v tabeli, a je ena največjih prednosti vizualizacije prav ta, da omogoča vizualni dostop do velikih količin podatkov v enostavno berljivih vizualnih prikazih.

Matplotlib podpira linijske, stolpčne, histogramске, tortne, signalne in razpršitvene grafikone. Vsakemu grafikonu lahko doda naslov, podnaslov, osi, skalo, oznake, legendo, omogoča spreminjanje barv, širine in oznake točk in črt ter spreminjanje samega grafikona, njegove velikosti, strukture, formata izvoza ter možnost dodajanja podgrafikonov in slik (The Matplotlib development team, 2023). Zaradi velike fleksibilnosti je modul Matplotlib zelo priljubljen med raziskovalci in podatkovnimi analitiki.

3.1.8 Modul NLTK

Modul NLTK (angl. natural language toolkit, v nadaljevanju NLTK) je glavno orodje v programskem jeziku Python za delo z analizami naravnega jezika. Zagotavlja vire za obdelavo besedil za klasifikacijo, tokenizacijo, stemming, označevanje, razčlenjevanje in semantično sklepanje.

Zaradi obsežne dokumentacije, razširjenosti uporabe in prenosljivosti med sistemi je NLTK modul primeren predvsem za jezikoslovce, inženirje, študente in raziskovalce. Poleg tega pa je NLTK brezplačen, saj se gre za odprtokoden projekt, ki ga vodi skupina razvijalcev (NLTK Project, 2023).

Modul NLTK omogoča analizo naravnega jezika več jezikov: angleščine, nemščine, francoščine, španščine, portugalsščine, italijanščine, nizozemščine ter grščine, navsezadnje pa tudi slovenščine. Za vsakega izmed teh jezikov nudi tudi nabor praznih besed (angl. stop words), ki jih je potrebno pred analizo besedila odstraniti, saj besedilu ne prinašajo dodane vrednosti. Primeri angleških praznih besed so npr. *myself*, *what*, *who*, *a*, *the*, *did*, primeri slovenskih praznih besed pa so *ali*, *ampak*, *in*, *marveč*, *oziroma*, *pa*, *temveč*, *toda* itd.

3.1.9 Modul Wordcloud

Besedni oblak (angl. word cloud) je način prikaza podatkov, ki se uporablja za predstavitev besedilnih podatkov, pri čemer velikost vsake besede označuje njeno pogostost oz. pomembnost, pogosto pa se uporabljajo za analizo podatkov iz spletnih mest družabnih omrežij. Podobni so stolpičnim grafikonom, vendar so vizualno privlačnejši, čeprav si jih je včasih težje razlagati zaradi težjega primerjanja pogostosti besed med sabo.

Prav zato so najbolj uporabni, ko je potrebno hitro prepoznati najpomembnejše teme v obsežnih besedilih, razumeti splošno razpoloženje ali ton besedila, raziskati vzorce ali trende v besedilu ter sporočiti ključne ideje ali koncepte na vizualno privlačen način. Vse to pa omogoča ravno modul Wordcloud, ki poleg teh funkcionalnosti ponuja še možnost izrisa besednega oblaka na poljuben vzorec s poljubno izbranimi barvami (Mueller, 2023).

3.1.10 Modul Re

Modul Re omogoča operacije ujemanja regularnih izrazov (angl. regular expression). Regularni izrazi so majhen, a visoko specializiran programski jezik, ki je vgrajen v Python in je na voljo preko tega modula. S tem jezikom se lahko določajo pravila za iskanje niza ali možnih nizov znotraj besedil, ki jih želi uporabnik najti, prešteti, primerjati, spremeniti ali izbrisati. S tem modulom se lahko znotraj besedila najdejo npr. vse besede, ki se začnejo s črko *a*, vsi elektronski naslovi, URL povezave ali kateri koli drugi nizi, ki pripadajo določenemu vzorcu.

Jezik regularnih izrazov je razmeroma majhen in omejen, zato vseh možnih opravil obdelave nizov ni mogoče opraviti z uporabo regularnih izrazov. Obstajajo tudi naloge, ki jih je mogoče opraviti z regularnimi izrazi, vendar se izkaže, da so lahko v tem primeru nekateri izrazi zelo zapleteni, še posebej, če se uporabnik prvič srečuje z regularnimi izrazi. V teh primerih je bolje, da uporabnik za obdelavo niza napiše Python skripto. Ta bo zagotovo počasnejša od regularnih izrazov, saj so ti implementirani v programerskem jeziku C, Pythonova koda pa je vedno počasnejša od zapletenega regularnega izraza, a je kljub temu bolj razumljiva (Kuchling, 2023).

3.1.11 Modul Geopy

Modul Geopy služi kot odjemalec za storitve geokodiranja. Razvijalcem omogoča, da z uporabo geokodirnikov spletnih mest in drugih podatkovnih virov enostavno najdejo koordinate naslovov, mest, držav in znamenitosti po vsem svetu.

Geopy vključuje geokodirne razrede za OpenStreetMap, Google Geocoding, Yandex, MapTiler, AzureMaps, Bing itd. Ponuja tudi vrsto drugih funkcionalnosti, saj poleg pridobivanja koordinat naslovov omogoča tudi računanje razdalje med dvema točkama, pridobivanje naslova na podlagi koordinat ter prikaz naslova v originalnem jeziku (Tokyo bo torej zapisan v japonsščini) (GeoPy Contributors, 2023).

3.1.12 Modul Plotly Express

Modul Plotly Express vsebuje funkcije, s katerimi lahko uporabnik zelo enostavno ustvari celotne grafikone in slike naenkrat, njihova posebnost pa je, da so interaktivne. Plotly Express je vgrajen del knjižnice plotly in je priporočljiv predvsem za ustvarjanje najbolj običajnih in pogostih slik.

Plotly Express ponuja več kot 35 funkcij za ustvarjanje različnih vrst grafikonov in slik. Vmesnik za te funkcije je bil skrbno zasnovan tako, da je čim bolj dosleden in enostaven za učenje, tako da je med programiranjem in obdelavo podatkov zelo enostavno preklopiti iz razpršenega diagrama na stolpčni diagram ali pa iz histograma na sončni grafikon. Zanimive pa so tudi interaktivne mape, na katerih lahko uporabnik prikaže določeno mesto na podlagi koordinat, oblikuje ga lahko z barvami in raznimi oblikami, da bo čim bolj vidno in razumljivo, velikost njegove oznake pa lahko prilagodi glede na razpoložljive podatke (Plotly, 2023).

3.1.13 Modul Datetime

Datum in čas nista samostojna podatkovna tipa v Pythonu, a ju je mogoče izraziti s pomočjo razredov iz modula Datetime. Modul Datetime je že privzeto vgrajen v programski jezik Python, zato ga ni potrebno nameščati iz zunanjih virov.

Modul Datetime ponuja razrede za delo z datumom in časom. Ti razredi zagotavljajo številne funkcije za delo z datumi, časi in časovnimi intervali. Glavni razredi modula Datetime so: `date` – datum z atributi leto, mesec in dan, osnovan na gregorijanskem koledarju, `time` – čas z atributi ura, minuta, sekunda, mikrosekunda in časovni pas, ki je neodvisen od posameznega dneva, `datetime` – kombinacija datuma in časa z atributi leto, mesec, dan, ura, minuta, sekunda, mikrosekunda in časovni pas ter `timedelta` – trajanje med dvema datumoma in časoma z mikrosekundno natančnostjo (Python Software Foundation, 2023b).

Modul Datetime zelo poenostavi obdelavo in računanje z datumi in časi, saj omogoča enostavno pretvarjanje različnih tipov zapisov datuma in časa, poleg tega pa podpira vključevanje Datetime spremenljivk v zanke. Razvijalcu je torej potrebno le definirati začetni in končni datum in/ali čas ter razmik med eno iteracijo in drugo, program pa bo

samodejno prišteval in odšteval datume in/ali čase (npr. izpis vseh datumov in ur v mesecu februarju s časovnim intervalom 10 min).

3.2 Razvojno okolje

Za razvoj opisane programske opreme sem uporabil razvojno okolje Visual Studio Code, ki je enostavno za uporabo in omogoča vse, kar sem potreboval za izdelavo za analizo javnega mnenja na podlagi objav uporabnikov platforme Twitter. Visual Studio Code je predstavljen v nadaljevanju.

Visual Studio Code je brezplačen urejevalnik programske kode (angl. code editor), ki ga je Microsoft predstavil javnosti aprila 2015 (McBreen, 2015). Skozi leta je postalo razvojno okolje Visual Studio Code zelo priljubljeno med razvijalci, raziskava, ki jo je leta 2022 izvedel StackOverflow, pa je pokazala, da tri četrtine razvijalcev uporablja Visual Studio Code za razvoj določene programske opreme (Stack Exchange Inc., 2022). Razlogov za priljubljenost tega razvojnega okolja je več.

Visual Studio Code namreč podpira operacijske sisteme MacOS, Linux in Windows, ti trije pa predstavljajo skupno več kot 92 % vseh operacijskih sistemov (StatCounter, 2023). Ima tudi razmeroma hiter urejevalnik izvorne kode, predvsem zaradi tega, ker gre za poenostavljen urejevalnik kode s podporo razhroščevanja, izvajanje opravil in nadzora različic in ne za integrirano razvojno okolje (angl. integrated development environment) (Microsoft, 2023b). Podpira več sto programskih jezikov in tudi označevanje sintakse, ujemanje oklepajev, samodejno indentiranje, bližnjice in še mnogo drugih funkcionalnosti.

Visual Studio Code vključuje tudi podporo za dokončanje kode IntelliSense, ki razvijalcu pomaga semantično razumeti kodo in kontekst, v katerem se uporablja, ter na podlagi tega predlagati primerne ukaze.

Visual Studio Code vključuje tudi interaktivni razhroščevalnik, ki razvijalcu omogoča pregled izvorne kode po korakih, obenem pa spremljanje spremenljivk in njihovih vrednosti, pregled sklada ter izvajanje ukazov v konzoli. Razhroščevalnik je ključen pri iskanju in odpravljanju napak, še posebej pri zelo obsežni programski opremi.

Visual Studio Code ima tudi podporo za Git – orodja za upravljanje izvorne kode. Git se uporablja za sledenje spremembam v izvorni kodi, kar omogoča sodelovanje več razvijalcev pri nelinearnem razvoju programske opreme, vse to znotraj razvojnega okolja. Git znotraj orodja Visual Studio Code deluje, ne da bi razvijalec zapustil urejevalnik, vključno z ogledom razlik v čakajočih spremembah (Microsoft, 2023c).

3.3 Analiza razpoloženja

Raziskovalno področje analize razpoloženja je tesno povezano z računalniškim jezikoslovjem, obdelavo naravnega jezika (angl. natural language processing) in rudarjenjem besedil (angl. text mining). To področje izhaja iz preučevanja psihologije in presoje ter skuša odgovoriti na vprašanja s katerimi se ukvarjata tudi podatkovno rudarjenje in računalniško jezikoslovje. Analizi razpoloženja nekateri pravijo tudi analiza subjektivnosti, rudarjenje mnenj in pridobivanje ocen.

Analiza razpoloženja običajno preučuje subjektivnost besedil na nivoju dokumentov, stavkov ali celo posameznih besed. Težava uporabe analize razpoloženja se ne pojavi v besedilih z eksplicitnim razpoloženjem (npr. "Lepo je živeti"), ampak pri teh, kjer je mnenje izraženo implicitno (npr. "Natakar je polil kozarec vode"). Logično je, da je eksplicitna besedila, v katerih nastopa veliko eksplicitnih pridevnikov (npr. okusno, čudovito, imenitno, grozno, strašno), veliko lažje analizirati, pri implicitnih pa je potrebno avtorjevo razpoloženje razumeti iz uporabljenih besed in konteksta (Mejova, 2009).

Za analizo avtorjevega razpoloženja iz besedila se vsaki besedi pripiše ti. polarnost (angl. polarity). Polarnost je lahko pozitivna ali negativna, najlažje pa si jo je predstavljati kot nek razpon med najbolj pozitivnimi in najbolj negativnimi čustvi v obravnavanem besedilu. Izraža se kot številčna ocena, ki zavzema vrednosti med -100 % in 100 % (oz. -1 in 1), pri čemer 0 pomeni nevtrarno razpoloženje. Tako imajo recimo promocijski letaki polarnost zelo blizu oceni 1, objektivni časopisni članki oceno čim bližje 0, negativni komentarji gostov propadajočega hotela ali slabe restavracije pa oceno blizu -1 (Bhadane, Dalal & Doshi, 2015).

Zaradi izjemnega uspeha platforme Twitter je postala ta platforma dragocen nabor podatkov, ki nenehno raste. Ta stalni tok informacij je pritegnil zanimanje raziskovalne skupnosti zlasti zaradi vseh možnih aplikacij analize razpoloženja na besedilih, ki jih objavljajo uporabniki, saj so na tej platformi objavljena mnenja o športu, religiji, politiki, ekonomiji itd. Niso pa zgolj raziskovalci tisti, ki se zanimajo za tvite, temveč se zanimajo tudi politiki in podjetja, ki želijo vedeti, kaj si ljudje mislijo o njih v praktično realnem času (Sarlan, Nadam & Basri, 2014).

Zaradi posebnosti platforme Twitter in značilnosti njenih besedil, je potrebno pri analizi razpoloženja tvitov upoštevati, da so v večini primerov besedila kratka (par stavkov) in da ne presegajo 280 znakov. Jezikovni slog tvitov je običajno neformalen, besedila vsebujejo veliko kratic in idiomov, zelo pogosta pa je uporaba žargona. Uporabniki večinoma ne skrbijo za pravilno rabo slovnice, kar pa povečuje težave pri izvajanju jezikovne analize. Velikokrat pa veliko težavo predstavlja tudi pomanjkanje konteksta, saj uporabniki platforme pogosto objavijo svoje mnenje, ne da bi ga bolj podrobno

razložili ali mu dodali kontekst, v katerega spada (Martínez-Cámara, Martín-Valdivia, Urena-López & Montejo-Ráez, 2014).

Zaradi zgoraj opisanih težav je pred izvedbo analize razpoloženja objavam, ki so bile pridobljene na družbenih omrežjih, potrebno nameniti dodatno pozornost. Ugotavljanje razpoloženja na platformi Twitter ni trivialna naloga in se precej razlikuje od ugotavljanja razpoloženja v običajnih besedilih, kot so blogi in forumi, zato je potrebno upoštevati posebne značilnosti te platforme.

Ena od edinstvenih značilnosti tvitov je njihova relativno kratka dolžina, ki je nekoč lahko znašala do 140 znakov, od leta 2017 pa do največ 280 znakov. Prav zaradi dolžine se analiza razpoloženja tvitov razlikuje od analize razpoloženja daljših besedil, kot so blogi ali filmske recenzije. Čeprav je to lahko tudi prednost, saj je razvrščanje krajših besedil precej enostavnejše od daljših, pa je slabost v tem, da različni klasifikatorji različno upoštevajo dolžino obravnavanega besedila, zato je potrebno preizkusiti več algoritmov. Objave na platformi Twitter so napisane v različnih jezikih, ki so lahko uporabljeni tudi istočasno, včasih celo v isti objavi. Težave pri odkrivanju jezikov, še posebej pri razlikovanju tistih iz skupnih ali sorodnih jezikovnih družin, se povečajo tudi zaradi kratke dolžine tvitov (Giachanou & Crestani, 2016).

Analiza razpoloženja se običajno opravi na predtem definirani in določeni temi in ne na besedilu, za katerega ne poznamo konteksta. Zato je veliko dela usmerjenega v preverjanje ustreznosti objav. Da bi zajeli tematsko relevantnost tvita, številni raziskovalci preprosto upoštevajo prisotnost določene besede, največkrat so to ključniki (angl. hashtag) (Giachanou & Crestani, 2016).

Zelo pogosta težava pa je tudi slovnica. Jezik, ki se uporablja na platformi Twitter, se zaradi neformalnega načina komuniciranja in omejene dolžine zelo razlikuje od jezika, ki se uporablja v drugih besedilnih zvrsteh (splet, blogi, novice itd.). Tviti vsebujejo besedilne posebnosti, vključno s poudarjenim pisanjem velikih začetnic, poudarjenim podaljševanjem, okrajšavami, emojiji ter uporabo slengov in neologizmov (Giachanou & Crestani, 2016).

Tviti vsebujejo veliko šuma zaradi obsežne uporabe nepravilne angleščine in pravopisnih napak. Ta pojav, ki je znan kot redkost podatkov (angl. data sparsity), vpliva na splošno uspešnost analize razpoloženja (Giachanou & Crestani, 2016).

Pri zaznavanju polarnosti čustev v sporočilu ima pomembno vlogo prisotnost zanikovalnih besed. Odkrivanje in ustrezno ravnanje z zanikanjem ni trivialno in predstavlja velik izziv. Odkrivanje negacij je pomembno, ker lahko povzročijo preobrat polarnosti sporočila (pozitivno postane negativno ali obratno), zato številni algoritmi preprosto obrnejo polariteto besede ob zaznavi zanikanja (Giachanou & Crestani, 2016).

Prazne besede so pogoste besede, ki pa nimajo vpliva na razpoloženje iz besedila (npr. the, is, a), zato jih je pred obdelavo besedila potrebno izločiti. Običajno se te besede izloči na podlagi vnaprej pripravljenega seznama besed, ki niso primerne za obravnavano platformo (Giachanou & Crestani, 2016).

Za določitev polarizacije posameznih besed pa je potrebno celotno besedilo najprej tokenizirati (angl. tokenization). Tokenizacija je postopek razdelitve besedne zveze, stavka, odstavka, enega ali več besedilnih dokumentov na manjše enote, običajno posamezne besede. Vsaki od teh manjših enot pravimo žeton (angl. token), lahko pa so poleg navadnih besed tudi znaki ali emoji (Giachanou & Crestani, 2016).

V nekaterih primerih pa vsebujejo tviti tudi multimedialno vsebino, kot je npr. slika ali videoposnetek. Slike in videoposnetki so lahko dragocen vir podatkov za razumevanje razpoloženja objave, saj lahko nudijo koristne informacije o avtorju, ideji, razpoloženju, viru podatkov itd., a je postopek pridobivanja teh informacij še vedno premalo raziskano področje, hkrati pa precej zahtevnejše od analize tekstovnega besedila (Giachanou & Crestani, 2016).

Zaradi vseh zgoraj naštetih posebnosti platforme Twitter, je potrebno za analizo razpoloženja tvitov izbrati tak model, ki se zna spopadati s težavami nepravilne slovnice, kratkih besedil, ključnikov, emojijev itd. Model, ki te posebnosti upošteva, je model roBERTa. roBERTa je model za analizo angleških besedil, ki je bil razvit na podlagi objav na platformi Twitter in ostalih družbenih spletnih omrežij ter predtreniran na skoraj 60 milijonih tvitov in se je izkazal za izjemno zanesljivega pri analizi razpoloženja tvitov. V raziskavi iz leta 2020 je premagal model, ki je bil predtreniran izključno na tvitih (Barbieri, Camacho-Collados, Neves & Espinosa-Anke, 2020), zelo podobne rezultate pa je požel že v študiji iz leta 2019 (Liu in drugi, 2019).

4 PREDSTAVITEV PODATKOV

4.1 Predstavitev vira in format podatkov

V tem poglavju bom podrobneje predstavil vir in format podatkov, na katerih sem izvedel analizo razpoloženja.

Preučeval sem objave na platformi Twitter, torej so bili predmet preiskave tviti. Tvite lahko objavljajo le registrirani in prijavljeni uporabniki platforme Twitter, anonimno objavljane ni možno. Tvit lahko sestavlja besedilo, ki ne sme presegati 280 znakov, uporabnik pa lahko svojemu tvitu doda en videoposnetek ali animirano sliko (.gif) ali do največ 4 slike. Uporabnik se lahko tudi odloči, da v svoj tvit vključi anketo z največ štirimi možnimi odgovori in omejenim časovnim trajanjem, bodisi za anketo bodisi za tvit pa lahko določi, kdo bo lahko nanjo odgovarjal oz. komu bo viden.

Ko uporabnik objavi tvit, ostali uporabniki vidijo besedilo in/ali naložene multimedialne vsebine, število komentarjev tvita in z dodatnim klikom tudi njihovo vsebino, število delitev tvita in z dodatnim klikom tudi profile oseb, ki so originalen tvit delile, število všečkov tvita in z dodatnim klikom tudi profile oseb, ki so tvit všečkale, število citiranj tvita in z dodatnim klikom tudi profile oseb, ki so tvit citirale, število dodanih zaznamkov ter število ogledov tvita. Uporabnik lahko tvit všečka, deli, citira, shrani v svoje zaznamke in nanj odgovori.

Poleg same vsebine tvita pa je vidna tudi ura objave ter avtor tvita oz. njegov profil. Vidno je njegovo ime, ki si ga je sam poljubno izbral, vidno je uporabniško ime, ki si ga je izbral ob registraciji v platformo, in vidna je tudi profilna slika. Kateri koli uporabnik platforme Twitter lahko spreminja svoje ime, profilno sliko in uporabniško ime, a mora biti to unikatno znotraj platforme, saj označuje posameznega uporabnika.

Ob dodatnem kliku na ime, uporabniško ime ali profilno sliko določenega uporabnika, lahko uporabnik vidi še sliko ozadja profila, mesec in leto registracije v platformo, opis in lokacijo, ki si ju uporabnik lahko po želji nastavi, število sledilcev in sledenj ter vse objave, všečke, komentarje in multimedialne vsebine uporabnika. Poleg tega ima uporabnik na voljo opcijo, da določenemu uporabniku sledi, pošlje zasebno sporočilo, ga blokira, prijavi ali pa deli njegov profil.

Večino izmed zgoraj opisanih podatkov oz. tiste, ki so bili relevantni za preučevanje javnega mnenja glede vojne v Ukrajini v okviru magistrskega dela, sem pridobil s pomočjo Python modula za luščenje spletnih strani Snsrape.

Predmet raziskave je javno mnenje v prvem letu vojne v Ukrajini, zato sem pridobil vse tvite, ki vsebujejo besedi "russia" in "ukraine", ki so bili objavljeni na platformi Twitter v obdobju od 24. 2. 2022 do 24. 2. 2023. Tvite sem filtriral po jeziku – izbral sem le angleškega, predvsem zaradi modela za analizo razporeditve roBERTa, ki je specializiran za objave v angleščini, a tudi zaradi manjšega nabora podatkov in dejstva, da so objave, ki želijo doseči širšo publiko, večinoma napisane v angleščini. Tvite sem filtriral tudi po prisotnosti zunanjih povezav – te sem namreč izločil predvsem z namenom, da bodo analizirane čim bolj negativne ali pozitivne objave, saj so objave z URL povezavami običajno objave spletnih časopisov in novic, za katere lahko predpostavim, da so nevtralne. Izločil sem tudi komentarje tvitov, predvsem zaradi manjšega nabora podatkov in posledičnega hitrejšega procesiranja rezultatov analize razporeditve.

Poizvedba je v tem primeru izgledala tako: `ukraine russia lang:en until:2023-02-24 since:2022-02-24 -filter:links -filter:replies`. Modul Snsrape mi je za vse rezultate, ki odgovarjajo tej poizvedbi, omogočal pridobivanje sledečih podatkov (JustAnotherArchivist, 2023):

url: niz, URL povezava objave
date: datum in ura, datum in ura objave v UTC+0 zapisu
rawContent: niz, besedilo objave
renderedContent: niz, besedilo objave, kot ga vidijo uporabniki platforme
id: celo število, unikatno identificira objavo
user: uporabnik, podatki o uporabniku
replyCount: celo število, število komentarjev objave
retweetCount: celo število, število delitev objave
likeCount: celo število, število všečkov objave
quoteCount: celo število, število citiranj objave
conversationId: celo število, unikatno identificira objavo in komentar ali citiranje
lang: celo število, jezik objave
source: niz, operacijski sistem naprave, iz katere je bil tvit objavljen
sourceUrl: niz, URL povezava do Twitter aplikacije za operacijski sistem naprave objave
sourceLabel: niz, ime aplikacije Twitter na operacijskem sistemu objave
links: tabela povezav, seznam URL povezav v tvitu
media: tabela multimedialnih vsebin, video in slike, objavljene v tvitu
retweetedTweet: tvit, podatki o deljenem tvitu
quotedTweet: tvit, podatki o citiranem tvitu
inReplyToTweetId: celo število, unikatno identificira komentirani tvit
inReplyToUser: celo število, unikatno identificira avtorja komentarja tvita
mentionedUsers: tabela uporabnikov, seznam vseh citiranih uporabnikov
coordinates: koordinate, geografske koordinate lokacije dodane tvitu
place: kraj, geografsko ime kraja dodanega tvitu
hashtags: seznam nizov, seznam vseh uporabljenih ključnikov
cashtags: seznam nizov, seznam vseh uporabljenih delniških oznak
card: kartica, multimedialni dodatek tvita
viewCount: celo število, število ogledov objave
vibe: status, uporabnikov trenutni status

Za avtorja objave in citirane uporabnike pa so na voljo še sledeči podatki (JustAnotherArchivist, 2023):

username: niz, uporabniško ime
id: celo število, unikatni identifikator uporabnika
displayName: niz, uporabnikovo ime
rawDescription: niz, opis uporabnika
renderedDescription: niz, opis uporabnika kot ga vidijo uporabniki platforme
descriptionLinks: tabela povezav, seznam URL povezav v opisu
verified: binarna vrednost, ali ima uporabnik verificiran uporabniški račun

created: datum in ura, datum in ura registracije v platformo v UTC+0 zapisu
followersCount: celo število, število sledilcev
friendsCount: celo število, število sledenj
statusesCount: celo število, število objav
favouritesCount: celo število, število všečkanih objav
listedCount: celo število, število seznamov, v katere je uporabnik vključen
mediaCount: celo število, število objavljenih multimedialnih vsebin
location: niz, nastavljena lokacija uporabnika
protected: binarna vrednost, ali ima uporabnik zaseben uporabniški račun
link: povezava, poljubna URL povezava uporabnika
profileImageUrl: niz, naslov profilne slike uporabnika
profileBannerUrl: niz, naslov slike ozadja uporabnika
label: oznaka, opis in URL povezava do profila uporabnika

Izmed vseh zgoraj naštetih atributov so bili za namen magistrskega dela izbrani sledeči: id, rawContent, date, replyCount, retweetCount, likeCount, viewCount za objekt Tweet ter username, displayName, id, verified, created, followersCount, friendsCount, statusesCount, location za objekt User.

Primer formata zapisa enega izmed izluščenih tvitov v angleškem jeziku, ki je bil objavljen v obdobju od 24. 2. 2022 do 24. 2. 2023 in ki vsebuje tako besedo "russia" kot besedo "ukraine", je prikazan spodaj:

```
{
  "Tweet": {
    "id": 1628889949440249858,
    "User": {
      "username": "@RandomIgnorance",
      "displayName": "Jimmy",
      "id": 42792991,
      "verified": false,
      "created": datetime.datetime(2009, 5, 27, 2, 26, 11,
tzinfo=datetime.timezone.utc),
      "followersCount": 6392,
      "friendsCount": 7022,
      "statusesCount": 111480,
      "location": "Lancaster, Ohio"
    },
    "rawContent": "It's very disappointing @PMOIndia @rashtrapatibhvn are too
afraid of Russia to condemn their unprovoked invasion of Ukraine",
    "date": datetime.datetime(2023, 2, 23, 22, 50, 1,
tzinfo=datetime.timezone.utc),
    "replyCount": 0,
    "retweetCount": 0,
```

```
    "likeCount": 0,  
    "viewCount": 35  
  }  
}
```

Primer izluščenega tvita.

4.2 Analiza kakovosti podatkov

Po procesu luščenja podatkov, ki je trajal skoraj en dan, je bila podatkovna zbirka dokončana. Sestavljalo jo je 1.073.786 zapisov, za vsak zapis pa so bili na voljo naslednji atributi: `id`, `user.username`, `user.displayName`, `user.id`, `user.verified`, `user.created`, `user.followersCount`, `user.friendsCount`, `user.statusesCount`, `user.location`, `rawContent`, `date`, `replyCount`, `retweetCount`, `likeCount` ter `viewCount`.

Pri atributu `id` sem zbral 1.073.786 zapisov, nobeden izmed njih pa ni bil podvojen ali ničeln. V primeru, da bi naletel na podvojene zapise, bi te odstranil, v primeru ničelnih zapisov pa bi preučil še ostale attribute. Če bi bili tudi ti ničelni, bi zapis odstranil, sicer pa bi ga ohranil.

Pri atributu `user.username` sem zbral 1.073.786 zapisov, nobeden izmed njih ni bil ničeln, je pa bilo 621.965 (57,92 %) zapisov podvojenih. To pomeni, da je določeno število uporabnikov objavilo več tvitov in ne samo enega, zato mi teh zapisov ni bilo treba odstraniti. Ena izmed možnih interpretacij podvojenih zapisov je tudi, da več različnih uporabnikov uporablja isto uporabniško ime.

Pri atributu `user.displayName` sem zbral 1.073.786 zapisov, od tega jih je bilo 1.073.589 (99,98 %) neničelnih, 661.881 (61,64 %) pa podvojenih. Ničelnih in podvojenih imen za potrebe raziskave ni smiselno odstranjevati, saj Twitter dovoli uporabnikom uporabljati isto ime ali pa tudi objavljanje brez nastavljenega imena.

Pri atributu `user.id` sem zbral 1.073.786 zapisov, ničelnih ni bilo, 621.673 (57,89 %) pa je bilo podvojenih. Iz tega podatka lahko sklepam, da je tvite objavljalo 452.113 različnih uporabnikov. Iz podatkov o atributih `user.username`, `user.name` ter `user.id` pa lahko sklepam, da 197 (0,04 %) uporabnikov nima nastavljenega imena oz. da ima 40.208 uporabnikov (8,89 %) nastavljeno isto ime.

Pri atributu `user.verified` sem zbral 1.073.786 zapisov, nobeden izmed njih ni bil ničeln. Število tvitov, ki so jih objavili uporabniki z verificiranim uporabniškim profilom, je bilo 55.414 (5,16 %).

Pri atributu `user.created` sem zbral 1.073.786 zapisov, nobeden izmed njih ni bil ničeln, je pa bilo 622.313 (57,95 %) zapisov podvojenih. Takšnega števila podvojenih zapisov ne pripisujem le dejstvu, da je določeno število avtorjev objavilo več tvitov, ampak tudi možnosti, da sta dva uporabnika ustvarila uporabniški račun v istem trenutku.

Pri atributu `user.followersCount` sem zbral 1.073.786 zapisov, 18.998 (1,77 %) je bilo ničelnih. Teh zapisov ni bilo potrebno odstraniti, saj obstajajo tudi taki profili, ki jim nihče ne sledi, njihovi tviti pa so lahko še vedno vidni javnosti. Minimalno število sledilcev je bilo 0, maksimalno 132.772.834, povprečno 38.393, medialno pa 571.

Pri atributu `user.friendsCount` sem zbral 1.073.786 zapisov, 22.070 (2,05 %) je bilo ničelnih. Teh zapisov ni bilo potrebno odstraniti, saj obstajajo tudi taki profili, ki ne sledijo nobenemu drugemu profilu. Minimalno število sledenj je bilo 0, maksimalno 838.722, povprečno 7.984, medialno pa 523.

Pri atributu `user.statusesCount` sem zbral 1.073.786 zapisov, zgolj eden izmed teh je bil ničeln. Po pregledu tega specifičnega zapisa sem se odločil, da zapis ohranim, saj so bila vsa ostala polja zapisa pravilno izpolnjena. Sklepam, da je šlo v tem primeru za napako platforme Twitter ali modula za luščenje podatkov, saj dotični uporabnik redno objavlja tvite. Minimalno število statusov je bilo 0, maksimalno 7.223.920, povprečno 42.006, medialno pa 11.112.

Pri atributu `user.location` sem zbral 1.073.786 zapisov, 338.532 (31,53 %) je bilo ničelnih, 960.636 (89,46 %) pa podvojenih. Sklepam, da ima več uporabnikov nastavljeno isto lokacijo. Nobenega izmed teh zapisov nisem odstranjeval, saj podatek o lokaciji ni obvezen podatek, platforma Twitter ga ne zahteva, uporabniki pa ga poljubno nastavijo ali izpustijo.

Pri atributu `rawContent` sem zbral 1.073.786 zapisov, ničelnih ni bilo, je pa bilo 32.709 (3,04 %) podvojenih. Tudi teh zapisov nisem odstranil, saj se razlikujejo v ostalih atributih, v tem primeru pa gre večinoma za tvite, ki so dolgi le par besed ali pa vsebujejo le en ključnik. Ti so bili zelo pogosti ob samem začetku spopadov, ko so uporabniki z nekaj besedami sporočali, da je Rusija napadla Ukrajino (npr. "Russia started the invasion", "Russia attacked Ukraine" itd.). Nekaj je tudi takih, ki jih uporabniki znova in znova delijo, a je koristno, da tudi te obdržim, saj so vsakič znova deležni drugačne obravnave: komentarji pod njimi se razlikujejo, razlikuje se število všečkov itd.

Pri atributu `date` sem zbral 1.073.786 zapisov, nobeden izmed njih ni bil ničeln, je pa bilo 117.133 (10,91 %) zapisov podvojenih. To pripisujem dejstvu, da je bilo določeno število tvitov objavljenih na isti dan ob isti uri. Ta pojav je bil zelo pogost na prvi dan vojne, ko je bilo objavljenih več kot 110.000 tvitov, torej več kot eden na sekundo.

Pri atributu `replyCount` sem zbral 1.073.786 zapisov, od tega jih je bilo 774.656 (72,14 %) ničelnih. Iz tega podatka je razvidno, da skoraj tri četrtine objavljenih tvitov ni nihče komentiral, a sklepam, da so tudi ti tviti zanimivi za analizo javnega mnenja, čeprav se ni iz njih razvila nobena debata. Minimalno število komentarjev je bilo 0, maksimalno 114.612, povprečno 4, medialno pa 0.

Pri atributu `retweetCount` sem zbral 1.073.786 zapisov, od tega jih je bilo 782.238 (72,85 %) ničelnih. Iz tega podatka je razvidno, da so uporabniki delili zgolj vsak četrti tweet, ki je bil objavljen na platformi. Minimalno število delitev je bilo 0, maksimalno 119.887, povprečno 16, medialno pa 0.

Pri atributu `likeCount` sem zbral 1.073.786 zapisov, od tega jih je bilo 510.637 (47,55 %) ničelnih. Iz tega podatka je razvidno, da skoraj polovice tvitov ni nihče všečkal. Minimalno število všečkov je bilo 0, maksimalno 513.312, povprečno 85, medialno pa 1.

Pri atributu `viewCount` sem zbral 1.073.786 zapisov, od tega jih je bilo 981.092 (91,37 %) ničelnih. Kljub temu pa ne moremo trditi, da manj kot desetina objavljenih tvitov sploh ni bila videna iz strani drugih uporabnikov. Razlog za tako visok delež ničelnih zapisov je v tem, da je Twitter implementiral funkcionalnost za štetje ogledov tvitov komaj konec meseca decembra 2022 (Twitter Support, 2022), kar pomeni, da tega podatka ni na voljo za tvite, ki so bili objavljeni pred 22. decembrom 2022, torej za prvih deset mesecev spopadov. Minimalno število ogledov za zadnja dva meseca prvega leta vojne je bilo 1, maksimalno 12.234.535, povprečno 5.408, medialno pa 89.

5 RAZVOJ PROGRAMSKE OPREME

Razvito programsko opremo sem razdelil na tri glavne dele, saj je tako potekal tudi razvoj. Odločil sem se, da bom vse tri skripte razvil posebej, predvsem zaradi boljše preglednosti in lažjega razhroščevanja napak, kot tudi zaradi enostavnejšega "vračanja nazaj" v primeru nepravilnosti tik pred koncem razvoja. Prva skripta obsega zajem in shranjevanje podatkov, druga opravljanje analize razporeditve na zajetih podatkih, tretja pa prikaz rezultatov analize razporeditve. Sledi predstavitev vsakega dela posebej.

5.1 Pridobivanje in shranjevanje objav na platformi Twitter

Glede na to, da sem se odločil za pridobivanje podatkov uporabiti Pythonov modul `Snsrape`, sem delo začel z branjem dokumentacije in preučevanjem funkcij, ki jih ta modul ponuja. Strukturo podatkov in funkcij sem našel na uradnem git repozitoriju modula (JustAnotherArchivist, 2023).

Na podlagi dokumentacije sem se odločil, da bom za svoje namene uporabil funkcijo `get_items()` razreda `TwitterSearchScraper`, ki na podlagi podane poizvedbe vrne

uporabniku seznam izluščenih podatkov. Lahko bi pridobil celo vrsto podatkov, a se je zdelo smiselno obdržati le `id`, `user.username`, `user.displayname`, `user.id`, `user.verified`, `user.created`, `user.followersCount`, `user.friendsCount`, `user.statusesCount`, `user.location`, `rawContent`, `date`, `replyCount`, `retweetCount`, `likeCount` ter `viewCount`. Ker funkcija `get_items()` vrača seznam tvitov, sem se odločil izbrane attribute tvitov shraniti v seznamu oz. v t.i. podatkovni strukturi `DataFrame`.

Python omogoča zelo enostavno izvažanje seznamov v najrazličnejše tipe datotek: `.txt`, `.csv`, `.xlsx` itd. Zaradi preglednosti in prenosljivosti sem se odločil, da bom izluščene podatke hranil v Excel datoteki (`.xlsx`). Delo z Excel datotekami je v Pythonu zelo enostavno, predvsem zaradi modula `Pandas`, ki omogoča hitro in učinkovito izvajanje operacij po vrsticah ali stolpcih. Težje bi bilo delati v tekstovnih datotekah, saj `.txt` datoteke nimajo tabelarične strukture, `.csv` datoteke pa bi lahko povzročale težave zaradi prisotnosti vejic ali drugih posebnih znakov (npr. `|`, `\&`, `;`) znotraj tvitov, ki jih `.csv` datoteke uporabljajo kot ločilo med stolpci. V `.csv` datoteko bi bilo tudi težje izvoziti emojije, teh pa je v tvitih polno.

Da bi pridobil, shranil in izvozil vse potrebne podatke, sem na platformi Twitter poiskal še funkcijo naprednega iskanja (angl. `advanced search`, povezava <https://twitter.com/search-advanced?f=live>), da bi lahko sestavil pravilno poizvedbo, ki bi vrnila vse željene podatke. V polje “All of these words” sem vnesel “ukraine russia”, saj sem želel pridobiti tvite, ki vsebujejo obe besedi. V spustnem meniju “Language” sem izbral opcijo “English”, saj sem želel pridobiti le tvite v angleškem jeziku, ker je izbrana metoda analize razpoloženja `roBERTa` najučinkovitejša prav v angleških tvitih. Izklopil sem opcijo “Replies”, saj so me zanimali le objavljeni tviti in ne komentarji pod njimi, ter opcijo “Links”, saj sem se hotel izogniti objavam, ki vsebujejo zunanje URL povezave, ker gre v tem primeru predvsem za zunanje časopisne članke, ki jih spletni mediji objavljajo na njihovih Twitter profilih. Na koncu sem v polje “From” vnesel datum 24. februar 2022, v polje “To” pa 24. februar 2023.

S klikom na gumb “Search” se je v spletnem brskalniku stran osvežila in prešla na sledeči naslov:

https://twitter.com/search?f=live&q=ukraine%20russia%20lang%3Aen%20until%3A2023-02-24%20since%3A2022-02-24%20-filter%3Alinks%20-filter%3Areplies&src=typed_query. V tem URL naslovu je mogoče v obliki kodiranih parametrov opaziti besedi “ukraine” in “russia”, “lang en”, “filter links” in “filter replies” ter “until” in “since” skupaj z izbranimi datumoma. Ker modul `Snsrape` zahteva poizvedbo kot niz in ne kot URL povezavo, sem prekopiral niz iz iskalnega polja na vrhu strani. Ta je videti tako: `ukraine russia lang:en until:2023-02-24 since:2022-02-24 -filter:links -filter:replies`.

Preučil sem funkcijo za pridobivanje podatkov. Odločil sem se za način shranjevanja, določil sem niz poizvedbe. Napisal sem for zanko, ki se vrši čez vse izluščene podatke, izbrane shrani v seznam, na koncu pa seznam pretvori v DataFrame in ga izvozi v obliki .xlsx datoteke.

```
import snsrape.modules.twitter as sntwitter
import pandas as pd

def export_tweets(tweets, index):
    df = pd.DataFrame(
        tweets,
        columns=[
            "Tweet ID", "User username", "User name", "User ID",
            "Verified user", "User created", "Followers count", "Following
count",
            "Statuses count", "User location", "Tweet", "Tweet created",
            "Reply count", "Retweet count", "Like count", "View count"
        ],
    )
    df["User created"] = df["User created"].dt.tz_localize(None)
    df["Tweet created"] = df["Tweet created"].dt.tz_localize(None)
    df.to_excel(f".\\data-parsed-{index}.xlsx", engine="xlsxwriter")

query = "ukraine russia lang:en until:2023-02-24 since:2022-02-24 -filter:links
-filter:replies"
tweets = []
index = 1
length = 1000000

for tweet in sntwitter.TwitterSearchScraper(query).get_items():
    tweets.append(
        [
            tweet.id, tweet.user.username, tweet.user.displayName,
            tweet.user.id,
            tweet.user.verified, tweet.user.created, tweet.user.followersCount,
            tweet.user.friendsCount,
            tweet.user.statusesCount, tweet.user.location, tweet.rawContent,
            tweet.date,
            tweet.replyCount, tweet.retweetCount, tweet.likeCount,
            tweet.viewCount,
        ]
    )
    if len(tweets) >= (length + 1):
        export_tweets(tweets, index)
        tweets.clear()
        index += 1
```

```
export_tweets(tweets, index)
```

Primer kode za pridobivanje podatkov.

Excel datoteka premore največ 1.048.576 vrstic zapisov (Microsoft, 2023a), zato sem znotraj zgoraj predstavljene zanke postavil pogoj, da se do tedaj zbrani podatki izvozijo v primeru, da jih je vsaj 1.000.000, nato pa se pridobivanje podatkov nadaljuje. Tako se vsakič, ko se nabere 1 milijon podatkov, ti podatki izvozijo.

Pri izvozu podatkov sem naletel na dve manjši težavi, obe povezani z izbiro Excel formata za izvozne datoteke. Excel datoteka namreč ne podpira zapisa datuma in ure v taki obliki, kot je pridobljena iz funkcije `get_items()` (torej s podatkom o časovnem pasu UTC+0), zato sem moral nad atributoma `user.created_in_date` uporabiti funkcijo `tz_localize()` s parametrom `None` (NumFOCUS Inc., 2023a), da sem datum in uro “delokaliziral” (izbrisal sem podatek o časovnem pasu UTC+0, saj je bil za vse enak).

Na koncu pa sem uporabil funkcijo `to_excel()`, ki `DataFrame` izvozi v obliki Excel datoteke. Kot prvi parameter sem navedel relativno pot in ime, ki ga bo izvozna datoteka imela, drugi parameter pa funkciji narekuje, da mora pri izvozu uporabiti način pisanja `xlsxwriter`. V primeru, ko drugega parametra nisem navedel, je pri izvozu prišlo do napake, saj privzeti način pisanja `openpyxl` ne podpira prav dobro vseh emojijev in posebnih znakov, ki jih Twitter uporabniki vnašajo v svoje tvite. Po tem, ko sem `openpyxl` zamenjal z `xlsxwriter`, ni bilo z izvozom več nobenih težav.

5.2 Formatiranje podatkov

Pred samim začetkom opravljanja analize razpoloženja in generiranjem grafikonov sem moral poskrbeti, da bodo vsi podatki pravilno formatirani, obravnavani v pravem podatkovnem tipu ter zagotoviti, da ne bo med njimi posebnih anomalij, kot so npr. tviti, daljši od 280 znakov, prazne lokacije, manjkajoče vrednosti ipd.

V ta namen sem napisal kratko Python skripto, ki je odpravila vse pomanjkljivosti. Ko sem opravil raziskovalno analizo pridobljenih podatkov, sem ugotovil, da največji problem predstavljajo ravno izluščena besedila objav, saj so bila v velikem številu primerov daljša od 280 znakov. Ugotovil sem, da je do tega prišlo zaradi znakov `>`, `<`, `_x000D_`, ki so se v objavah pojavili kot upodobitve teksta na starejših različicah spletnih brskalnikov, ki teh znakov ne podpirajo (Quinn, 2023).

Zato sem najprej napisal funkcijo, ki je iz vseh tvitov odstranila vse emojije, ostale simbole in piktograme, zemljevidne simbole, zastave ter kitajske znake. Vsi ti znaki namreč zasedajo več kot en sam znak dolžine.

```
def remove_emojis(text):
    emojis = re.compile(
        "[
        \U0001F600-\U0001F64F"
        \U0001F300-\U0001F5FF"
        \U0001F680-\U0001F6FF"
        \U0001F1E0-\U0001F1FF"
        \U00002500-\U00002BEF"
        \U00002702-\U000027B0"
        \U00002702-\U000027B0"
        \U000024C2-\U0001F251"
        \U0001f926-\U0001f937"
        \U00010000-\U0010ffff"
        \u2640-\u2642"
        \u2600-\u2B55"
        \u200d"
        \u23cf"
        \u23e9"
        \u231a"
        \ufe0f"
        \u3030"
        ]+",
        re.UNICODE,
    )
    return re.sub(emojis, "", text)
```

Primer kode funkcije za odstranjevanje emojijev in ostalih posebnih znakov.

Nato sem napisal še nekaj ukazov, ki so odpravili še ostale pomanjkljivosti. Prazna uporabniška imena in lokacije sem zamenjal s praznim nizom, manjkajoče podatke o številu ogledov starejših objav z 0, skoke v novo vrstico v objavah sem zamenjal s presledkom, znake >, <, & z >, <, &, znak `_x000D_`, do katerega pride takrat, ko Excel napačno interpretira skok v novo vrstico, pa s presledkom. Ti znaki nimajo posebnega vpliva v analizi razpoloženja, vplivajo pa na samo dolžino tvita – zaradi teh znakov so imeli nekateri tviti dolžino 300 ali več znakov, čeprav jih Twitter dovoljuje največ 280. Na koncu sem nad vsemi tviti izvedel predhodno definirano funkcijo `remove_emojis()`, ki je iz objav odstranila vse emojije in ostale nedovoljene znake.

```
df["User name"] = df["User name"].fillna("")
df["User location"] = df["User location"].fillna("")
df["View count"] = df["View count"].fillna(0)
df["Tweet"] = df["Tweet"].replace(["\n"], " ")
df["Tweet"] = df["Tweet"].replace({"&gt;": ">"}, regex=True)
df["Tweet"] = df["Tweet"].replace({"&lt;": "<"}, regex=True)
df["Tweet"] = df["Tweet"].replace({"&amp;": "&"}, regex=True)
```

```
df["Tweet"] = df["Tweet"].replace({"_x000D_": " "}, regex=True)
df["Tweet"] = df["Tweet"].apply(lambda text: remove_emojis(text))
```

Primer kode za odpravljanje naštetih pomanjkljivosti.

5.3 Analiza razpoloženja objav

Po tem, ko sem pridobil vse željene podatke, sem pričel s pisanjem Python skripte, ki bi na vsakem zbranem tvitu opravila analizo razpoloženja. Po korakih, ki so bili navedeni v dokumentaciji izbrane metode analize razpoloženja (Hugging Face, 2023a), sem moral najprej vse označene profile, to so besede, ki se začnejo z znakom @, sledi pa uporabniško ime, zamenjati z @user, saj ima veliko število uporabnikov takšna uporabniška imena, v katerih so jasno izražena določena čustva, ki pa v resnici ne odražajo čustev avtorja tvita v trenutku, ko ga je ta objavil (npr. uporabnik @HappyWife ne bo objavljajl samo veselih vsebin). Potrebno je bilo odstraniti tudi vse zunanje URL povezave oz. jih nadomestiti z značko http zaradi istega razloga – spletna stran www.huggingface.co namreč ne objavlja izključno besedil, kjer je govora le o lepih in toplih objemih.

Glede na to, da sem pri delu pridobivanja podatkov sestavil poizvedbo tako, da sem avtomatično izključil vse tvite, ki so vsebovali zunanje URL povezave, bi lahko ta del tudi preskočil, a se je izkazalo, da bi na ta način spregledal »skrite« URL povezave. Med luščenjem tvitov sem namreč naletel tudi na objave, ki so vsebovale URL povezave, ki jih platforma Twitter ni zaznala kot take. Bile so zapisane brez predpone `https://in/aliwww`, zato se navadnim uporabnikom platforme Twitter ob kliku nanje ni zgodilo nič, a so vsekakor vsebovale podatek o neki zunanji spletni strani, ki ni bila relevantna za namene te raziskave.

Pred aplikacijo roBERTa model na izluščene podatke sem moral najprej izbrani model definirati in naložiti na lokalno napravo. Isti postopek sem moral kasneje ponoviti še za t.i. tokenizator (angl. tokenizer), namensko orodje, ki objave, podane v obliki niza, razstavi na posamezne besede, nato pa na posamezne črke, jih razvrsti v vrstni red glede na njihovo pojavitev v objavi, nato pa jih kodira in jim na podlagi rezultatov kodiranja priredi določeno število.

Tokenizator modela roBERTa je osnovan na BPE algoritmu. BPE algoritem je bil prvotno zasnovan kot kompresijski algoritem s primarno nalogo stiskanja podatkov, danes pa se ga večinoma uporablja pri obdelavi in tokeniziranju besedil za iskanje najboljše predstavitve besedila z najmanjšim številom žetonov. BPE algoritem deluje na sledeči način: koncem besed v nizu se doda poseben identifikator (</w>), ki označuje konec besede, nato pa se izračuna pogostost besed v danem nizu. Besede se nato razdeli na posamezne znake, nato pa se izračuna pogostost vseh znakov znotraj niza. Iz žetonov

znakov se potem ponavlja postopek štetja pogostosti zaporednih parov bajtov in njihovega združevanja, dokler se ne doseže vnaprej določene omejitve (Tyagi, 2021).

S tem, ko sem imel na lokalni napravi nameščen tokenizator in model za analizo razpoloženja, sem vsak tweet posebej najprej tokeniziral, nato pa na rezultatu tokenizatorja, ki je bil videti kot seznam celih števil (rezultati kodiranja), apliciral še roBERTa model. Po apliciranju roBERTa modela so rezultati izgledali kot seznam treh decimalnih števil, vsako število pa je prikazovalo utež negativnih, nevtralnih in pozitivnih čustev obravnavanega tvita. Ker je imel seznam za vsak tweet drugačno skalo, sem vse te sezname najprej normaliziral oz. pretvoril v odstotke tako, da je bila vsota vseh treh števil enaka 1. Ker sem sedaj razpolagal z odstotki, sam na podlagi najvišjega deleža izmed negativnih, nevtralnih in pozitivnih čustev določil, ali je obravnavani tweet negativen, nevtralen ali pozitiven. Iz tega razloga sem svoji podatkovni zbirki dodal nov stolpec, ki sem ga poimenoval *Sentiment*. Tako sem tweet z `[0.25, 0.48, 0.27]` deleži v stolpcu *Sentiment* označil z oznako *Neutral*, saj je delež nevtralnih čustev (0,48) večji od negativnega (0,25) in pozitivnega (0,27).

Pognal sem napisano skripto, a sem kmalu ugotovil, da bo ta proces zelo časovno zamuden in energetsko potraten, saj je morala centralna procesna enota računalnika uvoziti, predpripraviti (odstraniti imena uporabnikov in zunanje URL povezave), tokenizirati in aplicirati model na 1.073.786 zapisih, na koncu pa rezultate tudi izvoziti v novo datoteko. Skripto sem pognal večkrat, a se je vsakič njeno izvajanje po nekaj minutah prekinilo, saj je za vse te naloge zmanjkalo prostora v glavnem pomnilniku.

Iz tega razloga sem napisal še eno skripto, ki je združila obe datoteki s podatki, nato pa ju ločila na 2000 manjših datotek, vsako s približno 520 zapisi.

```
import pandas as pd
import numpy as np

df1 = pd.read_excel(".\\data-parsed-1.xlsx", index_col=0)
df2 = pd.read_excel(".\\data-parsed-2.xlsx", index_col=0)

df = pd.concat([df1, df2], ignore_index=True)
df = df.drop_duplicates(subset="Tweet ID")

index = 1
dataframes = np.array_split(df, 2000)
for dataframe in dataframes:
    dataframe.to_excel(f".\\data-splitted-{index}.xlsx",
engine="xlsxwriter")
    index += 1
```

Primer kode za ločevanje podatkovne zbirke na manjše dele.

Skripto za opravljanje analize razporeditve sem moral zaradi sprememb podatkovne zbirke rahlo spremeniti oz. ji dodati še zanko, ki bi vsako datoteko posebej uvozila, na njej opravila analizo, nato pa jo izvozila z isto zaporedno številko, ta postopek pa bi ponovila dva tisočkrat.

```
from transformers import AutoTokenizer
from transformers.models.auto.modeling_auto import
AutoModelForSequenceClassification
from scipy.special import softmax
import pandas as pd
import numpy as np

for i in range(1, 2000 + 1):
    df = pd.read_excel(f"\\data-splitted-{i}.xlsx", index_col=0)
    df.info()

    df["Tweet processed"] = df["Tweet"].apply(
        lambda string: " ".join([
            "@user" if word.startswith("@") and len(word) > 1 else word
            for word in str(string).split()
        ])
    )
    df["Tweet processed"] = df["Tweet processed"].apply(
        lambda string: " ".join([
            "http" if word.startswith("http") or word.startswith("www.")
            else word
            for word in string.split()
        ])
    )

    roberta = "cardiffnlp/twitter-roberta-base-sentiment"

    model = AutoModelForSequenceClassification.from_pretrained(roberta)
    tokenizer = AutoTokenizer.from_pretrained(roberta)

    df["Tweet encoded"] = df["Tweet processed"].apply(lambda string:
tokenizer(string, return_tensors="pt"))
    df["Output"] = df["Tweet encoded"].apply(lambda string: model(**string))

    df["Scores"] = df["Output"].apply(lambda x:
softmax(x[0][0].detach().numpy()))

    labels = ["Negative", "Neutral", "Positive"]

    for j in range(len(labels)):
        df[labels[j]] = df["Scores"].apply(lambda arr: arr[j])
        df[labels[j]] = df["Scores"].apply(lambda arr: arr[j])
        df[labels[j]] = df["Scores"].apply(lambda arr: arr[j])
        df["Sentiment"] = df["Scores"].apply(lambda arr:
```

```
labels[np.argmax(arr)])
```

```
df.to_excel(f".\\data-processed-{i}.xlsx", engine="xlsxwriter")
```

Primer kode za opravljanje analize razporeditve.

S tem, ko sem podatkovno zbirko razbil na več manjših delov, je bila težava pomanjkanja prostora v glavnem pomnilniku odpravljena. Proces opravljanja analize na podatkih se je uspešno zaključil, a je bil vsekakor kar zahteven za moj računalnik s osemjedro centralno procesno enoto (štiri jedra s frekvenco 3228MHz ter štiri jedra s frekvenco 2064MHz), saj je trajal nekaj manj kot 24 ur.

5.4 Razvoj skripte za grafični prikaz rezultatov

Po opravljeni analizi razporeditve na vseh 1.073.786 zapisih sem pričel s pisanjem zadnjega dela Python kode, ki bi pridobljene rezultate očistila, izračunala in grafično prikazala. Glavni Python modul za risanje grafiknov in prikaz grafičnih podatkov je Matplotlib, zato sem na spletni strani z dokumentacijo iskal take grafikone (The Matplotlib development team, 2023), ki bi lahko prišli v poštev za namen te raziskave.

Pred razvojem skripte sem na papirju definiral dvajset grafiknov, ki jih je smiselno vključiti v raziskavo z namenom olajšanja prikaza rezultatov oz. povečanja njihove intuitivnosti. Zato bom v sledečih odstavkih predstavil posamezne uporabljene grafikone, koda, ki grafikone generira, pa je predstavljena v prilogi 1.

Najprej sem vse podatkovne datoteke znova združil v eno samo datoteko. V ta namen sem napisal for zanko, ki se je dva tisočkrat ponovila, vsakič pa je prebrala zaporedno podatkovno datoteko in jo prilepila na konec prejšnje podatkovne datoteke. Na tak način sem znova združil vse podatke v eno samo datoteko.

```
import pandas as pd

df = pd.read_excel(".\\data-processed-1.xlsx", index_col=0)

for i in range(2, 2000 + 1):
    dataframe = pd.read_excel(f".\\data-processed-{i}.xlsx", index_col=0)
    df = pd.concat([df, dataframe], ignore_index=True)
```

Primer kode za združevanje podatkovnih datotek.

Obenem sem definiral tudi dva seznama: prvega za hranjenje oznak rezultatov, drugega pa za barve, ki jih bom v grafiknih uporabil zato, da med sabo ločim negativne, nevtralne in pozitivne tvite. Barve sem izbral poljubno.

```
labels = ["Negative", "Neutral", "Positive"]
colors = ["#FF6663", "#D0CD94", "#53CAA7"]
```

Primer kode za definiranje seznamov oznak in barv.

Nato sem začel s pisanjem kode, ki bo prikazala vse željene grafikone. Sledijo opisi vseh posameznih grafikonov, koda za njihovo generiranje pa je v prilogi 1.

Grafikon za prikaz deleža negativnih, nevtralnih in pozitivnih tvitov (slika 1)

Prvi grafikon je tortni grafikon, ki prikazuje delež negativnih, nevtralnih in pozitivnih tvitov. Grafikonu sem dodal prej definirane oznake in barve, deleže pa zaokrožil na eno decimalno mesto (koda v prilogi 1).

Grafikon za prikaz besednega oblaka najbolj pogosto uporabljenih besed (slika 2)

Drugi grafikon prikazuje najbolj pogosto uporabljene besede v vseh tvitih. V ta namen sem najprej naložil modul in ustvaril klic funkcije, ki mi je vrnila seznam vseh angleških praznih besed, v nadaljevanju pa sem te besede, poleg vseh ločil, izločil iz niza, ki je združeval nize vseh tvitov. S pomočjo modula `WordCloud` sem nato prikazal grafikon n najpogostejše uporabljenih besed in se odločil, da bo v mojem primeru $n = 50$ (koda v prilogi 1).

Grafikon za prikaz besednega oblaka najbolj pogosto uporabljenih ključnikov (slika 3)

Tretji grafikon je zelo podoben drugemu, saj ne prikazuje najbolj pogostih besed, temveč najbolj pogosto uporabljene ključnike. Zato sem napisal kodo, ki združi vse tvite v en sam niz, vse velike tiskane črke pretvori v male (zato, da ne bo razlik med `#Ukraine` ter `#ukraine`), nato pa iz tega niza sestavi nov niz, ki vsebuje le ključnike (koda v prilogi 1).

Grafikon za prikaz besednega oblaka najbolj pogosto omenjenih uporabniških računov (slika 4)

Tudi četrti grafikon je podoben prejšnjima dvema, le da v tem primeru prikazujem najbolj pogosto omenjene uporabniške račune znotraj posameznih objav (koda v prilogi 1).

Grafikon za prikaz števila objavljenih tvitov glede na dan objave (slika 5)

Peti grafikon prikazuje število objavljenih tvitov na določen dan za obdobje od 24. februarja 2022 do 24. februarja 2023. Zato sem v stolpec `Date` vnesel datum objave tvita (brez podatka o uri objave), na podlagi tega pa ustvaril seznam, ki je na vsakem

zaporednem mestu nosil podatek o številu tvitov, objavljenih na določen dan (število 512 na indeksu 0 pomeni, da je bilo 24. februarja 2022 objavljenih 512 tvitov) (koda v prilogi 1).

Grafikon za prikaz števila objavljenih tvitov glede na uro objave (slika 6)

Šesti grafikon je osnovan na petem, le da namesto dni prikazuje ure objav tvitov (koda v prilogi 1).

Grafikon za prikaz števila objav glede na njihovo razpoloženje in dan objave (slika 7)

Sedmi grafikon prikazuje število objav na določen dan, a jih loči glede na njihov sentiment. Negativne, nevtralne in pozitivne objave bodo tako prikazane na istem grafikonu, a vsaka s črto s svojo barvo iz seznama, definiranega na samem začetku (koda v prilogi 1).

Grafikon za prikaz razporeditve dolžine objav glede na njihovo razpoloženje (slika 8)

Osmi grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano dolžine negativnih, druga nevtralnih, tretja pa pozitivnih tvitov (koda v prilogi 1).

Grafikon za prikaz razporeditve števila komentarjev objav glede na njihovo razpoloženje (slika 9)

Deveti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila komentarjev negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (angl. outliers) (koda v prilogi 1).

Grafikon za prikaz razporeditve števila delitev objav glede na njihovo razpoloženje (slika 10)

Deseti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila delitev negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (koda v prilogi 1).

Grafikon za prikaz razporeditve števila všečkov objav glede na njihovo razpoloženje (slika 11)

Enajsti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila všečkov negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (koda v prilogi 1).

Grafikon za prikaz razporeditve števila ogledov objav glede na njihovo razpoloženje (slika 12)

Dvanajsti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila ogledov negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. Grafikon vključuje le ogledske zapise, ki so bili objavljeni 15. decembra ali naprej (torej od točke, ko je Twitter uvedel funkcijo "view count") oziroma od indeksa 980233 dalje. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (koda v prilogi 1).

Grafikon za prikaz razporeditve števila sledilcev avtorjev objav glede na njihovo razpoloženje (slika 13)

Trinajsti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila sledilcev avtorjev negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (koda v prilogi 1).

Grafikon za prikaz razporeditve števila sledenja avtorjev objav glede na njihovo razpoloženje (slika 14)

Štirinajsti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila sledenih profilov avtorjev negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (koda v prilogi 1).

Grafikon za prikaz razporeditve števila objav avtorjev glede na njihovo razpoloženje (slika 15)

Petnajsti grafikon sestavljajo tri škatle z brki: prva prikazuje kvartile in mediano števila objav avtorjev negativnih, druga nevtralnih, tretja pa pozitivnih tvitov. V grafikon niso vključeni osamelci oz. izstopajoče vrednosti (koda v prilogi 1).

Grafikon za prikaz deleža verificiranih avtorjev objav glede na njihovo razpoloženje (slika 16)

Šestnajsti grafikon je sestavljen iz treh stolpcev, ki prikazujejo delež uporabnikov z verificiranim uporabniškim računom glede na razpoloženje njihovih objav. Vsak stolpec

sestavljata delež verificiranih uporabnikov spodaj in delež neverificiranih uporabnikov zgoraj, ti skupaj pa tvorijo celoto. Vsak stolpec prikazuje določeno razporeditev: prvi negativno, drugi nevtrarno, tretji pa pozitivno (koda v prilogi 1).

Grafikon za prikaz seštevek vseh komentarjev, delitev in všečkov glede na razporeditev objav (slika 17)

Sedemnajsti grafikon sestavlja 9 stolpcev. Trije stolpci prikazujejo seštevek vseh komentarjev negativnih, nevtrarnih in pozitivnih objav, drugi trije stolpci prikazujejo seštevek vseh delitev negativnih, nevtrarnih in pozitivnih objav, zadnji trije pa prikazujejo seštevek vseh všečkov negativnih, nevtrarnih in pozitivnih objav. Podatka o ogledih tvitov nisem vključil v grafikon, saj tega podatka nimam za prvih devet mesecev spopadov, poleg tega pa bi vključitev tega podatka tako spremenila skalo, da bi bili vsi ostali stolpci nevidni (logično je namreč, da je ogledov veliko več kot vseh ostalih interakcij skupaj) (koda v prilogi 1).

Grafikon za prikaz pogostosti uporabe besed “war” in “special military operation” znotraj objav glede na njihov datum objave (slika 18)

Osemnajsti grafikon prikazuje prisotnost besed “war” in “special military operation” v posameznih tvitih skozi čas. Za pravilen prikaz sem ustvaril dva seznama: prvi hrani podatek o pogostosti besede “war”, drugi pa “special military operation”. Zaradi razlikovanja med nizoma “war” in “War” sem vse tvite spravil v male tiskane črke, da bo obe besedi obravnaval na enak način (koda v prilogi 1).

Grafikon za prikaz števila uporabniških računov glede na leto njihovega odprtja (slika 19)

Devetnajsti grafikon prikazuje obdobje ustvarjanja uporabniških računov avtorjev objav v letih glede na razporeditev zadnjega objavljenega tvita. Upoštevana so leta od 2006 do vključno 2023, torej celotno obdobje obstoja platforme Twitter (koda v prilogi 1).

Grafikon za prikaz zemljevida razporeditve (slika 20)

Zadnji, dvajseti grafikon, prikazuje lokacije avtorjev objav na zemeljski površini glede na razporeditev njihovih objav. Za upodobitev lokacije na zemljevid je potrebno poznati njene natančne geografske koordinate, iskanje koordinat pa se izkaže za zelo zamuden proces. Najprej zato, ker ima približno polovica avtorjev objav podatek o lokaciji izpolnjen in viden v podatkih uporabniškega računa, ampak tudi zato, ker ta podatek ne prikazuje vedno natančne lokacije ali lokacije nasploh. Veliko uporabnikov namreč v polje, ki je namenjeno lokaciji, vpisuje druge stvari (npr. “proud american”, “she/him”, “Earth is my home”), nekateri pa imajo dejansko vpisano lokacijo, a je ta preveč splošna,

da bi lahko zanjo našel prave geografske koordinate (npr. "Odessa/London", "USA", "European Union").

Zato sem najprej napisal še eno dodatno skripto, ki preveri vse vpisane lokacije uporabnikov in jih vpiše na spletno orodje Nominatim, v primeru zadetka pa podane geografske koordinate pripiše uporabniku v stolpca Latitude in Longitude. V primeru, da je več zadetkov za podani naslov, izbere prvega, v primeru, da zadetkov ni, pa pusti ti dve polji prazni.

```
import pandas as pd
import numpy as np
from geopy.geocoders import Nominatim

geolocator = Nominatim(user_agent="magistrska")
for i in range(1, 2000 + 1):
    df = pd.read_excel(f".\\data-processed-{i}.xlsx", index_col=0)
    df["User location"] = df["User location"].fillna("")

    for index, row in df.iterrows():
        address = str(row["User location"])
        if address != "":
            try:
                location = geolocator.geocode(address, exactly_one=True,
                timeout=2)

                if location is not None:
                    df.at[index, "Latitude"] = location.latitude
                    df.at[index, "Longitude"] = location.longitude
            except Exception:
                pass

    df.to_excel(f".\\data-location-{i}.xlsx", engine="xlsxwriter")
```

Primer kode za pridobitev geografskih koordinat lokacij avtorjev objav.

Po tem, ko sem pridobil geografske koordinate za lokacije uporabnikov iz vseh dva tisoč podatkovnih datotek, sem napisal še kodo, ki prikaže zemljevid, na zemljevidu pa obarvane točke glede na pridobljene geografske koordinate. Točke so obarvane glede na razpoloženje objave (koda v prilogi 1).

6 ANALIZA REZULTATOV

V tem poglavju bom predstavil rezultate opravljene analize, jih analiziral in primerjal z rezultati javnomnenjskih anket iz sorodnega obdobja ter povzel ključne ugotovitve. Na koncu bom namenil še par besed vsem omejitvam opravljene raziskave ter podal možne izboljšave in napotke za nadaljnje delo.

6.1 Predstavitev rezultatov

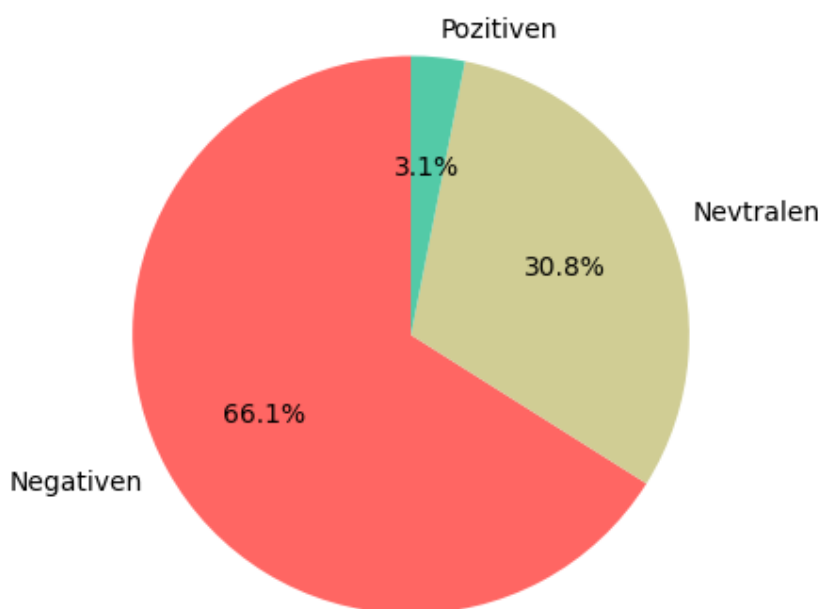
V tem podpoglavju bom prikazal in predstavil vse izrisane grafikone iz prejšnjega poglavja, v naslednjem podpoglavju pa jih bom komentiral in poskušal utemeljiti razloge, ki prikazujejo javno mnenje, razbrano iz podatkov.

Grafikon za prikaz deleža negativnih, nevtralnih in pozitivnih tvitov

Iz prvega grafikona (slika 1) je razvidno, da je iz skoraj dveh tretjin vseh analiziranih tvitov možno razbrati negativno mnenje o vojni v Ukrajini, manj kot tretjina tvitov ima o vojni mešane občutke, torej niti negativne niti pozitivne, dobri 3% tvitov pa izražajo večinoma pozitivna čustva, mnenja in besede.

Slika 1: Delež negativnih, nevtralnih in pozitivnih tvitov

Delež negativnih, nevtralnih in pozitivnih tvitov



Vir: lastno delo.

Grafikon za prikaz besednega oblaka najbolj pogosto uporabljenih besed

Iz drugega grafikona (slika 2) je možno razbrati, da so najbolj pogosto uporabljene besede, poleg logičnih "russia" in "ukraine" ter pridevnikov "russian" in "ukranian", še besede "war", "nato", "europe", "usa", "china", "peace", "putin", "biden", "nuclear" in "sanction".

Slika 2: Besedni oblak najbolj pogosto uporabljenih besed



Vir: lastno delo.

Grafikon za prikaz besednega oblaka najbolj pogosto uporabljenih ključnikov

Tretji grafikon (slika 3) prikazuje najbolj pogosto uporabljene ključnike znotraj posameznih objav. Tudi v tem primeru sta, po pričakovanjih, ključnika `#russia` in `#ukraine` na samem vrhu lestvice, sledijo pa ključniki `#ukraineinvasion`, `#ukrainerussiawar`, `#russiaukrainewar`, `#ukrainewar`, `#russianukrainianwar`, `#russiaukraineconflict` ter `#putin`. Bolj zanimivi, a manj pogosto uporabljeni ključniki, so `#stoprussia`, `#worldwar3`, `#nato`, `#zelensky` ter `#russiaisaterroriststate`.

Slika 3: Besedni oblak najbolj pogosto uporabljenih ključnikov



Vir: lastno delo.

Grafikon za prikaz besednega oblaka najbolj pogosto omenjenih uporabniških računov

Iz četrtega grafikona (slika 4) je razvidno, da so bili v tvitih največkrat sledeči profili: @potus (President Of The United States – uradni Twitter profil ameriškega predsednika), @nato (zveza Nato), @un (United Nations – Organizacija združenih narodov),

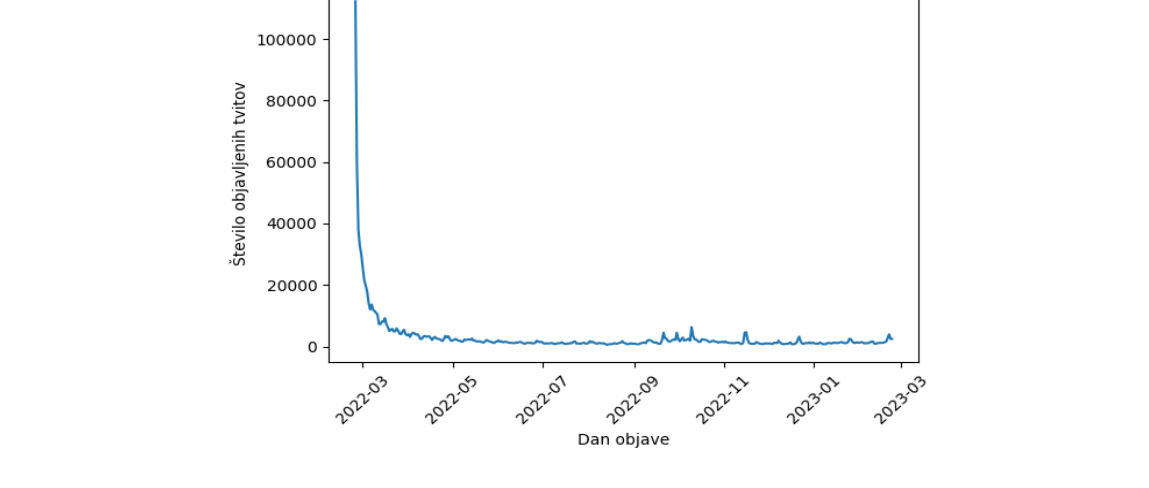
Slika 4: Besedni oblak najbolj pogosto omenjenih uporabniških računov

Besedni oblak najbolj pogosto omenjenih uporabniških računov



Slika 5: Število objavljenih tvitov glede na dan objave

Število objavljenih tvitov glede na dan objave



Grafikon za prikaz števila objavljenih tvitov glede na uro objave

Šesti grafikon (slika 6) prikazuje število objavljenih tvitov glede na uro njihove objave. Iz njega je razvidno, da je najmanj tvitov objavljenih v nočnih urah in da je bila večina tvitov objavljena okoli 15:00 ure po UTC+0 času, kar pomeni okoli 16:00 ure po zimskem oz. 17:00 ure po poletnem času v Sloveniji.

Slika 6: Število objavljenih tvitov glede na uro objave

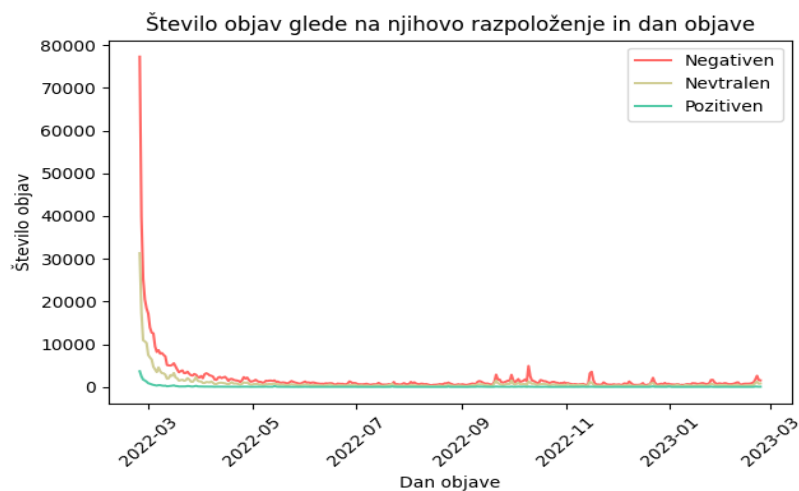


Vir: lastno delo.

Grafikon za prikaz števila objav glede na njihovo razpoloženje in dan objave

Iz sedmega grafikona (slika 7) je znova razvidno število objav na določen dan, a tokrat so negativni, nevtralni in pozitivni tviti posebej označeni. Iz grafikona lahko razberemo, da od prvega dne do prve obletnice spopadov prevladujejo negativni tviti, sledijo nevtralni, na zadnjem mestu pa pozitivni. Pozitivnih tvitov je bilo največ v prvem dnevu vojne, od takrat dalje pa so skoraj konstantno ob ničli. Iz tega grafikona opazim, da so negativni in nevtralni tviti tisti, ki so odgovorni za skoke v obdobju od polovice septembra 2022 do konca leta 2022 ter ob bližanju prve obletnice vojne.

Slika 7: Število objav glede na njihovo razpoloženje in dan objave

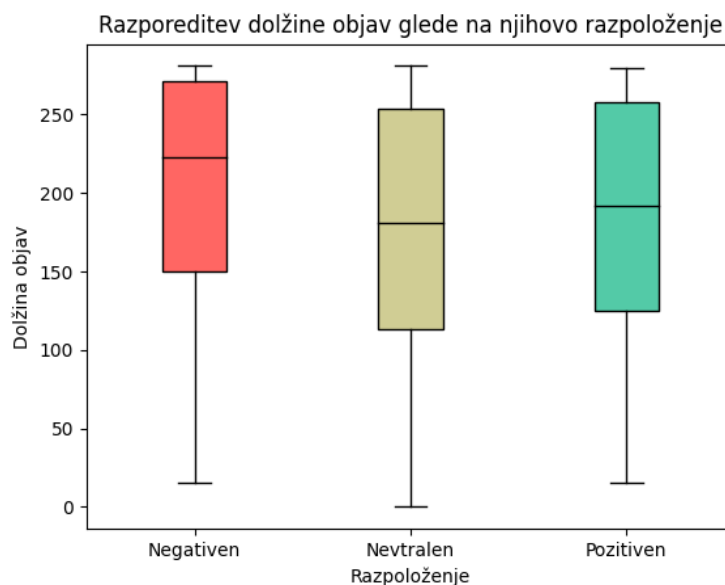


Vir: lastno delo.

Grafikon za prikaz razporeditve dolžine objav glede na njihovo razpoloženje

Osmi grafikon (slika 8) prikazuje razporeditev dolžine tvitov glede na njihovo razpoloženje. Iz grafikona je mogoče razbrati, da je mediana nevtralnih in pozitivnih tvitov zelo podobna (približno 180 znakov), mediana negativnih tvitov pa je bistveno večja (približno 230 znakov).

Slika 8: Razporeditev dolžine objav glede na njihovo razpoloženje

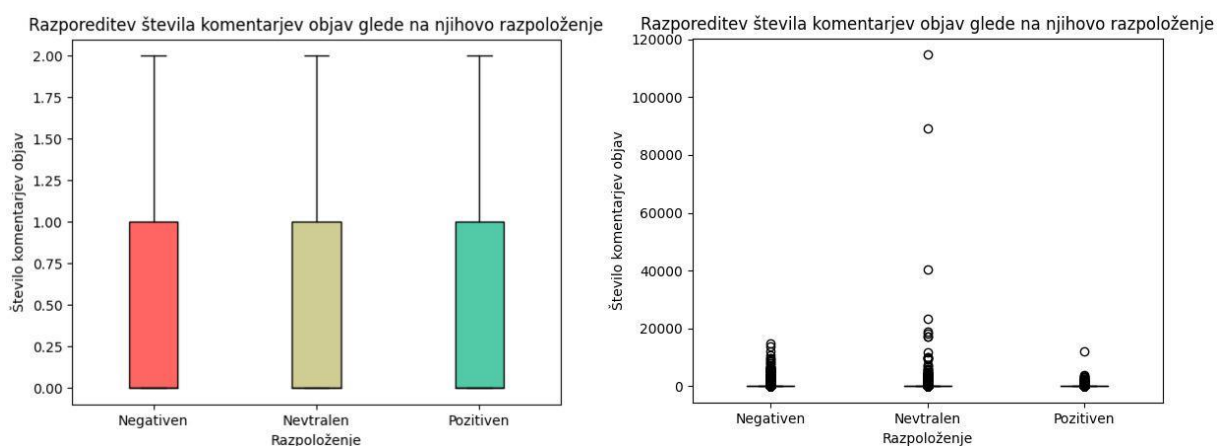


Vir: lastno delo.

Grafikon za prikaz razporeditve števila komentarjev objav glede na njihovo razpoloženje

Deveti grafikon (slika 9) prikazuje število komentarjev objav glede na razpoloženje tvita. Iz grafikona, iz katerega sem prvotno izključil osamelce, je mogoče razbrati, da so razporeditve negativnih, nevtralnih in pozitivnih tvitov popolnoma enake, zato sem se odločil, da prikažem raje grafikon z osamelci vred. Na tem drugem grafikonu lahko razberem, da imajo največ komentarjev nevtralni tviti.

Slika 9: Razporeditev števila komentarjev objav glede na njihovo razpoloženje (levo brez osamelcev, desno z osamelci)

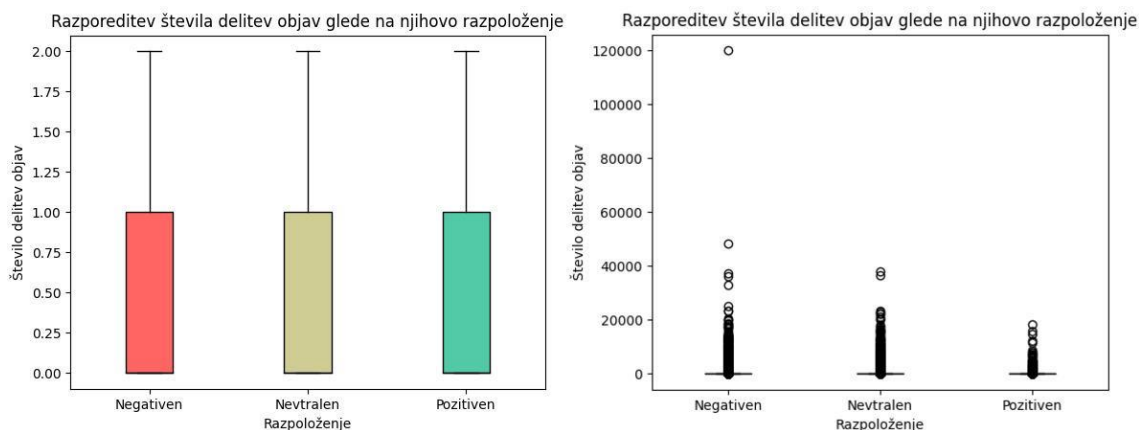


Vir: lastno delo.

Grafikon za prikaz razporeditve števila delitev objav glede na njihovo razpoloženje

Pri desetem grafikonu (slika 10), ki prikazuje razporeditev števila delitev tvitov glede na njihovo razpoloženje, sem naletel na enako težavo kot pri prejšnjem, zato sem se znova odločil prikazati oba grafikona. Iz grafikona brez osamelcev razberem, da je bila izmed vseh objav najbolj deljena objava z negativnim razpoloženjem.

Slika 10: Razporeditev števila delitev objav glede na njihovo razpoloženje (levo brez osamelcev, desno z osamelci)

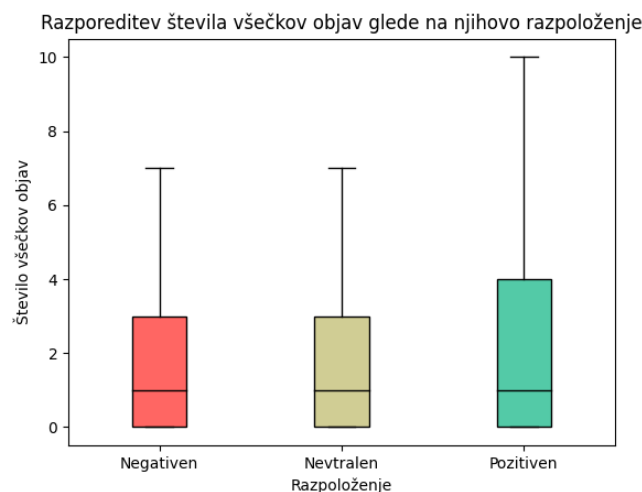


Vir: lastno delo.

Grafikon za prikaz razporeditve števila všečkov objav glede na njihovo razpoloženje

Iz enajstega grafikona (slika 11), ki prikazuje razporeditev števila všečkov tvitov glede na njihovo razpoloženje, razberem, da so bili najbolj všečkani pozitivni tviti.

Slika 11: Razporeditev števila všečkov objav glede na njihovo razpoloženje



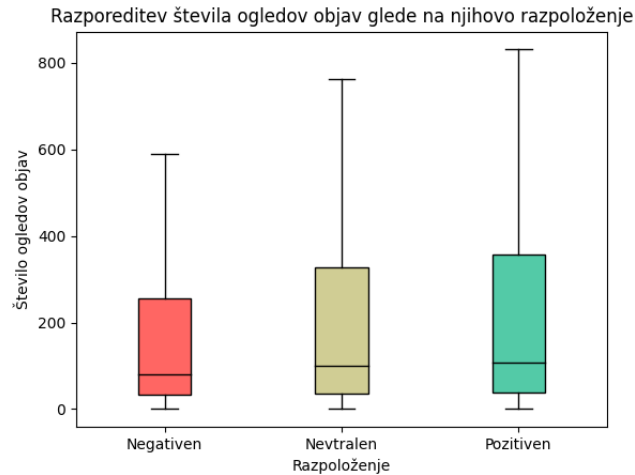
Vir: lastno delo.

Grafikon za prikaz razporeditve števila ogledov objav glede na njihovo razpoloženje

Dvanajsti grafikon (slika 12) prikazuje razporeditev števila ogledov tvitov (objavljenih od 15. decembra 2022 dalje) glede na njihovo razpoloženje. Zanimivo je, da imajo

negativni tviti najmanj ogledov, sledijo nevtralni tviti, tviti s pozitivnim razpoloženjem pa imajo največ ogledov.

Slika 12: Razporeditev števila ogledov objav glede na njihovo razpoloženje

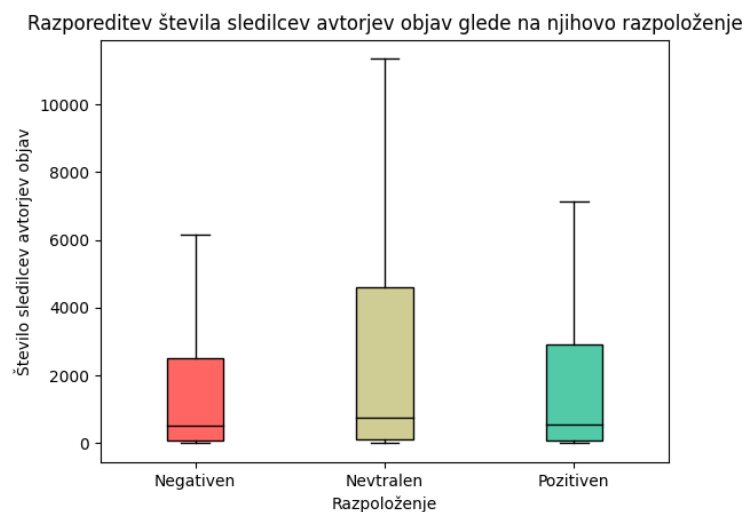


Vir: lastno delo.

Grafikon za prikaz razporeditve števila sledilcev avtorjev objav glede na njihovo razpoloženje

Iz trinajstega grafikona (slika 13), ki prikazuje razporeditev števila sledilcev, ki jih imajo avtorji negativnih, nevtralnih in pozitivnih tvitov, je mogoče razbrati, da imajo največ sledilcev avtorji nevtralnih objav, sledijo jim avtorji pozitivnih objav, na koncu pa so avtorji negativnih objav.

Slika 13: Razporeditev števila sledilcev avtorjev objav glede na njihovo razpoloženje

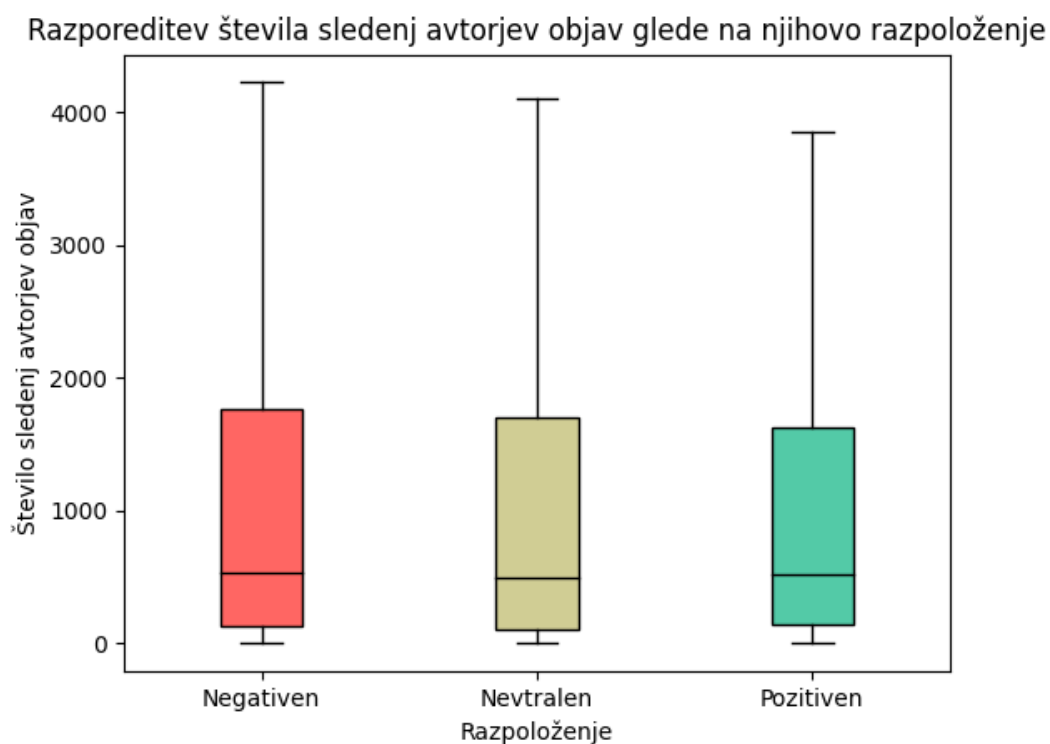


Vir: lastno delo.

Grafikon za prikaz razporeditve števila sledenj avtorjev objav glede na njihovo razpoloženje

Štirinajsti grafikon (slika 14) prikazuje razporeditev števila sledenih profilov s strani avtorjev negativnih, nevtralnih in pozitivnih objav. Iz grafikona je možno videti, da ni velikih razlik med številom sledenih profilov avtorjev glede na razpoloženje njihovih objav, le da največ profilom sledijo avtorji negativnih tvitov, najmanj pa avtorji pozitivnih.

Slika 14: Razporeditev števila sledenj avtorjev objav glede na njihovo razpoloženje

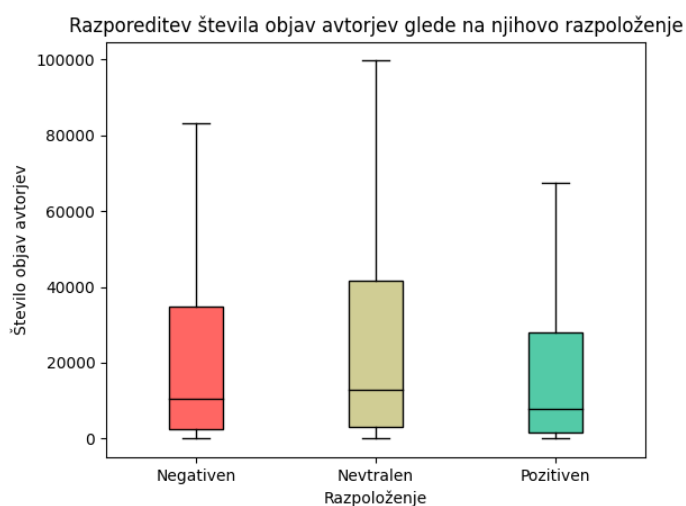


Vir: lastno delo.

Grafikon za prikaz razporeditve števila objav avtorjev glede na njihovo razpoloženje

Petnajsti grafikon (slika 15), iz katerega je vidna razporeditev števila objav avtorjev negativnih, nevtralnih in pozitivnih tvitov, razkriva dejstvo, da so na platformi Twitter najbolj aktivni avtorji nevtralnih objav, najmanj aktivni pa avtorji pozitivnih.

Slika 15: Razporeditev števila objav avtorjev glede na njihovo razpoloženje

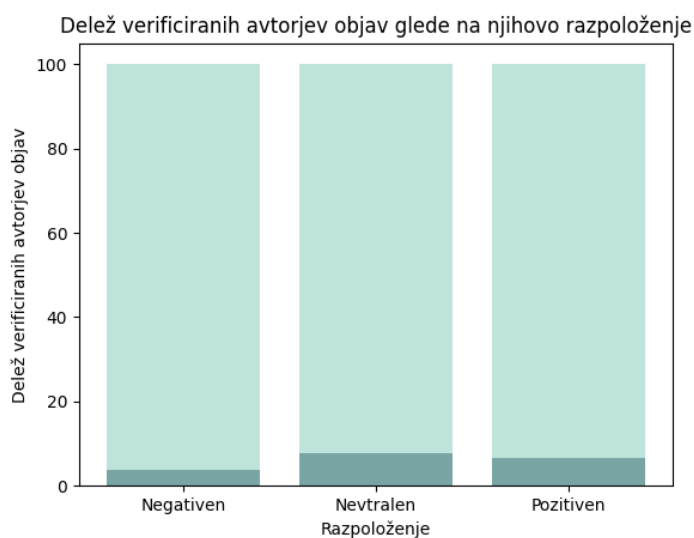


Vir: lastno delo.

Grafikon za prikaz deleža verificiranih avtorjev objav glede na njihovo razpoloženje

Šestnajsti grafikon (slika 16) prikazuje delež verificiranih uporabniških računov avtorjev glede na razpoloženje njihovih objav. Iz grafikona je mogoče razbrati, da imajo najnižji odstotek verificiranih profilov avtorji negativnih objav, avtorji nevtralnih in pozitivnih pa imajo skoraj dvakrat toliko višjega. Avtorji nevtralnih objav imajo v povprečju največji odstotek verificiranih uporabnikov.

Slika 16: Delež verificiranih avtorjev objav glede na njihovo razpoloženje

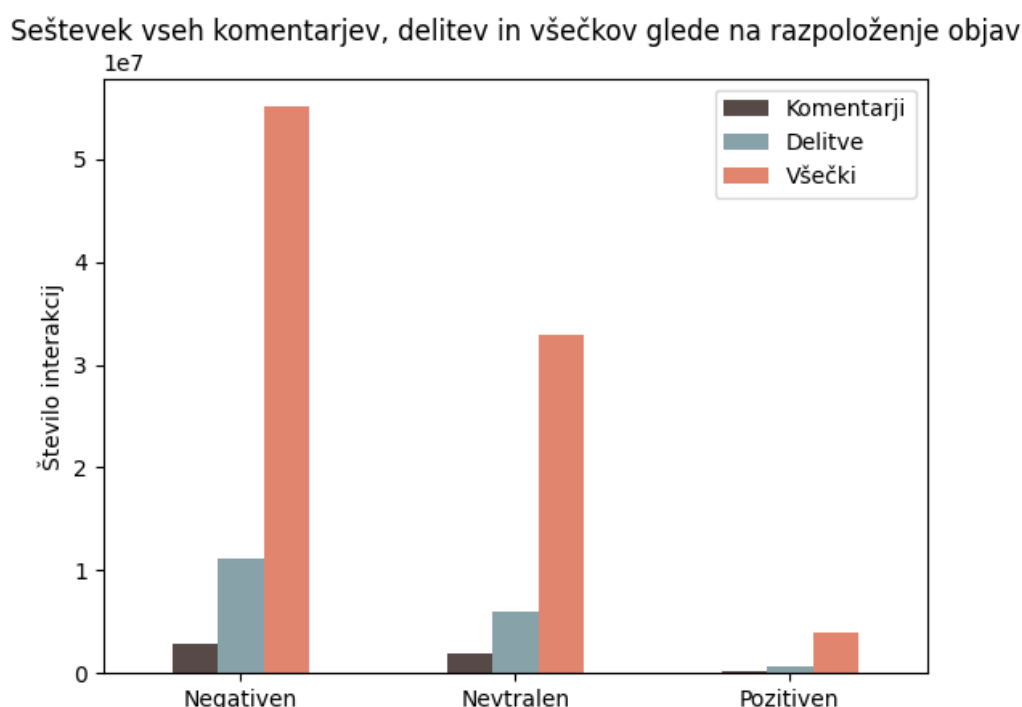


Vir: lastno delo.

Grafikon za prikaz seštevka vseh komentarjev, delitev in všečkov glede na razpoloženje objav

Sedemnajsti grafikon (slika 17) prikazuje skupno število komentarjev, delitev in všečkov tvitov glede na njihovo razpoloženje. Negativni tviti imajo tako največje skupno število interakcij, pozitivni pa najmanj, a rezultati niso presenetljivi, saj gre pri tem predvsem za dejstvo, da je negativnih objav 66,1 %, pozitivnih pa le 3,1 %.

Slika 17: Seštevke vseh komentarjev, delitev in všečkov glede na razpoloženje objav

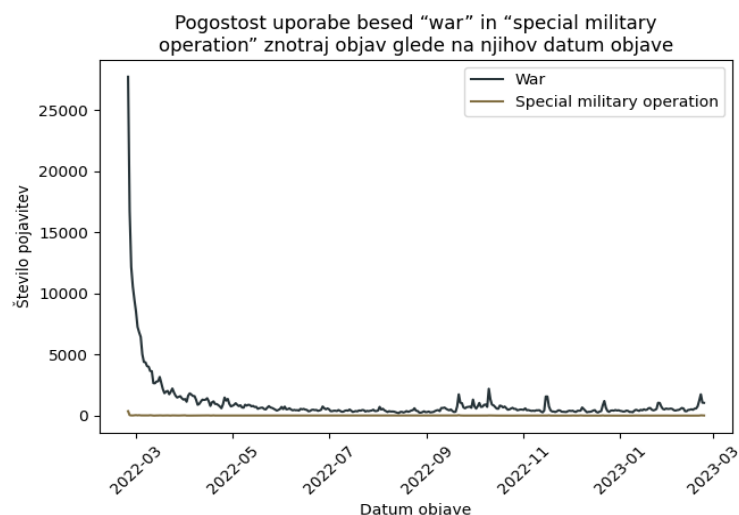


Vir: lastno delo.

Grafikon za prikaz pogostosti uporabe besed “war” in “special military operation” znotraj objav glede na njihov datum objave

Osemnajsti grafikon (slika 18) prikazuje pogostost uporabe besed “war” in “special military operation” v tvitih v prvem letu spopadov. Iz grafikona je razvidno, da je bila beseda “special military operation” uporabljena predvsem v prvih dneh spopadov in je zatem zelo kmalu izginila iz objav.

Slika 18: Pogostost uporabe besed “war” in “special military operation” znotraj objav glede na njihov datum objave

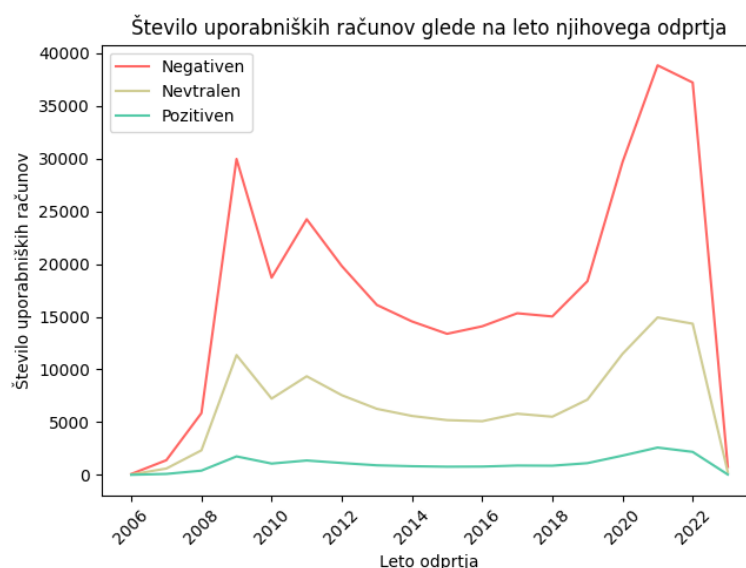


Vir: lastno delo.

Grafikon za prikaz števila uporabniških računov glede na leto njihovega odprtja

Devetnajsti grafikon (slika 19) prikazuje leto odprtja uporabniških računov avtorjev tvitov glede na njihovo razporejenost. Iz grafikona lahko vidimo, da obdobja ustvarjanja novih uporabniških računov sovpadajo med vsemi tremi skupinami, in sicer v letih 2009, 2011, 2021 ter 2022. Večina uporabniških računov avtorjev izluščenih tvitov je bila ustvarjena v letih 2021 in 2022.

Slika 19: Število uporabniških računov glede na leto njihovega odprtja

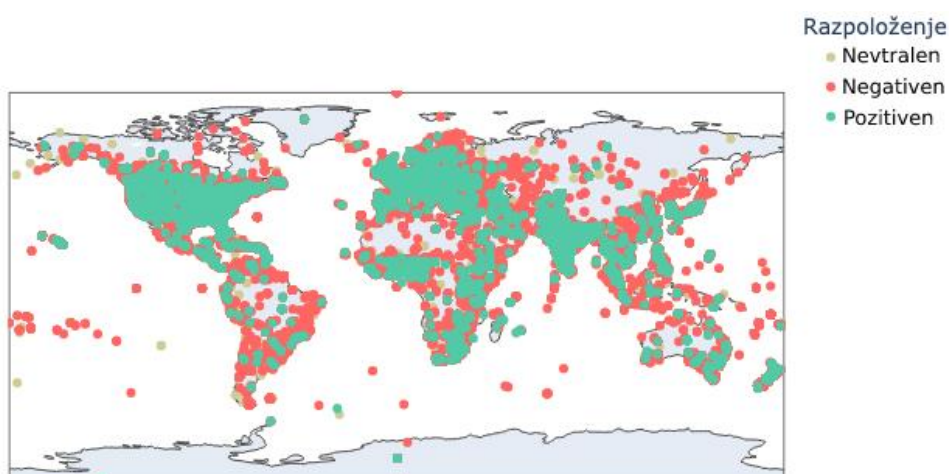


Vir: lastno delo.

Grafikon za prikaz zemljevida razporeditve

Dvajseti grafikon (slika 20) prikazuje lokacije avtorjev tvitov glede na njihovo razporeditve. V grafikon so vključeni le avtorji, ki imajo v svojem uporabniškem računu vnesen podatek o lokaciji. Grafikon prikaže precej zanimivo sliko: avtorji negativnih in nevtralnih tvitov so razpršeni po celem svetu, avtorji pozitivnih tvitov pa se nahajajo večinoma v ZDA, Evropi, Indiji ter subsaharski Afriki.

Slika 20: Zemljevid razporeditve



Vir: lastno delo.

6.2 Ključne ugotovitve

V tem podglavju so predstavljene ključne ugotovitve, ki izhajajo iz opravljene analize.

V javnost so že tedne in mesece pred 24. februarjem 2022 prihajala opozorila s strani ameriških in evropskih obveščevalnih agencij, da se oborožene sile Ruske federacije pripravljajo na invazijo, a večina javnosti teh ni jemala preveč resno. Potem pa je napočil 24. februar, ko je vojska Ruske federacije prečkala državno mejo z Ukrajino, s tem pa začela invazijo na to državo. Z ozirom na to, da je bilo v prvem dnevu spopadov objavljenih preko 110.000 tvitov oz. več kot 10 % tvitov celotnega zajetega obdobja (slika 5), od katerih je bila velika večina do vojne kritičnih ali nevtralnih, pozitivnih je bilo zelo malo (slika 7), je mogoče sklepati, da je bila splošna javnost presenečena nad nepričakovanim razvojem dogodkov.

Na podlagi zemljevida razporeditve sklepam (slika 20), da je po celem svetu prevladovalo kritično mnenje glede vojne, predvsem v Ukrajini in Zahodnem svetu, a tudi v Rusiji. Zanimiv se zdi podatek, da je le malokdo verjel retoriki Kremlja, ki se je vseskozi vzdrževala uporabe besede “vojna” in spopade opisovala kot “posebno vojaško

operacijo". To sklepam na podlagi dejstva, da je bila besedna zveza "posebna vojaška operacija" prisotna le v nekaj sto tvitih v prvih dneh spopadov, za tem pa je praktično izginila in je ni več nihče uporabljal (slika 18). To pripisujem dejstvu, da se je javnost glede na to, da je večina javnega mnenja na platformi Twitter proti vojni oz. ima o njej negativno mnenje (slika 1), odločila raje za besedo "vojna", saj se hoče nekako oddaljiti od propagande in tonov iz Kremlja. Res pa je, da je beseda "war" bistveno krajša od "special military operation", kar verjetno tudi vpliva na izbiro besed na platformi, kjer ima lahko vsaka objava le omejeno število znakov.

Kar se tiče same vsebine tvitov – je težko posplošiti besedila več kot milijon objav v enem samem odstavku. Zato pa si bom pomagal z rezultati iz grafikonov najbolj uporabljenih besed in ključnikov ter najbolj omenjenih profilov (slika 2, slika 3, slika 4). Uporabniki so zelo pogosto uporabljali besede "war", "military", "nato", "usa", "time", "end", "stop", "nuclear", "people", "support", "west", "peace". Res je, da so te besede izvzete iz konteksta, a se mi zdi, da lahko iz njih sestavim par smiselnih misli: potrebna je pomoč Zahoda, bodisi humanitarna bodisi vojaška pomoč, še bolj pa je potreben konec spopadov in mir, saj si nihče ne želi konflikta z jedrskim orožjem.

Poleg ključnikov, ki opisujejo trenutni potek dogodkov (npr. #russiaukrainewar ali #ukraineinvasion), je veliko tudi takih, ki povzemajo mnenje avtorja tvita, to pa so #stopputin, #stoprussia, #putinwarcriminal, #russiaisaterroriststate, #standwithukraine, #stopwar, #wwiii ter #worldwar3. Ti ključniki sodijo med najbolj pogosto uporabljene ključnike, vsi obsojajo dejanja predsednika Ruske federacije in njihovih oboroženih sil ter držijo z Ukrajino, vsi bi radi zaključek spopadov, eni pa se očitno tudi bojijo, da bo splet dogodkov tak prej ali slej privedel do tretje svetovne vojne.

Zanimivo, a ne presenetljivo, se mi zdi tudi dejstvo, da so najpogostejše omenjeni profili znotraj obravnavanih tvitov profili politikov iz celega sveta. Omenjeni so politiki evropskih držav (nemški kancler, francoski predsednik, bivši predsednik ter bivša predsednica vlade Združenega kraljestva), politiki evropskih institucij (predsednica evropske komisije, visoki predstavnik Unije za zunanje zadeve, evropski parlament, evropska komisija), politične figure iz Severne Amerike (ameriški predsednik, kanadski predsednik vlade), osebnosti in ustanove s področja obrambe (generalni sekretar zveze NATO, ameriški obrambni sekretar, ukrajinsko ministrstvo za obrambo) ter še druge mednarodne, evropske in azijske politične osebnosti in ustanove (generalni sekretar OZN, rusko ministrstvo za zunanje zadeve, ukrajinsko ministrstvo za zunanje zadeve, indijski predsednik vlade, ukrajinski zunanji minister itd.). Iz tega seznama je mogoče razumeti, da želijo avtorji izluščenih tvitov pridobiti pozornost najvplivnejših državnih in mednarodnih političnih figur, morda s kakšnim predlogom za začetek mirovnih pogajanj, prekinitev spopadov ali za dodatno pošiljanje zahodne pomoči Ukrajini. Pomembno pa se mi zdi izpostaviti, da je tudi ameriški podjetnik južnoafriških korenin Elon Musk, ki je konec oktobra 2022 prevzel platformo Twitter (Clayton & Hosking, 2022), velikokrat

omenjen v tvitih. Sklepam, da predvsem zaradi njegove odločitve, da nekaj dni po začetku spopadov dobavi Ukrajini prvo pošiljko Starlink terminalov, ki je tej državi bistveno pomagala pri ohranjanju telekomunikacijskega in internetnega omrežja tako za civilne kot tudi za vojaške namene, šest mesecev pozneje pa je dejal, da ga ta storitev preveč stane (Marquardt, 2022). Leto dni po začetku spopadov pa se je odločil, da bo Ukrajini omejil uporabil Starlink omrežja zaradi navedb, da ga ukrajinske oborožene sile uporabljajo tudi za vodenje napadov z droni na ruske položaje (FitzGerald, 2023). Zanimivo pa se mi zdi, da med najbolj omenjenimi profili ni profila papeža Frančiška (@Pontifex), kljub njegovim rednim prizadevanjem za prekinitev spopadov (Lubov Castellano, 2023).

Glede na ure objave (slika 6) lahko sklepam, da je povprečen avtor izluščenega tvita najbolj aktiven okoli 15. ure popoldne (UTC+0) oziroma okoli 17. ure po trenutnem poletnem času v Sloveniji. Če gledam na ta podatek iz evropske perspektive, lahko trdim, da si povprečen uporabnik vzame čas za tvitanje v popoldanskih oziroma večernih urah, ko pride iz službe ali iz šole. Če pa vzamem povprečnega ameriškega uporabnika, ta tvita v dopoldanskih urah, nekje okoli 9. do 11. ure po lokalnem času, povprečen azijski uporabnik pa v večernih oziroma nočnih urah, od 20. ure dalje.

Razlike torej obstajajo v najbolj priljubljenih urah za tvitanje, obstajajo pa tudi med skupinami glede na razporeditev njihovih objav. Avtorji objav z negativnim mnenjem o vojni v Ukrajini namreč pišejo razmeroma daljše objave, najkrajše pa so objave z nevtralnimi mnenjem (slika 8). Sklepam, da je to posledica negativnih misli in obupa, ki človeka privržejo do tega, da skuša v 280 znakov strniti vse svoje misli, kar je misija nemogoče. Uporabniki z nevtralnimi mnenjem pa se verjetno ne počutijo preveč vpleteni v dogajanje oziroma se skušajo od njega čim bolj distancirati, zato tudi skrbno premislijo, kaj bodo vključili v tvit za to, da bo dosegel svoj namen. Nekaj je tudi takih uporabnikov, ki pišejo razmeroma kratke tvite (pod 100 znaki), a večina jih doseže od 150 do 200 znakov.

Avtorji nevtralnih objav, ki sem jih zgoraj opisal kot bolj objektivne osebe, imajo tudi največ sledilcev. To lahko potrjuje zgornjo trditev, saj med take avtorje sodijo predvsem bolj znane osebe, a to niso politiki, temveč pisatelji, slikarji, glasbeniki, novinarji, internetne osebnosti ter podjetniki. Razlog za njihovo objektivnost je lahko poleg iskrene nevtralnosti tudi v tem, da jih določeni poslovni interesi še vedno vežejo na Rusijo, s pretirano subjektivnimi izpadi pa bi zelo verjetno izgubili tudi ugled in status intelektualca, ki ga uživajo v javnosti (slika 13).

Avtorji nevtralnih objav pa so tisti z najvišjim deležem verificiranih uporabniških računov, kar pomeni, da so najbolj odprti javnosti in nimajo namena skrivati svojega obraza v lastnih objavah. Glede na to, da so to profili z največjim številom sledilcev, lahko sklepam, da jim je v lastnem interesu prevzemati odgovornost, predvsem pa zasluge

lastnih objav, zato bi bilo objavljane anonimnih tvitov zanje tudi škodljivo. Sledijo avtorji pozitivnih objav, avtorji negativnih pa imajo najnižji delež verificiranih uporabniških računov. Lahko bi rekel, da rezultat ni presenetljiv, saj so na družbenih omrežjih negativne objave oziroma objave polne žaljivk, laži in sovraštva take, pri katerih je avtor običajno neznan (slika 16).

Iz zemljevida lokacij (slika 20) pa lahko sklepam, da so avtorji razpršeni po celem svetu. Ob predpostavki, da so vse lokacije, ki so jih avtorji imeli navedene v svojem profilu resnične, vidimo, da je večina uporabnikov locirana v Evropi, Severni Ameriki ter Indiji. Veliko je tudi takih, ki se nahajajo v Južni Ameriki, subsaharski Afriki, Aziji ter Oceaniji. Nekaj je tudi takih na Severnem in Južnem tečaju, nekaj pa tudi po otokih v Pacifiku, Indijskem oceanu ter južnem Atlantiku. Glede na razporeditev njihovih objav lahko trdim, da pozitivne objave prevladujejo predvsem v Evropi in Severni Ameriki ter Indiji, v Rusiji pa je pozitivnih objav zelo malo. Javno mnenje se med državami zelo razlikuje, teh razlik pa ne bi pripisal drugačnemu javnemu mnenju posamezne države, ki je bolj ali manj naklonjeno vojni v Ukrajini, ampak predvsem odstotku prebivalstva posamezne države, ki obvlada angleški jezik in ga uporablja v svojih objavah na platformi Twitter. Logično je namreč, da je takih uporabnikov veliko več v ZDA kot v Rusiji, tudi zato pa je v ZDA možno zaznati veliko več pozitivnih objav. Izrisani zemljevid me ni preveč presenetil, saj sem take rezultate tudi nekoliko pričakoval. Izpostavil pa bi Avstralijo. Uradni jezik v Avstraliji je namreč angleščina, Avstralcev je preko 26 milijonov (Australian Bureau of Statistics, 2023), a se mi vseeno zdi, da je iz tega otoka razmeroma malo tvitov na to temo. Res je, da so veliki deli Avstralije nenaseljeni, da se prebivalstvo nahaja večinoma ob morju, a bi lahko pozitivne objave iz Avstralije preštel kar na roke. Sklepam, da se avstralsko javno mnenje ne zanima preveč na dogajanje v Ukrajini iz več razlogov: najprej je seveda oddaljenost spopadov, saj ti dobesedno potekajo na drugem koncu sveta (Melbourne je od Kijeva oddaljen skoraj 15.000 km), zelo verjetno pa je avstralsko javno mnenje veliko bolj osredotočeno na Kitajsko in njene grožnje ter sodelovanje v zavezništvu AUKUS (Australia, United Kingdom, United States), saj Avstralija ni država članica zveze NATO.

6.3 Omejitve raziskave in možnosti za nadaljnje delo

Izvedba analize razporeditve temelji na modelu roBERTa. Kljub temu, da je glede na opravljeno raziskavo v podpoglavju 3.3 roBERTa model najprimernejši model za mojo raziskavo, ima nekaj pomanjkljivosti. Model citatov ne obravnava najbolj učinkovito, čeprav so podani v navednicah in je po citatu naveden tudi avtor. Tak primer je recimo tweet, ki se glasi "PUTIN (2014): "With Ukraine, our western partners have crossed the line... They must have really lacked political instinct and common sense not to foresee the consequences of their actions. Russia found itself in a position it could not retreat from. If you compress the spring..."", ki ga je algoritem označil kot 77 % negativnega, 22 % nevtralnega in 1 % pozitivnega. Smotrno bi bilo, da bi bili vsi citati obravnavani

nevtravno, saj ni zagotovo, da avtor tvita, v katerem citat nastopa, deli enako mišljenje citirane osebe. Velikokrat se namreč zgodi, da avtor navede citat v svoj tvit zato, da lahko s svojimi besedami dokaže, da se citirana oseba moti, kar pomeni, da avtor tvita nasprotuje mišljenju citirane osebe. Rad bi izpostavil tudi ironične objave – tudi v tem primeru ima model roBERTa kar nekaj težav, saj zelo težko prepozna ironijo v tvitih, predvsem v tistih, kjer je zelo malo ali sploh ni konteksta. Tvit “Russia hasn't defeated Ukraine, and NATO is expanding its borders. Putin is the best master strategist and NATO recruiter.” algoritem označi kot 18,3 % negativno, 56,4 % nevtravno in 25,3 % pozitivno, jasno pa je, da tvit ni nevtralen. V prihodnjih raziskavah bi bilo smotrno model pred analizo naučiti ironije na izluščenih tvitih. Smiselno bi bilo tudi uporabiti druge razpoložljive modele in primerjati rezultate.

Naslednja omejitev raziskave izhaja iz načina pridobivanja podatkov. V pričetku izdelave programske opreme za analizo javnega mnenja na podlagi objav uporabnikov platforme Twitter na primeru vojne v Ukrajini sem na podlagi takrat veljavne politike glede zbiranja podatkov pričakoval, da bom podatke pridobil preko uradnega API-ja platforme Twitter, za uporabo katerega sem zaprosil preko obrazca na uradni spletni strani. Vendar pa je ravno v tistem času podjetje X corp, ki upravlja s platformo Twitter, spremenilo svojo politiko glede možnosti pridobivanja podatkov in je pričelo dostopanje do podatkov preko uradnega API-ja ponujati proti plačilu. Tako je trenutno za najnižjo stopnjo API dostopa potrebno plačati 100 USD na mesec (Porter, 2023). Zaradi spremembe politike sem za pridobivanje podatkov uporabil Snsrape.

Težave pa so bile tudi v samih podatkih. Kar nekajkrat se je zgodilo, da je Snsrape modul izluščil podatke v napačni obliki: tviti brez besed “ukraine” in “russia”, prazni tviti, tviti z datumom objave 1. 1. 1970 itd., za kar so bili večinoma krive motnje v internetni povezavi na moji strani ali pa težave na Twitter in/ali Snsrape strežniku. V svojo raziskavo bi lahko vključil recimo tudi komentarje, ki sem jih prvotno izločil, saj bi na tak način pridobil še večji nabor podatkov, ki bi mi koristil za prikaz širše in bolj podrobne slike o javnem mnenju Twitter uporabnikov. Ocenjujem, da bi se s tem nabor podatkov povečal za 8- do 10-krat, kar bi še dodatno otežilo izris grafikonov, še posebej pa opravljanje analize razporeditve. V raziskavo bi bilo možno vključiti tudi multimedialne vsebine, na njih pa opraviti posebno audiovizualno analizo, preko katere bi skušali razumeti, ali gre za humoristično, novinarsko, propagandistično ali kakšno drugo objavo.

Ena izmed možnosti za nadaljnje raziskovanje izhaja iz časovne dinamike raziskave. Smiselno bi bilo namreč razdeliti celotno obravnavano leto na posamezne faze; denimo tiste štiri faze, ki sem jih navedel v poglavju 1.1. Znotraj vsake faze bi bilo zanimivo bolj specifično prikazati javno mnenje in spreminjanje le-tega glede na razvoj posameznih dogodkov. V zadnji, četrti fazi, bi npr. bil fokus raziskave na mestu Bahmut, zato pa bi prišli v poštev zgolj tviti, ki vsebujejo besedo ali ključnik “bakhmut”. Na tak način bi bilo

mogoče zelo podrobno prikazati javno mnenje glede bitke in obleganja tega mesta na vzhodu Ukrajine.

Verjetno bi bilo za različne zainteresirane javnosti (npr. novinarske hiše) zanimivo pripraviti analizo spreminjanja javnega mnenja na dnevni ravni. V ta namen bi bilo potrebno razviti namensko programsko opremo, ki bi luščila le tvite, objavljene na prejšnji dan, in prikazala nihanje javnega mnenja glede na zadnje objavljene rezultate. Menim, da bi bilo takšno orodje zelo koristno predvsem za novinarske in medijske hiše, saj bi lahko na tak način odkrile, katera čustva in v kolikšni meri lahko vzbujajo njihove objave v njihovih bralcih in gledalcih.

Nenazadnje je mogoče v magistrskem delu prikazan razvoj in posamezne module kot programsko kodo uporabiti za analizo javnega mnenja na kakšnem drugem področju. To pomeni, da je možno razvito programsko opremo uporabiti tudi za analizo javnega mnenja na drugih področjih, kot so recimo šport, politika, glasba, moda, književnost, videoigre, umetna inteligenca itd. Vse, kar bi bilo v tem primeru potrebno storiti, je zamenjati iskalni niz v naprednem iskalniku platforme Twitter s poljubnim nizom, ki bi vračal željene objave.

Razvita programska oprema pa je prilagodljiva tudi za izvajanje analiz. Analizo razpoloženja, ki uporablja roBERTa algoritem, je možno zamenjati z ostalimi razpoložljivimi analizami. Algoritem roBERTa namreč podpira tudi analizo čustev znotraj objave, ironije, sovražnega govora, žaljivk ter emojijev. Algoritem podpira tudi analizo položaja avtorja objave glede splava, ateizma in podnebja, za vse to pa bi bilo potrebno spremeniti le ime algoritma, ki bi analizo izvedel.

Verjamem, da je prilagodljivost razvite programske opreme prednost, saj omogoča relativno hitro spremembo področja in načina analize, ne da bi bilo pred tem potrebno znova preiti skozi vse faze razvoja programske opreme. Potrebno bi bilo le določiti zahteve, glede na zahteve izvesti spremembe v kodi (zamenjava iskalnega niza in/ali ime algoritma za analizo), na koncu pa še testirati. Menim, da bi opisani postopek izkušeni razvijalec opravil v največ nekaj urah, tudi manj izkušeni pa bi ga moral biti sposoben izpeljati brez večjih zapletov, saj ni faz planiranja, načrtovanja in vzdrževanja.

SKLEP

V magistrskem delu sem pregledal strokovno in znanstveno domačo in tujo literaturo s področja razvoja programske opreme in analize razpoloženja. Vsa obravnavana področja so zelo obsežna in zaradi razvoja novih tehnologij, algoritmov in metod dela vsak dan bolj aktualna, zaradi česar se njihova opredelitev in področje uporabe nenehno spreminjata.

V času pisanja tega poglavja magistrskega dela se spopadi v Ukrajini nadaljujejo. Od začetnih faz invazije, ko so enote ruske vojske prodrle do in skoraj zavzele Kijev, do njihovega umika na vzhod in jug Ukrajine, od ukrajinske osvoboditve mest Herson in Harkov do bojev za Bahmut, se je stanje na terenu precej spremenilo. Res je, da je zadnje mesece frontna črta bolj statična, a to ne pomeni, da se je vojna končala. Vsakodnevno namreč izhajajo poročila o novih žrtvah, enotah in dobavah orožja, ki jih vsaka stran skuša uporabiti za doseganje končnih ciljev.

Načelnik združenega štaba ameriške vojske Mark Milley je zatrdil, da je zelo malo verjetno, da se bo vojna končala leta 2023 (Wu, 2023), kar pomeni, da ostaja ta tema še vedno zelo aktualna, ne le v Ukrajini, temveč tudi drugod po svetu. Pomembno pa je seveda tudi mnenje ljudi, ki jim ta spopad na direkten ali indirektn način kroji življenje. V ta namen je analiza javnega mnenja, opravljena z analizo razpoloženja objav uporabnikov na spletu, pravo orodje, ki lahko medijem ali vladnim organizacijam pomaga bolje razumeti dogajanje in tako pozitivno vplivati na splošno javnost.

Glede na dejstvo, da število spletnih objav na to temo zelo hitro narašča, je koristno proces analize javnega mnenja avtomatizirati, za kar je potrebno osnovno znanje programiranja in urejevalnik kode. Pred pričetkom izdelave magistrskega dela nisem podrobno poznal vseh modulov in funkcij, ki sem jih uporabil v magistrskem delu, niti najboljših algoritmov za opravljanje analize razpoloženja, zato sem preučil veliko člankov in primerov, s pomočjo katerih sem nadgradil svoje znanje do take mere, da sem lahko uspešno razvil namensko programsko opremo. Ker sem se omejil na prvo leto spopadov, sem v ta namen pridobil besedila 1.073.786 objav na platformi Twitter, za vsako objavo pa še dodatnih 16 atributov (podatke o avtorju objave in metapodatke objave). Vse zapise sem ustrezno prečistil, jim odstranil anomalije ter jih ustrezno formatiral.

Po tem, ko sem na pridobljenih podatkih opravil analizo razpoloženja, sem na podlagi rezultatov prikazal še dvajset grafikonov, ki omogočajo lažje razumevanje javnega mnenja o vojni v Ukrajini, ki je razvidno iz objav uporabnikov platforme Twitter. Na podlagi analize lahko ugotovim, da so ljudje oz. uporabniki platforme Twitter vojni nenaklonjeni. Večina uporabnikov ima namreč negativno mnenje o tej vojni, kar pa ne pomeni, da se zavzemajo ali simpatizirajo z eno ali drugo vpleteno stranjo. Iz objav je bilo sicer mogoče razbrati, da je večina uporabnikov na strani Ukrajine, a sklepam, da na to vpliva tudi dejstvo, da je Twitter ameriška družbena platforma. Rusi povečini uporabljajo platformo VKontakte. V objavah je bilo mogoče zaslediti tudi prošnje oz. predloge in kritike, ki jih hočejo ljudje nasloviti predsednikom vlad in držav vsega sveta. Čeprav se je evforija okoli vojne v splošni javnosti umirila po prvotnem šoku 24. februarja 2022, ostaja ta tema še vedno pomemben del vsakdana ljudi, tako v zunanjem svetu kot v spletnih debatah.

Glede na količino propagande in lažnih novic obeh vpletenih strani na svetovnem spletu menim, da bi morali javni in zasebni mediji ter vladne organizacije bolj aktivno spremljati in sodelovati v objavah uporabnikov na spletnih platformah, kot je npr. Twitter. To ne pomeni, da bi morale npr. vlade cenzurirati vse kar ni v interesu države, ampak da bi morale državne institucije poskrbeti za širjenje resnice in omejitev laži, ki po spletu zlahka krožijo, še posebej pa bi morale obvarovati starejše in digitalno nepismene državljane, ki so manj odporni na laži in zavajanja, ki se pojavljajo na družbenih omrežjih (Loos & Nijenhuis, 2020). Slovenija je, kot članica EU in zveze NATO, v vojno v Ukrajini vpletena indirektno, kar pomeni, da so v igri tudi slovenski državni interesi, zato bi lahko oblasti bolj podrobno preučile javno mnenje v Sloveniji na to temo in ga po potrebi oblikovale z učinkovitimi kampanjami preprečevanja laži ter direktno komunikacijo z državljani.

V magistrskem delu sem prikazal, kako poteka razvoj programske opreme za analizo javnega mnenja na podlagi objav uporabnikov platforme Twitter na primeru vojne v Ukrajini. Razvita programska oprema omogoča določeno stopnjo prilagodljivosti, ki je v današnjih hitro spreminjajočih se časih zelo priljubljena. V primeru večjih sprememb pa bi bilo smiselno obdržati le del pridobivanja podatkov iz platforme Twitter, še posebej v primeru uporabe drugih algoritmov (ne roBERTa). V primeru pridobivanja podatkov iz platform, ki jih modul Snsrape ne podpira, bi bilo potrebno razviti popolnoma novo programsko opremo, to magistrsko delo pa bi lahko bilo razvijalcem v oporo glede uporabe programskega jezika, modulov, odpravljanja najbolj pogostih napak in kakovostne interpretacije javnega mnenja v zvezi s proučevanim pojavom.

LITERATURA IN VIRI

1. Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). Agile software development methods: Review and analysis. *Proc. Espoo 2002*, str. 3-107.
2. AFP. (2023). *Russia taken 180,000 dead or wounded in Ukraine, France24*. Pridobljeno 5. junija 2023 iz <https://www.france24.com/en/live-news/20230122-russia-taken-180-000-dead-or-wounded-in-ukraine-norwegian-army>
3. Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. J. (2011). Sentiment analysis of twitter data. *Proceedings of the workshop on language in social media* (str. 30-38). Portland: Association for Computational Linguistics
4. Agarwal, R., Prasad, J., Tanniru, M. & Lynch, J. (2000). Risks of rapid application development. *Communications of the ACM*, 43(11), 1.
5. Al-Saqqa, S., Sawalha, S. & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11), 246.
6. Alshamrani, A. & Bahattab, A. (2015). A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, 12(1), 106.

7. Antezza, A., Bushnell, K., Frank, A., Frank, P., Franz, L., Kharitonov, I., . . . Schramm, S. (2023). *Ukraine Support Tracker data set*, Kiel institute for the world economy. Pridobljeno 5. junija 2023 iz <https://www.ifw-kiel.de/publications/data-sets/ukraine-support-tracker-data-17410/>
8. Apel, S. & Kästner, C. (2009). An overview of feature-oriented software development. *Journal of Object Technology*, 8(4), 49-84.
9. Arora, R. & Arora, N. (2016). Analysis of SDLC models. *International Journal of Current Engineering and Technology*, 6(1), 268-272.
10. Australian Bureau of Statistics. (2023). *Population*, Australian Bureau of Statistics. Pridobljeno 5. junija 2023 iz <https://www.abs.gov.au/statistics/people/population>
11. Bachega, H. & Murphy, M. (2022). Ukraine counter-offensive: Russian forces retreat as Ukraine takes key towns. *BBC*. Pridobljeno 5. junija 2023 iz <https://www.bbc.com/news/world-europe-62860774>
12. Balaji, S. & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26-30.
13. Barbieri, F., Camacho-Collados, J., Neves, L. & Espinosa-Anke, L. (2020). *Tweeteval: Unified benchmark and comparative evaluation for tweet classification*. School of Computer Science and Informatics, Cardiff University.
14. BBC. (2022). *Ukraine conflict: Russian forces attack from three sides*. Pridobljeno 5. junija 2023 iz <https://www.bbc.com/news/world-europe-60503037>
15. Bebler, A. (2015). Crimea and the Russian-Ukrainian Conflict. *Romanian journal of European affairs*, 15(1), 35-54.
16. Beedle, M., Devos, M., Sharon, Y., Schwaber, K. & Sutherland, J. (1999). SCRUM: An extension pattern language for hyperproductive software development. *Pattern languages of program design*, 637-651.
17. Berenbach, B. (2006). Impact of organizational structure on distributed requirements engineering processes: lessons learned. *Proceedings of the 2006 international workshop on Global software development for the practitioner* (str. 15-19). <https://doi.org/10.1145/1138506.1138511>
18. Beynon-Davies, P., Carne, C., Mackay, H. & Tudhope, D. (1999). Rapid application development (RAD): an empirical review. *European Journal of Information Systems*, 8, 211-223.
19. Bhadane, C., Dalal, H. & Doshi, H. (2015). Sentiment analysis: Measuring opinions. *Procedia Computer Science*, 45, 808-814.
20. Cerar, G. (2022). Nemški tovornjaki v zameno za slovenske tanke M55S, ki bodo poslani v Ukrajino. *RTV Slovenija*. Pridobljeno 5. junija 2023 iz <https://www.rtv slo.si/slovenija/nemski-tovornjaki-v-zameno-za-slovenske-tanke-m55s-ki-bodo-poslani-v-ukrajino/640914>
21. Cho, J. (2009). A hybrid software development method for large-scale projects: rational unified process with scrum. *Issues in Information Systems*, X(2), 340-348.

22. Clayton, J. & Hosking, P. (2022). Elon Musk takes control of Twitter in \$44bn deal. *BBC*. Pridobljeno 5. junija 2023 iz <https://www.bbc.com/news/technology-63402338>
23. Cockburn, A. (2002). Learning from agile software development—part one. *CrossTalk*, 10-14.
24. Coleman, G. & Verbruggen, R. (1998). A quality software process for rapid application development. *Software Quality Journal*, 7, 107-122.
25. Council of the EU. (2014a). *Reinforced restrictive measures against Russia*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/media/21992/144868.pdf>
26. Council of the EU. (2014b). *EU strengthens sanctions against separatists in Eastern Ukraine*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/media/21990/145967.pdf>
27. Council of the EU. (2014c). *Statement by the President of the European Council Herman Van Rompuy and the President of the European Commission in the name of*
28. *the European Union on the agreed additional restrictive measures against Russia*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/media/22015/144158.pdf>
29. Council of the EU. (2014d). *Foreign Affairs Council, 17/03/2014*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/meetings/fac/2014/03/17/>
30. Council of the EU. (2022a). *EU adopts its latest package of sanctions against Russia over the illegal annexation of Ukraine's Donetsk, Luhansk, Zaporizhzhia and Kherson regions - Consilium*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/10/06/eu-adopts-its-latest-package-of-sanctions-against-russia-over-the-illegal-annexation-of-ukraine-s-donetsk-luhansk-zaporizhzhia-and-kherson-regions/>
31. Council of the EU. (2022b). *EU adopts new set of measures to respond to Russia's military aggression against Ukraine - Consilium*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/02/28/eu-adopts-new-set-of-measures-to-respond-to-russia-s-military-aggression-against-ukraine/>
32. Council of the EU. (2022c). *EU adopts package of sanctions in response to Russian recognition of the non-government controlled areas of the Donetsk and Luhansk oblasts of Ukraine and sending of troops into the region - Consilium*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/02/23/russian-recognition-of-the-non-government-controlled-areas-of-the-donetsk-and-luhansk-oblasts-of-ukraine-as-independent-entities-eu-adopts-package-of-sanctions/>
33. Council of the EU. (2022d). *EU imposes restrictive measures on 160 individuals as a consequence of Russia's military aggression against Ukraine - Consilium*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/03/09/eu-imposes-restrictive-measures-on-160-individuals-as-a-consequence-of-russia-s-military-aggression-against-ukraine/>

34. Council of the EU. (2022e). *EU imposes sanctions on state-owned outlets RT/Russia Today and Sputnik's broadcasting in the EU - Consilium*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/03/02/eu-imposes-sanctions-on-state-owned-outlets-rt-russia-today-and-sputnik-s-broadcasting-in-the-eu/>
35. Council of the EU. (2022f). *Fifth package of sanctions in view of Russia's military aggression against Ukraine: EU adopts restrictive measures against an additional 217 individuals and 18 entities*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/04/08/fifth-package-of-sanctions-in-view-of-russia-s-military-aggression-against-ukraine-eu-adopts-restrictive-measures-against-an-additional-218-individuals-and-18-entities/>
36. Council of the EU. (2022g). *Russia's aggression against Ukraine: EU adopts "maintenance and alignment" package*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/07/21/russia-s-aggression-against-ukraine-eu-adopts-maintenance-and-alignment-package/>
37. Council of the EU. (2022h). *Russia's aggression against Ukraine: EU adopts sixth package of sanctions*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/06/03/russia-s-aggression-against-ukraine-eu-adopts-sixth-package-of-sanctions/>
38. Council of the EU. (2022i). *Russia's military aggression against Ukraine: EU agrees new sectoral measures targeting Belarus and Russia*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/03/09/russia-s-military-aggression-against-ukraine-eu-agrees-new-sectoral-measures-targeting-belarus-and-russia/>
39. Council of the EU. (2022j). *Russia's military aggression against Ukraine: EU bans certain Russian banks from SWIFT system and introduces further restrictions*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/03/02/russia-s-military-aggression-against-ukraine-eu-bans-certain-russian-banks-from-swift-system-and-introduces-further-restrictions/>
40. Council of the EU. (2022k). *Russia's military aggression against Ukraine: EU imposes sanctions against President Putin and Foreign Minister Lavrov and adopts wide ranging individual and economic sanctions*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/02/25/russia-s-military-aggression-against-ukraine-eu-imposes-sanctions-against-president-putin-and-foreign-minister-lavrov-and-adopts-wide-ranging-individual-and-economic-sanctions/>
41. Council of the EU. (2022l). *Russia's military aggression against Ukraine: fourth EU package of sectoral and individual measures*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/03/15/russia-s-military-aggression-against-ukraine-fourth-eu-package-of-sectoral-and-individual-measures/>

42. Council of the EU. (2022m). *Russia's war of aggression against Ukraine: EU adopts 9th package of economic and individual sanctions*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/12/16/russia-s-war-of-aggression-against-ukraine-eu-adopts-9th-package-of-economic-and-individual-sanctions/>
43. Council of the EU. (2022n). *Russian oil: EU agrees on level of price cap*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/press/press-releases/2022/12/03/russian-oil-eu-agrees-on-level-of-price-cap/>
44. Demianyk, G. (2022). Ukrainian Flags Raised In Kherson After Russian Retreat. *MSN*. Pridobljeno 5. junija 2023 iz <https://www.msn.com/en-us/news/world/ukrainian-flags-raised-in-kherson-after-russian-retreat/ar-AA140GJe>
45. Dickson, J. & Morrow, A. (2022). Putin signs documents to illegally annex four Ukrainian regions, in drastic escalation of Russia's war. *The Globe and Mail*. Pridobljeno 5. junija 2023 iz <https://www.theglobeandmail.com/world/article-putin-signs-documents-to-unlawfully-claim-4-ukrainian-regions-in/>
46. Dima, A. M. & Maassen, M. A. (2018). From Waterfall to Agile software: Development models in the IT sector, 2006 to 2018. Impacts on company management. *Journal of International Studies*, 11(2), 315-326.
47. Ebert, C., Abrahamsson, P. & Oza, N. (2012). Lean software development. *IEEE Software*, 29(5), 22-25.
48. European Council. (2022). *EU sanctions against Russia explained - Consilium*. Pridobljeno 5. junija 2023 iz <https://www.consilium.europa.eu/en/policies/sanctions/restrictive-measures-against-russia-over-ukraine/sanctions-against-russia-explained/>
49. FitzGerald, J. (2023). Ukraine war: Elon Musk's SpaceX firm bars Kyiv from using Starlink tech for drone control. *BBC*. Pridobljeno 5. junija 2023 iz <https://www.bbc.com/news/world-europe-64579267>
50. Galindo Salgado, P. (2021). *Python 3.11 Release Schedule*. Pridobljeno 5. junija 2023 iz <https://peps.python.org/pep-0664/>
51. Garcia, M. B. & Cunanan-Yabut, A. (2022). Public Sentiment and Emotion Analyses of Twitter Data on the 2022 Russian Invasion of Ukraine. *9th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)* (str. 242-247). Semarang: IEEE.
52. GeoPy Contributors. (2023). *Welcome to GeoPy's documentation! — GeoPy 2.3.0 documentation*. Pridobljeno 5. junija 2023 iz <https://geopy.readthedocs.io/en/stable/#>
53. Giachanou, A. & Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2), 1-41.
54. Godnov, U. & Redek, T. (2014). The use of Twitter for political purposes in Slovenia. *Romanian journal of political science*, 14(1), 4-34.
55. Gonçalves, P., Araújo, M., Benevenuto, F. & Cha, M. (2013). Comparing and combining sentiment analysis methods. *Proceedings of the first ACM conference on Online social networks* (str. 27-38).

56. Goyal, S. (2008). *Major seminar on feature driven development*. Technical University Munich.
57. Gupta, B., Negi, M., Vishwakarma, K., Rawat, G., Badhani, P. & Tech, B. (2017). Study of Twitter sentiment analysis using machine learning algorithms on Python. *International Journal of Computer Applications*, 165(9), 29-34.
58. Gutiérrez, P. & Kirk, A. (2023). A year of war: how Russian forces have been pushed back in Ukraine. *The Guardian*. Pridobljeno 5. junija 2023 iz <https://www.theguardian.com/world/ng-interactive/2023/feb/21/a-year-of-war-how-russian-forces-have-been-pushed-back-in-ukraine#:~:text=A%20new%20Guardian%20analysis%20of,in%20the%20south%20and%20east>.
59. Hossain, E., Babar, M. A. & Paik, H. Y. (2009). Using scrum in global software development: a systematic literature review. *Fourth IEEE International Conference on Global Software Engineering* (str. 175-184). Limerick: IEEE.
60. Hugging Face. (2023a). *Twitter-roBERTa-base for Sentiment Analysis*. Pridobljeno 5. junija 2023 iz <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment>
61. Hugging Face. (2023b). *Transformers*. Pridobljeno 5. junija 2023 iz <https://huggingface.co/docs/transformers/index>
62. Institute for the Study of War. (2022). *Russian Offensive Campaign Assessment - April 3, Institute for the Study of War*. Pridobljeno 5. junija 2023 iz <https://www.understandingwar.org/backgrounder/russian-offensive-campaign-assessment-april-3>
63. Iqbal, S. Z. & Idrees, M. (2017). Z-SDLC model: a new model for software development life cycle (SDLC). *International Journal of Engineering and Advanced Research Technology (IJEART)*, 3(2), 1-8.
64. Jiang, L. & Eberlein, A. (2009). An analysis of the history of classical software development and agile development. *IEEE International Conference on Systems, Man and Cybernetics* (str. 3733-3738). San Antonio: IEEE.
65. JustAnotherArchivist. (2023). *snsrape: A social networking service scraper in Python*. Pridobljeno 5. junija 2023 iz <https://github.com/JustAnotherArchivist/snsrape>
66. Kelly, L. (2022). Russian troops stationed in Belarus to begin military exercises. *Reuters*. Pridobljeno 5. junija 2023 iz <https://www.reuters.com/world/europe/russian-troops-stationed-belarus-begin-military-exercises-interfax-2022-12-19/>
67. Khurshudyan, I. & Berger, M. (2022). Why Kharkiv, a city known for its poets, has become a key battleground in Ukraine. *The Washington Post*. Pridobljeno 5. junija 2023 iz <https://www.washingtonpost.com/world/2022/02/28/kharkiv-ukraine-russia-war-east/>
68. Knific, B. (2022). Slovenija je Ukrajini predala serijo bojnih vozil pehote M-80A. *Obramba*. Pridobljeno 5. junija 2023 iz <https://www.obramba.com/novice/kopno/slovenija-je-ukrajini-predala-serijo-bojnih-vozil-pehote-m-80a/>

69. Kuchling, A. M. (2023). *Regular Expression HOWTO*. Pridobljeno 5. junija 2023 iz <https://docs.python.org/3/howto/regex.html>
70. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D. & Stoyanov, V. (2019). *Roberta: A robustly optimized bert pretraining approach*. University of Washington.
71. Loos, E. & Nijenhuis, J. (2020). Consuming Fake News: A Matter of Age? The perception of political fake news stories in Facebook ads. *Human Aspects of IT for the Aged Population. Technology and Society: 6th International Conference, ITAP 2020, Held as Part of the 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19–24, 2020, Proceedings, Part III* 22 (str. 69-88). Copenhagen: Springer International Publishing.
72. Lubov Castellano, D. (2023). Pope Francis appeals for end to "absurd and cruel" Ukraine war. *Vatican News*. Pridobljeno 5. junija 2023 iz <https://www.vaticannews.va/en/pope/news/2023-02/pope-francis-prays-for-war-torn-ukraine.html>
73. Marquardt, A. (2022). Exclusive: Musk's SpaceX says it can no longer pay for critical satellite services in Ukraine, asks Pentagon to pick up the tab. *CNN*. Pridobljeno 5. junija 2023 iz <https://edition.cnn.com/2022/10/13/politics/elon-musk-spacex-starlink-ukraine/index.html>
74. Martínez-Cámara, E., Martín-Valdivia, M. T., Urena-López, L. A. & Montejó-Ráez, A. R. (2014). Sentiment analysis in Twitter. *Natural language engineering*, 20(1), 1-28.
75. Masters, J. & Merrow, W. (2023). Council of Foreign Relations. *How Much Aid Has the U.S. Sent Ukraine? Here Are Six Charts*. Pridobljeno 5. junija 2023 iz <https://www.cfr.org/article/how-much-aid-has-us-sent-ukraine-here-are-six-charts>
76. Matharu, G. S., Mishra, A., Singh, H. & Upadhyay, P. (2015). Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1-6.
77. McBreen, S. (2015). *Announcing Visual Studio Code* [objava na blogu]. Pridobljeno 5. junija 2023 iz <https://web.archive.org/web/20151009211114/http://blogs.msdn.com/b/vscode/archive/2015/04/29/announcing-visual-studio-code-preview.aspx>
78. McCann, A., Gamio, L., Lu, D. & Robles, P. (2022). Russia Is Destroying Kharkiv. *The New York Times*. Pridobljeno 5. junija 2023 iz <https://www.nytimes.com/interactive/2022/03/17/world/europe/russia-destroying-kharkiv.html>
79. McGregor, A. (2022). Russian Airborne Disaster at Hostomel Airport. *Aberfoyle International Security*. Pridobljeno 5. junija 2023 iz <https://www.aberfoylesecurity.com/?p=4812>
80. Mejova, Y. (2009). *Sentiment analysis: An overview*. University of Iowa, Computer Science Department.
81. Meyer, H., Champion, M. & Halpin, A. (2022). Russia-Ukraine Latest: Putin to Recognize Ukraine Separatist Republics. *Bloomberg*. Pridobljeno 5. junija 2023 iz

- <https://www.bloomberg.com/news/articles/2022-02-21/putin-to-recognize-ukraine-separatist-republics-kremlin-says>
82. Microsoft. (2023a). *Excel specifications and limits*. Pridobljeno 5. junija 2023 iz <https://support.microsoft.com/en-us/office/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3>
 83. Microsoft. (2023b). *Frequently Asked Questions, Visual Studio Code*. Pridobljeno 5. junija 2023 iz <https://code.visualstudio.com/docs/supporting/faq>
 84. Microsoft. (2023c). *Why Visual Studio Code?, Visual Studio Code*. Pridobljeno 5. junija 2023 iz <https://code.visualstudio.com/docs/editor/whyvscode>
 85. MilitaryLand. (2023). *Invasion Day 362 – Summary*. Pridobljeno 5. junija 2023 iz <https://militaryland.net/news/invasion-day-362-summary/>
 86. Mills, C. (2022). House of Commons Library. *Military assistance to Ukraine 2014-2021*. Pridobljeno 5. junija 2023 iz <https://commonslibrary.parliament.uk/research-briefings/sn07135/>
 87. Mills, C. & Curtis, J. (2022). *Military assistance to Ukraine since the Russian invasion*. House of Commons Library.
 88. Ministry of Defence. (2022). *The illegal and unprovoked invasion of Ukraine is continuing* [Twitter]. Pridobljeno 5. junija 2023 iz <https://twitter.com/DefenceHQ/status/1498666642284425223>
 89. Mirza, M. S. & Datta, S. (2019). Strengths and Weakness of Traditional and Agile Processes-A Systematic Review. *Journal of Software*, 14(5), 209-219.
 90. Mueller, A. (2023). *WordCloud for Python documentation*. Pridobljeno 5. junija 2023 iz https://amueller.github.io/word_cloud/
 91. Nardelli, A. & Jacobs, J. (2021). Ukraine Update: U.S. Intel Shows Russia Troops Ready for Large-Scale Attack. *Bloomberg*. Pridobljeno 5. junija 2023 iz <https://www.bloomberg.com/news/articles/2021-11-21/u-s-intel-shows-russian-plans-for-potential-ukraine-invasion>
 92. NLTK Project. (2023). *Natural Language Toolkit*. Pridobljeno 5. junija 2023 iz <https://www.nltk.org/>
 93. NumFOCUS Inc. (2023a). *pandas 2.0.1 documentation*. Pridobljeno 5. junija 2023 iz <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>
 94. NumFOCUS Inc. (2023b). *Python Data Analysis Library*. Pridobljeno 5. junija 2023 iz <https://pandas.pydata.org/>
 95. NumPy. (2023). *NumPy*. Pridobljeno 5. junija 2023 iz <https://numpy.org/>
 96. Ojala, M. M., Pantti, M. K. & Kangas, J. (2018). Professional role enactment amid information warfare: War correspondents tweeting on the Ukraine conflict. *Journalism*, 19(3), 297-313.
 97. Osborn, A. & Nikolskaya, P. (2022). Russia's Putin authorises 'special military operation' against Ukraine. *Reuters*. Pridobljeno 5. junija 2023 iz <https://www.reuters.com/world/europe/russias-putin-authorises-military-operations-donbass-domestic-media-2022-02-24/>

98. P., E. (2001). Prenovljeni, predragi, prepočasni. *24ur*. Pridobljeno 5. junija 2023 iz <https://www.24ur.com/novice/slovenija/prenovljeni-predragi-prepocasni.html>
99. Pandas. (2023). *Pandas: Flexible and powerful data analysis / manipulation library for Python, providing labeled data structures similar to R data.frame objects, statistical functions, and much more*. Pridobljeno 5. junija 2023 iz <https://github.com/pandas-dev/pandas>
100. Pannett, R., Thebault, R., Dixon, R., Stern, D. & Harris, S. (2022). Ukraine ends Mariupol battle; Azovstal steel plant fighters evacuated. *The Washington Post*. Pridobljeno 5. junija 2023 iz <https://www.washingtonpost.com/world/2022/05/17/azovstal-mariupol-fighters-ukraine-steelworks/>
101. Pilkington, E. (2022). US intelligence believes Russia has ordered Ukraine invasion – reports. *The Guardian*. Pridobljeno 5. junija 2023 iz <https://www.theguardian.com/us-news/2022/feb/20/russia-invasion-ukraine-biden-blinken-us-national-security-council>
102. Plotly. (2023). *Plotly express in Python*. Pridobljeno 5. junija 2023 iz <https://plotly.com/python/plotly-express/>
103. Poppendieck, M. & Cusumano, M. A. (2012). Lean software development: A tutorial. *IEEE software*, 29(5), 26-32.
104. Porter, J. (2023). Twitter announces new API pricing, posing a challenge for small developers. *The Verge*. Pridobljeno 5. junija 2023 iz <https://www.theverge.com/2023/3/30/23662832/twitter-api-tiers-free-bot-novelty-accounts-basic-enterprice-monthly-price>
105. Python Software Foundation. (2023a). *Numpy*. Pridobljeno 5. junija 2023 iz <https://pypi.org/project/numpy/>
106. Python Software Foundation. (2023b). *Python*. Pridobljeno 5. junija 2023 iz <https://docs.python.org/3/library/datetime.html>
107. Python Software Foundation. (2023c). *Python 3.11.3 documentation*. Pridobljeno 5. junija 2023 iz <https://docs.python.org/3/faq/general.html>
108. Python Software Foundation. (2023d). *scipy*. Pridobljeno 5. junija 2023 iz <https://pypi.org/project/scipy/>
109. Python Software Foundation. (2023e). *What is Python? Executive Summary*. Pridobljeno 5. junija 2023 iz <https://www.python.org/doc/essays/blurb/>
110. Quinn, L. (2023). *HTML 4.0 Special Entities*. Pridobljeno 5. junija 2023 iz <https://www.htmlhelp.com/reference/html40/entities/special.html>
111. Reuters. (2022a). Factbox: Is the Kakhovka dam in Ukraine about to be blown?. *Reuters*. Pridobljeno 5. junija 2023 iz <https://www.reuters.com/world/europe/is-kakhovka-dam-ukraine-about-be-blown-2022-10-21/>
112. Reuters. (2022b). *Russia calls up 300,000 reservists, says 6,000 soldiers killed in Ukraine*. Pridobljeno 5. junija 2023 iz <https://www.reuters.com/world/europe/russias-partial-mobilisation-will-see-300000-drafted-defence-minister-2022-09-21/>

113. Santora, M., Mpoke Bigg, M. & Nechepurenko, I. (2022). 'Constant' Russian Airstrikes Stifle Ukraine's Efforts to Restore Power. *The New York Times*. Retrieved from <https://www.nytimes.com/2022/12/01/world/europe/russia-ukraine-war-infrastructure.html>
114. Sarlan, A., Nadam, C. & Basri, S. (2014). Twitter sentiment analysis. *Proceedings of the 6th International conference on Information Technology and Multimedia* (str. 212-216). Putrajaya: IEEE.
115. Schwirtz, M. & Pérez-Peña, R. (2022). Russian Forces Seize Kherson, the First Ukrainian City to Fall. *The New York Times*. Pridobljeno 5. junija 2023 iz <https://www.nytimes.com/2022/03/02/world/europe/kherson-ukraine-russia.html>
116. SciPy. (2023). SciPy. Pridobljeno 5. junija 2023 iz SciPy: <https://scipy.org/>
117. Shaydulin, R. & Sybrandt, J. (2017). *To agile, or not to agile: A comparison of software development methodologies*. <https://arxiv.org/pdf/1704.07469.pdf>
118. Shylesh, S. (2017). A study of software development life cycle process models. *National Conference on Reinventing Opportunities in Management, IT, and Social Sciences* (str. 534-541). <http://dx.doi.org/10.2139/ssrn.2988291>
119. STA. (2022). Slovenija Ukrajini že poslala puške, streliva in čelade. *N1*. Pridobljeno 5. junija 2023 iz <https://n1info.si/novice/svet/slovenija-ukrajini-ze-poslala-puske-streliva-in-celade/>
120. Stack Exchange Inc. (2022). *Stack Overflow Developer Survey 2022*. Pridobljeno 5. junija 2023 iz <https://survey.stackoverflow.co/2022/#section-most-popular-technologies-integrated-development-environment>
121. StatCounter. (2023). *Statcounter Global Stats*. Pridobljeno 5. junija 2023 iz <https://gs.statcounter.com/os-market-share/desktop/worldwide>
122. Statista Research Department. (2023). *Most sanctioned countries 2023*. Pridobljeno 5. junija 2023 iz <https://www.statista.com/statistics/1294726/number-of-global-sanctions-by-target-country/>.
123. Stoica, M., Ghilic-Micu, B., Mircea, M. & Uscatu, C. (2016). Analyzing agile development-from waterfall style to scrumban. *Informatica Economica*, 20(4), 5.
124. The Kyiv Independent. (2022). Russian troops moving towards town of Nova Kakhovka in Kherson Oblast. *The Kyiv Independent*. Pridobljeno 5. junija 2023 iz <https://kyivindependent.com/uncategorized/russian-troops-moving-towards-town-of-nova-kakhovka-in-kherson-oblast>
125. The Matplotlib development team. (2023). *Matplotlib 3.7.1 documentation*. Pridobljeno 5. junija 2023 iz <https://matplotlib.org/stable/gallery/index>
126. The Python core team and community. (2018). *Python Language Governance*. Pridobljeno 5. junija 2023 iz <https://peps.python.org/pep-0013/>
127. Twitter. (2023). *Twitter Developer Platform*. Pridobljeno 5. junija 2023 iz <https://developer.twitter.com/en/docs/counting-characters>
128. Twitter Archive. (2022). *Worldwide Twitter Trends 24/02/2022*. Pridobljeno 5. junija 2023 iz <https://archive.twitter-trending.com/worldwide/24-02-2022>

129. Twitter Support. (2022). *Twitter Support on Twitter*. Pridobljeno 5. junija 2023 iz <https://twitter.com/TwitterSupport/status/1606055187348688896>
130. Tyagi, H. (2021). *The Evolution of Tokenization – Byte Pair Encoding in NLP*. Pridobljeno 5. junija 2023 iz <https://www.freecodecamp.org/news/evolution-of-tokenization/>
131. UAwar. (2023). *Russia-Ukraine War 2022: Statistics and Facts*. Pridobljeno 5. junija 2023 iz <https://uawar.net/stats>
132. Venners, B. (2003). *The Making of Python*. Pridobljeno 5. junija 2023 iz <https://www.artima.com/articles/the-making-of-python>
133. Wadhams, N. (2022). Russia surges past Iran to become world's most sanctioned nation. *Bloomberg*. Pridobljeno 5. junija 2023 iz <https://www.bloomberg.com/news/articles/2022-03-07/russia-surges-past-iran-to-become-world-s-most-sanctioned-nation>
134. Wu, D. (2023). US General Mark Milley warns it will be 'very difficult' to eject Russian President Vladimir Putin's forces from war-torn Ukraine in 2023. *Sky News Australia*. Pridobljeno 5. junija 2023 iz <https://www.skynews.com.au/world-news/us-general-mark-milley-warns-it-will-be-very-difficult-to-eject-russian-president-vladimir-putins-forces-from-wartorn-ukraine-in-2023/news-story/b48c4929e31e593f5cac20ca7fe1feb2>

PRILOGE

Priloga 1: Primeri kod za generiranje grafikonov

Grafikon za prikaz deleža negativnih, nevtralnih in pozitivnih tvitov (slika 1)

```
total_numbers = [  
    (df["Sentiment"] == "Negative").sum(),  
    (df["Sentiment"] == "Neutral").sum(),  
    (df["Sentiment"] == "Positive").sum(),  
]  
  
plt.pie(total_numbers, labels=labels, startangle=90, colors=colors,  
autopct="%1.1f%%")  
plt.title("Delež negativnih, nevtralnih in pozitivnih tvitov")  
plt.show()
```

Grafikon za prikaz besednega oblaka najbolj pogosto uporabljenih besed (slika 2)

```
nlTK.download("stopwords")  
stop_words = stopwords.words("english")  
  
entire_text = " ".join(  
    df["Tweet"]  
    .apply(  
        lambda x: " ".join(  
            [word for word in str(x).split() if word not in (stop_words)]  
        )  
    )  
    .apply(lambda x: "".join([i for i in x if i.isalpha() or i == " "]))  
).lower()  
  
wordcloud = WordCloud(  
    max_font_size=50, max_words=50, background_color="white", collocations=False  
).generate(entire_text)  
plt.imshow(wordcloud)  
plt.title("Besedni oblak najbolj pogosto uporabljenih besed")  
plt.axis("off")  
plt.show()
```

Grafikon za prikaz besednega oblaka najbolj pogosto uporabljenih ključnikov (slika 3)

```
entire_text = " ".join(df["Tweet"]).lower()  
hashtags = " ".join(re.compile(r"#(\w+)").findall(entire_text))  
  
wordcloud = WordCloud(  
    max_font_size=50, max_words=50, background_color="white", collocations=False  
).generate(hashtags)
```

```

        max_font_size=50, max_words=50, background_color="white", collocations=False
    ).generate(hashtags)
plt.imshow(wordcloud)
plt.title("Besedni oblak najbolj pogosto uporabljenih ključnikov")
plt.axis("off")
plt.show()

```

Grafikon za prikaz besednega oblaka najbolj pogosto omenjenih uporabniških računov (slika 4)

```

entire_text = " ".join(df["Tweet"]).lower()
hashtags = " ".join(re.compile(r"@(\w+)").findall(entire_text))

wordcloud = WordCloud(
    max_font_size=50, max_words=50, background_color="white", collocations=False
).generate(hashtags)
plt.imshow(wordcloud)
plt.title("Besedni oblak najbolj pogosto omenjenih uporabniških računov")
plt.axis("off")
plt.show()

```

Grafikon za prikaz števila objavljenih tvitov glede na dan objave (slika 5)

```

df["Date"] = df["Tweet created"].dt.date
df = df.sort_values("Date")

dates = []
tweets_on_date = []
start_date = df["Date"].iloc[0]
end_date = df["Date"].iloc[-1]
delta = datetime.timedelta(days=1)
while start_date <= end_date:
    dates.append(start_date)
    tweets_on_date.append(len(df.loc[(df["Date"] == start_date)]))
    start_date += delta

plt.xticks(rotation=45)
plt.title("Število objavljenih tvitov glede na dan objave")
plt.xlabel("Dan objave")
plt.ylabel("Število objavljenih tvitov")
plt.plot(dates, tweets_on_date)
plt.show()

```

Grafikon za prikaz števila objavljenih tvitov glede na uro objave (slika 6)

```
df["Time"] = df["Tweet created"].dt.hour
df = df.sort_values("Time")

hours = []
tweets_on_hour = []
start_time = df["Time"].iloc[0]
end_time = df["Time"].iloc[-1]
while start_time <= end_time:
    hours.append(start_time)
    tweets_on_hour.append(len(df.loc[(df["Time"] == start_time)]))
    start_time += 1

plt.xticks(rotation=45)
plt.title("Število objavljenih tvitov glede na uro objave")
plt.xlabel("Ura objave")
plt.ylabel("Število objavljenih tvitov")
plt.plot(hours, tweets_on_hour)
plt.show()
```

Grafikon za prikaz števila objav glede na njihovo razpoloženje in dan objave (slika 7)

```
df["Date"] = df["Tweet created"].dt.date
df = df.sort_values("Date")

dates = []
neg_tweets_on_date = []
neu_tweets_on_date = []
pos_tweets_on_date = []
start_date = df["Date"].iloc[0]
end_date = df["Date"].iloc[-1]
delta = datetime.timedelta(days=1)
while start_date <= end_date:
    dates.append(start_date)
    neg_tweets_on_date.append(
        len(df.loc[(df["Date"] == start_date) & (df["Sentiment"] == labels[0])])
    )
    neu_tweets_on_date.append(
        len(df.loc[(df["Date"] == start_date) & (df["Sentiment"] == labels[1])])
    )
    pos_tweets_on_date.append(
        len(df.loc[(df["Date"] == start_date) & (df["Sentiment"] == labels[2])])
    )
    start_date += delta
```

```

plt.plot(dates, neg_tweets_on_date, label=labels[0], color=colors[0])
plt.plot(dates, neu_tweets_on_date, label=labels[1], color=colors[1])
plt.plot(dates, pos_tweets_on_date, label=labels[2], color=colors[2])
plt.xticks(rotation=45)
plt.title("Število objav glede na njihovo razpoloženje in dan objave")
plt.xlabel("Dan objave")
plt.ylabel("Število objav")
plt.legend(loc="upper right")
plt.show()

```

Grafikon za prikaz razporeditve dolžine objav glede na njihovo razpoloženje (slika 8)

```

length_negative = np.array([])
length_neutral = np.array([])
length_positive = np.array([])
for i, row in df.iterrows():
    tweet = row["Tweet"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        length_negative = np.append(length_negative, len(tweet))
    elif sentiment == labels[1]:
        length_neutral = np.append(length_neutral, len(tweet))
    elif sentiment == labels[2]:
        length_positive = np.append(length_positive, len(tweet))
data = [length_negative, length_neutral, length_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"}
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title("Razporeditev dolžine objav glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Dolžine objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()

```

Grafikon za prikaz razporeditve števila komentarjev objav glede na njihovo razpoloženje (slika 9)

```
replies_negative = np.array([])
replies_neutral = np.array([])
replies_positive = np.array([])
for i, row in df.iterrows():
    reply = row["Reply count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        replies_negative = np.append(replies_negative, reply)
    elif sentiment == labels[1]:
        replies_neutral = np.append(replies_neutral, reply)
    elif sentiment == labels[2]:
        replies_positive = np.append(replies_positive, reply)
data = [replies_negative, replies_neutral, replies_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title("Razporeditev števila komentarjev objav glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Število komentarjev objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()
```

Grafikon za prikaz razporeditve števila delitev objav glede na njihovo razpoloženje (slika 10)

```
retweets_negative = np.array([])
retweets_neutral = np.array([])
retweets_positive = np.array([])
for i, row in df.iterrows():
    retweet = row["Retweet count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        retweets_negative = np.append(retweets_negative, retweet)
    elif sentiment == labels[1]:
        retweets_neutral = np.append(retweets_neutral, retweet)
    elif sentiment == labels[2]:
        retweets_positive = np.append(retweets_positive, retweet)
data = [retweets_negative, retweets_neutral, retweets_positive]
```

```

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title("Razporeditev števila delitev objav glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Število delitev objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()

```

Grafikon za prikaz razporeditve števila všečkov objav glede na njihovo razpoloženje (slika 11)

```

likes_negative = np.array([])
likes_neutral = np.array([])
likes_positive = np.array([])
for i, row in df.iterrows():
    likes = row["Like count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        likes_negative = np.append(likes_negative, likes)
    elif sentiment == labels[1]:
        likes_neutral = np.append(likes_neutral, likes)
    elif sentiment == labels[2]:
        likes_positive = np.append(likes_positive, likes)
data = [likes_negative, likes_neutral, likes_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title("Razporeditev števila všečkov objav glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Število všečkov objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()

```

Grafikon za prikaz razporeditve števila ogledov objav glede na njihovo razpoloženje (slika 12)

```

df["Date"] = df["Tweet created"].dt.date
df = df.sort_values("Date")

```

```

views_negative = np.array([])
views_neutral = np.array([])
views_positive = np.array([])
for i, row in df.iloc[980233:].iterrows():
    views = row["View count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        views_negative = np.append(views_negative, views)
    elif sentiment == labels[1]:
        views_neutral = np.append(views_neutral, views)
    elif sentiment == labels[2]:
        views_positive = np.append(views_positive, views)
data = [views_negative, views_neutral, views_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title("Razporeditev števila ogledov objav glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Število ogledov objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()

```

Grafikon za prikaz razporeditve števila sledilcev avtorjev objav glede na njihovo razpoloženje (slika 13)

```

followers_negative = np.array([])
followers_neutral = np.array([])
followers_positive = np.array([])
for i, row in df.iterrows():
    followers = row["Followers count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        followers_negative = np.append(followers_negative, followers)
    elif sentiment == labels[1]:
        followers_neutral = np.append(followers_neutral, followers)
    elif sentiment == labels[2]:
        followers_positive = np.append(followers_positive, followers)
data = [followers_negative, followers_neutral, followers_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)

```

```

)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title(
    "Razporeditev števila sledilcev avtorjev objav glede na njihovo
    razpoloženje"
)
plt.xlabel("Razpoloženje")
plt.ylabel("Število sledilcev avtorjev objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()

```

Grafikon za prikaz razporeditve števila sledenj avtorjev objav glede na njihovo razpoloženje (slika 14)

```

following_negative = np.array([])
following_neutral = np.array([])
following_positive = np.array([])
for i, row in df.iterrows():
    following = row["Following count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        following_negative = np.append(following_negative, following)
    elif sentiment == labels[1]:
        following_neutral = np.append(following_neutral, following)
    elif sentiment == labels[2]:
        following_positive = np.append(following_positive, following)
data = [following_negative, following_neutral, following_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title(
    "Razporeditev števila sledenj avtorjev objav glede na njihovo razpoloženje"
)
plt.xlabel("Razpoloženje")
plt.ylabel("Število sledenj avtorjev objav")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()

```

Grafikon za prikaz razporeditve števila objav avtorjev glede na njihovo razpoloženje (slika 15)

```
statuses_negative = np.array([])
statuses_neutral = np.array([])
statuses_positive = np.array([])
for i, row in df.iterrows():
    statuses = row["Statuses count"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        statuses_negative = np.append(statuses_negative, statuses)
    elif sentiment == labels[1]:
        statuses_neutral = np.append(statuses_neutral, statuses)
    elif sentiment == labels[2]:
        statuses_positive = np.append(statuses_positive, statuses)
data = [statuses_negative, statuses_neutral, statuses_positive]

bp = plt.boxplot(
    data, patch_artist=True, medianprops={"linestyle": "-", "color": "black"},
    sym=""
)
for patch, color in zip(bp["boxes"], colors):
    patch.set_facecolor(color)
plt.title("Razporeditev števila objav avtorjev glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Število objav avtorjev")
plt.xticks([1, 2, 3], [labels[0], labels[1], labels[2]])
plt.show()
```

Grafikon za prikaz deleža verificiranih avtorjev objav glede na njihovo razpoloženje (slika 16)

```
verified_negative = np.array([0, 0])
verified_neutral = np.array([0, 0])
verified_positive = np.array([0, 0])
for i, row in df.iterrows():
    verified = row["Verified user"]
    sentiment = row["Sentiment"]
    if sentiment == labels[0]:
        if verified:
            verified_negative[0] += 1
        else:
            verified_negative[1] += 1
    elif sentiment == labels[1]:
        if verified:
            verified_neutral[0] += 1
        else:
```

```

        verified_neutral[1] += 1
    elif sentiment == labels[2]:
        if verified:
            verified_positive[0] += 1
        else:
            verified_positive[1] += 1

percentage_verified_neg = (
    verified_negative[0] * 100 / (verified_negative[0] + verified_negative[1])
)
percentage_verified_neu = (
    verified_neutral[0] * 100 / (verified_neutral[0] + verified_neutral[1])
)
percentage_verified_pos = (
    verified_positive[0] * 100 / (verified_positive[0] + verified_positive[1])
)
percentage_unverified_neg = (
    verified_negative[1] * 100 / (verified_negative[0] + verified_negative[1])
)
percentage_unverified_neu = (
    verified_neutral[1] * 100 / (verified_neutral[0] + verified_neutral[1])
)
percentage_unverified_pos = (
    verified_positive[1] * 100 / (verified_positive[0] + verified_positive[1])
)
verified = [percentage_verified_neg, percentage_verified_neu,
percentage_verified_pos]
unverified = [
    percentage_unverified_neg,
    percentage_unverified_neu,
    percentage_unverified_pos,
]

plt.bar(labels, verified, color="#78A5A3")
plt.bar(labels, unverified, bottom=verified, color="#BEE3DB")
plt.title("Delež verificiranih avtorjev objav glede na njihovo razpoloženje")
plt.xlabel("Razpoloženje")
plt.ylabel("Delež verificiranih avtorjev objav")
plt.show()

```

Grafikon za prikaz seštevka vseh komentarjev, delitev in všečkov glede na razpoloženje objav (slika 17)

```

count_negative = np.array([0, 0, 0])
count_neutral = np.array([0, 0, 0])
count_positive = np.array([0, 0, 0])
for i, row in df.iterrows():

```

```

sentiment = row["Sentiment"]
replies = row["Reply count"]
retweets = row["Retweet count"]
likes = row["Like count"]
if sentiment == labels[0]:
    count_negative[0] += replies
    count_negative[1] += retweets
    count_negative[2] += likes
elif sentiment == labels[1]:
    count_neutral[0] += replies
    count_neutral[1] += retweets
    count_neutral[2] += likes
elif sentiment == labels[2]:
    count_positive[0] += replies
    count_positive[1] += retweets
    count_positive[2] += likes

replies = [count_negative[0], count_neutral[0], count_positive[0]]
retweets = [count_negative[1], count_neutral[1], count_positive[1]]
likes = [count_negative[2], count_neutral[2], count_positive[2]]

grouped_bar = pd.DataFrame(
    {"Komentarji": replies, "Delitve": retweets, "Všečki": likes}, index=labels
)
ax = grouped_bar.plot.bar(
    rot=0, color={"Komentarji": "#564946", "Delitve": "#88A2AA", "Všečki":
"#E2856E"}
)
ax.set_ylabel("Število interakcij")
ax.set_title(
    "Seštevek vseh komentarjev, delitev in všečkov glede na razpoloženje objav"
)
plt.show()

```

Grafikon za prikaz pogostosti uporabe besed “war” in “special military operation” znotraj objav glede na njihov datum objave (slika 18)

```

df["Date"] = df["Tweet created"].dt.date
df = df.sort_values("Date")

dates = []
war_on_date = []
special_military_operation_on_date = []
start_date = df["Date"].iloc[0]
end_date = df["Date"].iloc[-1]
delta = datetime.timedelta(days=1)
while start_date <= end_date:

```

```

dates.append(start_date)
war_on_date.append(
    len(
        df.loc[
            (df["Date"] == start_date)
            & (df["Tweet"].apply(str.lower).str.contains("war"))
        ]
    )
)
special_military_operation_on_date.append(
    len(
        df.loc[
            (df["Date"] == start_date)
            & (
                df["Tweet"]
                .apply(str.lower)
                .str.contains("special military operation")
            )
        ]
    )
)
start_date += delta

plt.plot(dates, war_on_date, label="War", color="#253237")
plt.plot(
    dates,
    special_military_operation_on_date,
    label="Special military operation",
    color="#806D40",
)
plt.xticks(rotation=45)
plt.title(
    'Pogostost uporabe besed "war" in "special military operation" znotraj objav
glede na njihov datum objave'
)
plt.xlabel("Datum objave")
plt.ylabel("Število pojavitev")
plt.legend(loc="upper right")
plt.show()

```

Grafikon za prikaz števila uporabniških računov glede na leto njihovega odprtja (slika 19)

```

df["Date"] = df["User created"].dt.year
df = df.sort_values("Date")
df = df.drop_duplicates(subset="User ID")

```

```

dates = []
users_on_date_neg = []
users_on_date_neu = []
users_on_date_pos = []
for year in range(2006, 2023 + 1):
    dates.append(year)
    users_on_date_neg.append(
        len(df.loc[(df["Date"] == year) & (df["Sentiment"] == "Negative")])
    )
    users_on_date_neu.append(
        len(df.loc[(df["Date"] == year) & (df["Sentiment"] == "Neutral")])
    )
    users_on_date_pos.append(
        len(df.loc[(df["Date"] == year) & (df["Sentiment"] == "Positive")])
    )

plt.plot(dates, users_on_date_neg, color=colors[0])
plt.plot(dates, users_on_date_neu, color=colors[1])
plt.plot(dates, users_on_date_pos, color=colors[2])
plt.xticks(rotation=45)
plt.legend([labels[0], labels[1], labels[2]])
plt.title("Število uporabniških računov glede na leto njihovega odprtja")
plt.xlabel("Leto odprtja")
plt.ylabel("Število uporabniških računov")
plt.show()

```

Grafikon za prikaz zemljevida razpoloženja (slika 20)

```

df_map = pd.DataFrame(
    list(zip(df["Latitude"], df["Longitude"], df["Sentiment"])),
    columns=["Latitude", "Longitude", "Razpoloženje"],
)

fig = px.scatter_geo(
    df_map,
    lat="Latitude",
    lon="Longitude",
    color="Razpoloženje",
    color_discrete_map={
        labels[0]: colors[0],
        labels[1]: colors[1],
        labels[2]: colors[2],
    },
)
fig.update_layout(title="Zemljevid razpoloženja", title_x=0.5)
fig.show()

```